THROUGH LIFE SUPPORT STANDARD

TLSS Reference Data Development Methodology

LSC REFERENCE: EWSESLTLSS0605.077
EUROSTEP REFERENCE: Eurostep.ESUKPC12.000621

AUTHORS	CHECKED	APPROVED
David Price - Eurostep Phil Spiby - Eurostep	Tim Turner – LSC Group Rob Bodington - Eurostep	Simon Dick - Eurostep Paul Clark – LSC Group
ISSUE	**DATE**	**CLIENT**
2.0	01 November 2007	TLSD Policy Coordination - TLSS Project Manager

LSC Group Ltd.
Lincoln House
Fradley Park
Lichfield
Staffordshire
WS13 8RZ
United Kingdom

Eurostep Limited.
Cwttir Lane
St. Asaph
Denbighshire
LL17 0LQ
United Kingdom

Amendment Record

Issue	Date	Summary of changes
1.0	2007-02-16	Initial release
2.0	2007-11-01	Updated to reflect changes in the DEXlib development environment and to align with the 'Guidance on creating TLSS Reference Data' document.

Distribution

This document has been distributed to:

Copy No	Media	Location	Title
1	PDF File		TLSD Policy Coordination - TLSS Project Manager

Executive Summary

This report describes the proposed MoD Through Life Support Standard (TLSS) Reference Data development methodology, which builds on previous work undertaken for TES ILS Policy and Co-ordination Group under TAF Eurostep/ILS/022, in 2004, to develop a prototype Reference Data Library (RDL) server.

The purpose of the TLSS Reference Data is to support data exchange requirements for integrating TLS systems. MoD has adopted the ISO 10303-239 (PLCS) standard as the basis for that data exchange capability. ISO 10303-239 (PLCS) includes a flexible information model that using organizations can extend via a mechanism called Reference Data. A standard set of Reference Data is being defined by the OASIS PLCS Technical Committee. For the TLSS usage scenarios, MoD will extend the standard Reference Data with its own. Figure 1 illustrates how TLSS Reference Data fits into the data exchange concept, thinking of ISO 10303-239 (PLCS) as a data pipe between application systems.

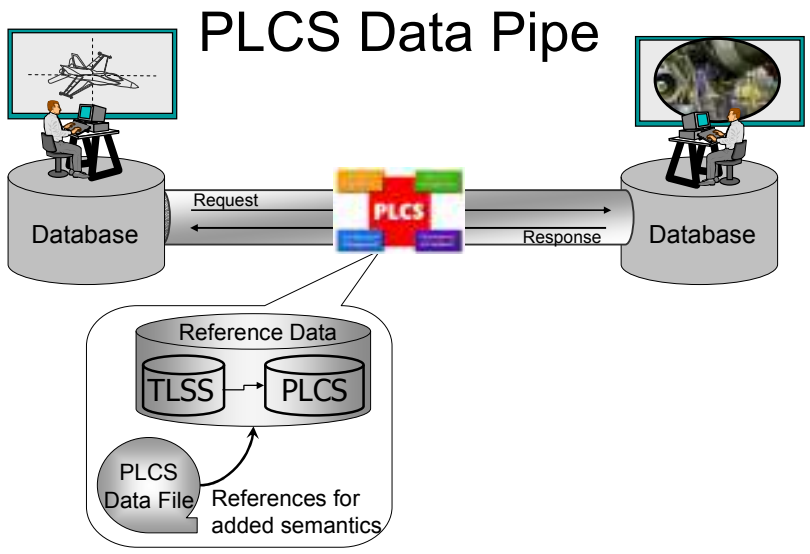


Figure 1: TLSS Reference Data usage in data exchange

This document specifies the development methodology to be used for the creation of that TLSS Reference Data. At its core Reference Data is simply an ontology, or categorization, applied to concepts defined in the ISO 10303-239 (PLCS) information model. The ISO 10303-239 (PLCS) community has adopted the W3C Web Ontology Language (OWL) for the specification of Reference Data. Therefore, TLSS Reference Data should also be represented using the OWL language. The key characteristics of TLSS Reference Data and its development methodology are:

- each phase of TLSS Reference Data development focuses on a specific information flow in the TLSS Process Model with a well-defined scope;
- the specification of the TLSS Reference Data does not force any particular implementation methodology;
- TLSS Reference Data semantics are developed based on basic set theory;
- each concept in the TLSS Reference Data is unambiguously specified;

- each concept in the TLSS Reference Data is uniquely identified and that identifier does not change over time, even as versions of the concept are managed in the library;
- concepts in the TLSS Reference Data can be marked as Obsolete or Deprecated signalling that they are not to be used any more;
- each concept in the TLSS Reference Data is directly or indirectly related to a ISO 10303-239 (PLCS) information model concept;
- the semantics of the ISO 10303-239 (PLCS) information model and OASIS PLCS Reference Data are not violated by the TLSS Reference Data;
- the TLSS Reference Data can, and should be, tested to ensure its consistency and validity prior to being released, as should the OASIS PLCS Reference Data;
- the TLSS Reference Data should be peer reviewed prior to being released;
- the TLSS Reference Data may be extended by other MoD and/or Industry projects using the same development methodology;
- the open-source Protégé-OWL toolset is recommended for TLSS Reference Data development – although any conforming OWL tool can potentially be used;
- the TLSS Reference Data, along with the OASIS PLCS Reference Data, is required when validating an ISO 10303-239 (PLCS) exchange file to check that the classifications assigned within that file are allowed.

While testing and validation of data exchanged between partners using TLSS Reference Data, as well as the Reference Data itself, is described in this document, what to do in the cases where something is found to be invalid is not within the scope of a Reference Data development methodology.

Table of Contents

Tables

Figures

1 Introduction

The Through Life Support Standard (TLSS) will provide a framework of policies, processes, procedures and data exchange specifications, designed to address the requirements of both MoD and Industry in the context of the new partnering and contracting arrangements.

UK MoD is supporting the ISO 10303-239 (PLCS) initiative to develop a new ISO standard for product life cycle support information. The development of ISO 10303-239 (PLCS) as an international standard, is now complete and the standard is available from ISO as a full International Standard.

As ISO 10303-239 (PLCS) is an International Standard that needs to satisfy diverse requirements, it has been designed to be a generic model which is tailored for specific industrial use by reference data. TLSS will become the UK MoD profile of ISO 10303-239 (PLCS), enabling its use and application in the UK Defence business environment. The core of that profile is the development of TLSS Reference Data specific to the UK MoD.

This report documents the recommended TLSS Reference Data Development Methodology and records the work undertaken by Eurostep Limited and LSC Group for the UK MoD (TES ILS Policy and Co-ordination Group) under Task Authorization Form (TAF) Eurostep 06/05. The TAF requirement was to develop the methodology and process by which the TLSS Reference Data is developed. This will include specification of the following details:

- the process by which the reference data is to be developed, promulgated as base lined releases, subsequently managed and maintained (i.e. a configuration change management process developed for the reference data);
- the methodology by which the reference data is developed. This would detail the semantic analysis required to extend the OASIS PLCS Reference Data with domain specific, e.g. TLSS reference data;
- how the reference data is to be represented. The OASIS PLCS Reference Data is defined using the W3C Web Ontology Language, OWL. The subset of OWL and the OWL-based file format to be used needs to be agreed and documented;
- the Dublin Core meta data to be associated with the reference data classes defined;
- the tools for the development of the reference data should be defined;
- the structuring of the OWL ontologies that store the reference data;
- the identification of the OWL classes.

The TAF deliverables from that requirement are:

- a documented process and methodology for developing and maintaining TLSS Reference Data (this document);
- a recommendation for the toolset to be used for developing TLSS Reference Data.

2 The TLSS Reference Data

2.1 Reference Data Roles in TLS

The focus of this development methodology is to allow MoD to create TLSS-specific Reference Data. It is expected that specific MoD projects may extend the TLSS Reference Data as required and that the same development methodology and usage scenarios will apply to any project extensions. For simplicity, the phrase "TLSS Reference Data" is used throughout the document and should be taken to mean "the MoD cross-service TLSS Reference Data including any service- or project-specific extensions". The TLSS Reference Data is to be used in at least two roles, explained in more detail below:

1. adding semantics to PLCS-based data exchange; and
2. as part of the TLSS Tailoring Tool appearing as part of contracts between the MoD and industry.

The primary purpose of the TLSS Reference Data is to support MoD-specific data exchange requirements. The TLSS Reference Data, along with the OASIS PLCS Reference Data, provide additional semantics for use with the ISO 10303-239 (PLCS) information model. The information model, or schema, is written using the ISO EXPRESS language and that schema is used to drive validation of data being exchanged. Within the data exchange file, however, classifications of the data that use Reference Data are supported. These classifications can be checked for consistency against the Reference Data itself. Therefore, the combination of the ISO 10303-239 (PLCS) schema and the complete set of related Reference Data is required in order to validate an exchange file. The Reference Data is published in a computer-interpretable language, such as OWL, in order to help enable the validation process.

While the creation of TLSS Reference Data is primarily aimed at PLCS-based exchange, it may be that the TLSS Reference Data reuses existing MoD taxonomies, ontologies, and terminology lists and may reuse existing taxonomies from other industries such as the ISO 15926 Oil and Gas community. It may also be the case that TLSS Reference Data is reused in other contexts, perhaps for example as part of a MODAF taxonomy or AP233 Systems Engineering data exchange. Additionally, future logic-based applications might make use of the TLSS Reference Data directly. Figure 2 illustrates these concepts in an example where an automated maintenance planning system directs ships to particular shipyards based on location and distance since most recent maintenance.

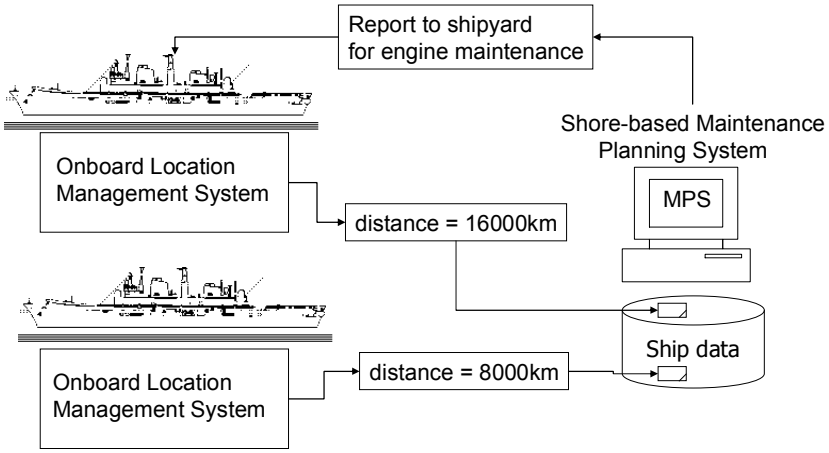


Figure 2: Possible Future Logic-Based Application Using TLSS Reference Data.

Considerations around reuse therefore have an impact on the methodology for Reference Data development, publication and maintenance. These impacts are most noticeable as:

- basing Reference Data development on set theory and representing semantics correctly;
- the decision to use mainstream technology (e.g. OWL and the Dublin Core) rather than inventing things specifically for Reference Data;
- following Semantic Web, OWL and Web best practices where possible.

The overall TLSS activity model and detailed process models will integrate the processes of existing TLS standards (and associated system specifications) and the TLSS information requirements repository will provide a consolidated and rationalized set of information requirements contained within those standards and implemented in the associated systems. The relationships between the elements of TLSS, existing TLS standards, OASIS PLCS Reference Data and the role of a TLSS Tailoring Tool are illustrated in Figure 3. TLSS Reference Data extends the OASIS PLCS Reference Data with TLSS-specific concepts as explained in the TLSS tailoring methodology [1] and software architecture [2] reports.

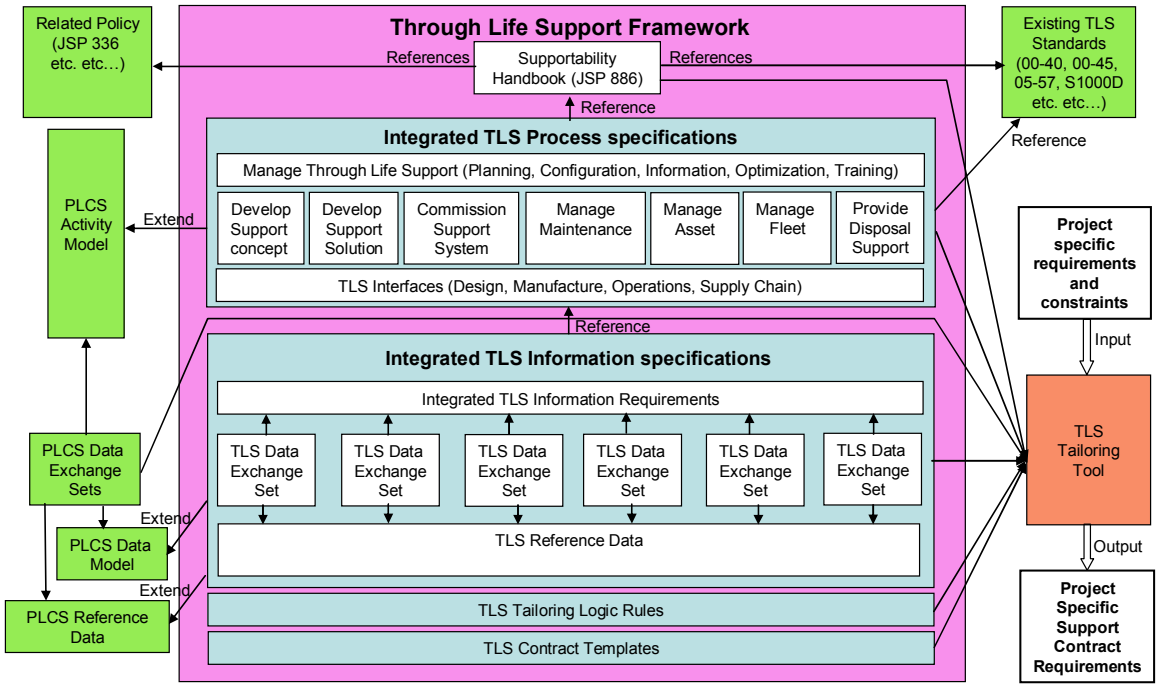


Figure 3: Relationship between the elements of TLSS, existing TLS standards, PLCS and the role of a TLSS Tailoring tool.

3 TLSS Reference Data Development Methodology Requirements

This section describes the high level requirements of the TLSS Reference Data Development Methodology and on the resulting Reference Data itself.

The use of the term "reference data" when used as "OASIS PLCS Reference Data" or "TLSS Reference Data" implies a specific set of rules about what is acceptable as valid reference data. For the purpose of specifying a process, a methodology, tools, management mechanisms and general characteristics of reference data the following definition is applied:

TLSS/PLCS Reference Data is an ontology, used during PLCS-based data exchange, that provides domain-specific semantics that extend those defined in the ISO 10303-239 (PLCS) information model. However, because of the approach taken where semantics are represented properly and widespread technology used, it is expected that the same Reference Data can be used in other application scenarios as well (e.g. a maintenance management software application).

All TLSS/PLCS Reference Data is related to concepts from the ISO 10303-239 (PLCS) information model either by being a direct, or indirect, subclass of a ISO 10303-239 (PLCS) concept or by being a class which may have ISO 10303-239 (PLCS) instances as members.

If TLSS Reference Data or data exchange requirements are found that are beyond the scope of ISO 10303-239 (PLCS), they are recorded for possible future extensions to ISO 10303-239 (PLCS). Other sources for concepts are also investigated such as whether the concept is in one of the Open Applications Group, Inc. (OAGi) Specifications.

The OASIS PLCS Reference Data is defined using the W3C Web Ontology Language (OWL) enabling direct publication on the Web. The OASIS PLCS Reference Data consists of multiple OWL ontologies designed to be extended by organizations that use Reference Data to satisfy their business specific requirements.

The TLSS Reference Data will be defined using OWL. The TLSS OWL ontologies import the entirety of the OASIS PLCS Reference Data and extend it for TLSS usage.

Despite the limitations and rules for the validity of TLSS/PLCS Reference Data, the intent is that existing or in-development ontologies, taxonomies, reference data, coding schemes, etc. within the scope of TLS can be used directly rather than having to always be rebuilt for ISO 10303-239 (PLCS) usage.

While OWL is used for TLSS/PLCS Reference Data, the development methodology and reference data itself do not prevent implementations from processing it into another form for more efficient access and usage. Therefore, efficiency of implementation is not taken into account during the reference data development.

The stability of the TLSS Reference Data is important to enable consistent usage over time. The following general requirements are applicable to the Reference Data and their related Uniform Resource Identifiers (i.e. Reference Data Identifiers).

- Reference Data Identifiers will remain stable so long as the semantics of the term do not change.
- The default Reference Data Identifiers of individual terms do not contain version information. Any version information is contained in annotation.
- If the Reference Data Library contains versions of the TLSS Reference Data, the original identifiers of the Reference Data are maintained as the default, so that usage of a specific version of the TLSS Reference Data Library requires access to a versioned snapshot that has the version included as part of its identifier.
- Individual Reference Data elements may be created, updated, obsoleted or deprecated, but not deleted.

4 The PLCS Reference Data Framework

In order that the ISO 10303-239 (PLCS) information model can be used in many different business contexts, it is deliberately generic. Precise semantic definition of constructs such as a safety critical part, or the number of gun firings property are not represented directly in the ISO 10303-239 (PLCS) standard. Instead provision is made to enable the same precision by **classifying** the basic

constructs so refining or augmenting the meaning of the ISO 10303-239 (PLCS) information model entity type.

This approach relies on the use of a common set of classes for a particular data exchange, together with a shared understanding of what each class means. This is referred to as "Reference Data" which is made available as a shared class library referred to as a "Reference Data Library" or RDL. A Reference Data Library is a managed collection of reference data. Reference data is a key success factor for consistent sharing and integration of data, i.e. to ensure consistent meaning of data. In the PLCS community a technical committee in the OASIS standards-making body is creating a cross-industry set of standardized Reference Data. Using organizations may extend that standard OASIS PLCS Reference Data Library and may, or may not, propose those extensions to the OASIS community.

The OASIS PLCS Reference Data is made up of several related datasets:

- a representation of the ISO 10303-239 (PLCS) EXPRESS schema;
- the use of a subset of the Dublin Core meta-data elements and terms for internal management of the Reference Data (e.g. creator and date modified);
- the Reference Data itself, which may include Classes, a Class hierarchy, Properties of the Classes and Instances or Individuals based on the Classes and relationships that specify which Reference Data elements are applicable for which ISO 10303-239 (PLCS) EXPRESS schema elements;
- one overarching dataset that combines the other datasets into the complete OASIS PLCS Reference Data Library.

Not all Reference Data will be standardized in OASIS, or anywhere else. Organization-specific Reference Data that extends the standard Reference Data may be defined for any particular exchange between partners. It is up to the partners to determine whether any or all of the Reference Data used in their data exchange scenario should be submitted into the OASIS, or any other, standardization process. Figure 4 shows the layers of Reference Data and how one builds upon the other.

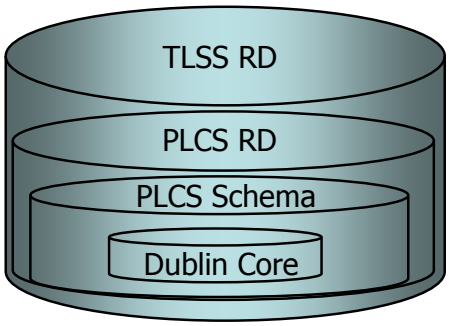


Figure 4: Layers of Reference Data.

The TLSS Reference Data is the top layer, imports other necessary Reference Data, and is the basis or context for TLSS data exchange. Note that for TLSS data exchange, the MoD may choose to ignore the OASIS PLCS In-Development Reference Data until it is more stable – or perhaps to contribute to its development.

For the purpose of enabling validation of ISO 10303-239 (PLCS) data exchange files, defining the OWL ontology (or ontologies) that provide the context of the exchange is required. The term "provide the context" includes:

1. specifying the possible OWL reference data classes that are available for use as part of the exchange
2. specifying the subclass/superclass relationships between those OWL reference data classes
3. specifying which OWL reference data classes are valid classifications of which ISO 10303-239 (PLCS) entity types

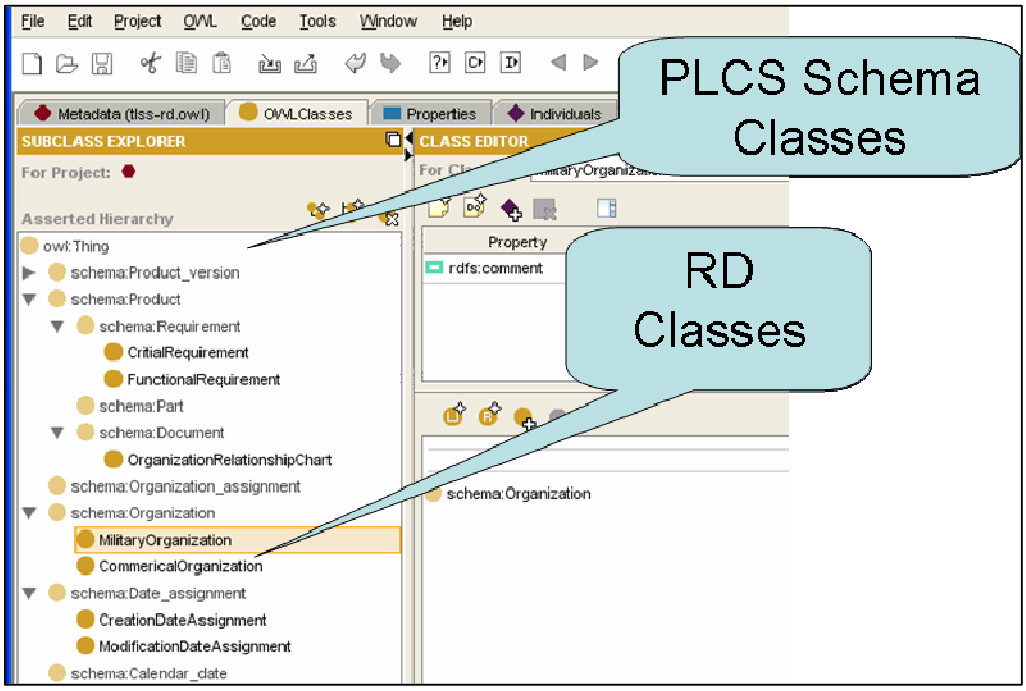


Figure 5 Development of Reference Data Using Protégé

Multiple ontologies can be combined for use in the same exchange file (e.g. the standard OASIS PLCS classes and a set of MODAF classes could be used together, if required). In these cases, a mechanism is required to specify that a particular set of ontologies constitute the context for the exchange (as defined earlier). The mechanism for doing that, which therefore enables validation, is to specify one ontology that imports all the other ontologies – either directly or transitively.

5 Alternative TLSS Reference Data Methodology Approaches

As TLSS Reference Data will be an ontology specified using the OWL language, and both OWL and ontologies are widely-used, numerous development approaches exist. In fact, that was one of the rationales for choosing the OWL language – ontology development is not a specialized process and therefore different organizations can adopt different processes, methodologies and toolsets yet still produce interoperable Reference Data.

Although a specific toolset is suggested for TLSS Reference Data development, any OWL tool can be used provided it supports the OWL constructs used in the OASIS PLCS Reference Data. For example, the Object Management Group (OMG) has standardized the Ontology Definition

Metamodel (ODM) which extends the OMG Unified Modeling Language (UML) to support OWL (among other things). Therefore, ODM-conforming tools could be used by Reference Data developers who prefer a graphical environment. However, at this point in time no testing of any ODM-conforming tools was done to validate this approach.

6 The TLSS Reference Data Development Methodology

The semantic core of TLSS Reference Data Development is based in set theory. Reference Data classes are sets that have ISO 10303-239 (PLCS) data as members. Reference Data classes, including those representing the ISO 10303-239 (PLCS) entity types, have relationships that represent the relationships between sets as follows:

1) disjoint represents non-intersecting
2) subclass represents subset
3) superclass represents superset
4) overlapping represents intersecting
5) sets can be members of other sets

A summary of set theory, OWL and EXPRESS terminology is provided in the table below.

Set Theory	**OWL RD**	**EXPRESS**
Set	Class	Entity Type
Member	Individual or Instance	Entity Instance
Subset	Subclass	Subtype
Null Intersection	Disjoint	ONE OF
Intersect	Not Disjoint	ANDOR
Union of Set	Union	SELECT

Table 1: Terminology for Set Theory, OWL and EXPRESS

Figure 6 shows that the TLSS Reference Data class “Failure” is disjoint with the ISO 10303-239 (PLCS) entity types “Person” and “Requirement”.

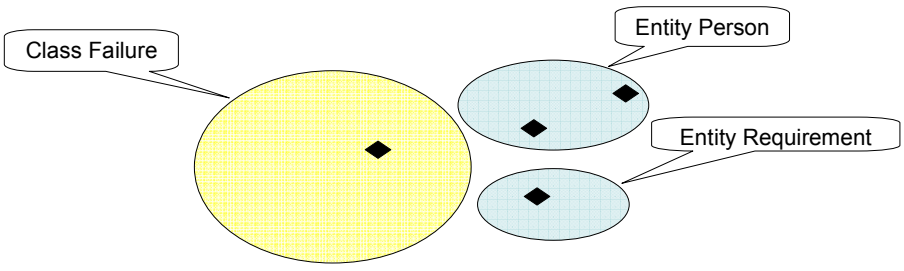


Figure 6: Disjoint/Non Overlapping Sets.

Figure 7 shows that the TLSS Reference Data class “Repair” is a subclass of the ISO 10303-239 (PLCS) entity type “Activity_actual”.

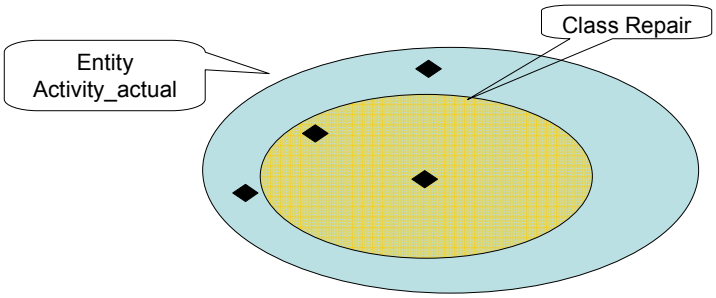


Figure 7: Subsets/Subclasses.

Figure 8 shows that the TLSS Reference Data class “ToBe Thing” overlaps with the ISO 10303-239 (PLCS) entity types “Organization” and “Document”.

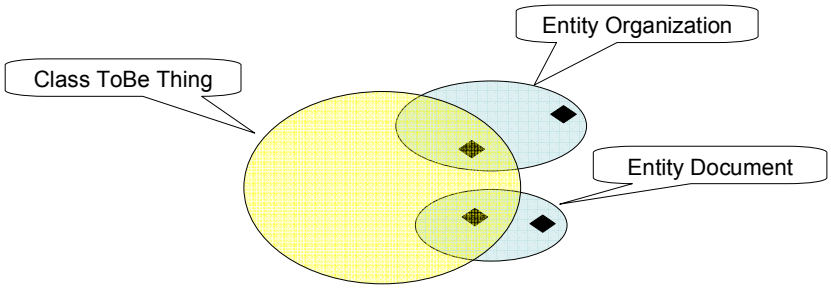


Figure 8: Intersecting sets/Overlaps.

Figure 9 shows that some sets are members of other sets. Therefore, some Reference Data classes may be members of other Reference Data classes.

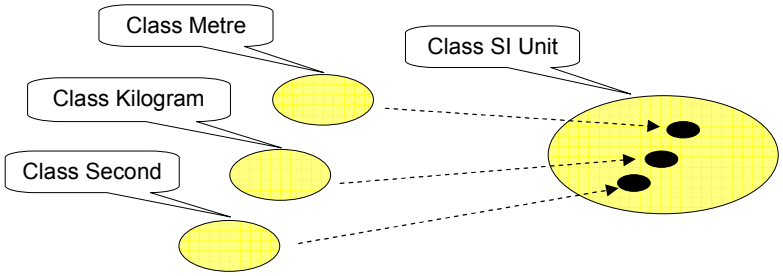


Figure 9: Sets as members of other Sets.

There are issues with the use of the OWL language and OWL-related software reasoners when OWL classes are made to be members of other OWL classes. The reasoning software, which is based on a subset of logic called Description Logic, ignores these cases. This may mean the complete set of Reference Data ontologies are not testable using the testing methodology outlined

elsewhere in this development methodology. An approach to addressing this problem is to separate these cases into a separate OWL ontology that is not included during testing, which is not ideal but does allow for testing for the simpler Reference Data cases.

The basic thought process Reference Data developers go through is:

- Identify the TLS Concept of interest.[1]
- Determine if the concept is within the scope of ISO 10303-239 (PLCS).
- Determine if the concept is in a related standard like the Open Applications Group, Inc. Specification (OAGIS) [11].
- If not in ISO 10303-239 (PLCS) and MoD thinks it should be, then use the ISO TC 184, SC4 Standards Enhancement and Discrepancy System (SEDS) process outlined elsewhere in this document.
- Determine if the concept is actually a thing, or about a thing, in the real world. If so, it's likely data in an exchange, not Reference Data.
- Determine if the concept is a class or an instance. If it's a class, it's a candidate for Reference Data. If it's an instance representing something in the real world, then it's not typically Reference Data.
- Choose examples of the TLS concept and determine of which ISO 10303-239 (PLCS) entity type(s) they are members (i.e. instances).
- Determine if the existing Reference Data related to that ISO 10303-239 (PLCS) entity type sufficient to represent the TLS concept. If not, then the need for a new TLSS Reference Data class has been found.
- Determine if the class is a subclass of another Reference Data class.
- Determine if the class is a subclass of particular ISO 10303-239 (PLCS) entity types. This is so when every member of the TLSS class is also a member of the entity type.
- Determine if the class overlaps with (i.e. is not disjoint with) particular PLCS entity type(s). This is so when some, but not all, members of the TLSS class are also members of the entity type.

To describe this within the process of developing TLSS business objects and the corresponding business templates, reference data will present itself[2] as the context in which the domain requirements are used. The domain requirements may appear to be independent of any reference data before any detailed separation of the basic underlying data.

For example, the domain requirement may refer to a date, but the date is used in the context of when something was delivered, e.g. 'date_delivered'. Another use of date might be for when an order was received, such as 'date_ordered'. Hence 'delivered_date' and 'date_ordered' might both be concepts of interest. However, the common denominator across the semantics of the requirements is 'date' which is a commonly used concept in the real world and has a PLCS equivalent (calender_date). Therefore, the date component is not reference data.

Since the commonly accepted use of calendar_date is the Gregorian one and that neither 'delivered_date' nor 'date_ordered' are defining new ways of calculating calendar_dates, they are not subtypes of calendar_date. Hence, no change is needed to the PLCS model.

[1] This could be any concept in the requirements domain – it may be reference data, it may not.
[2] To the trained eye, but will become evident after a few attempts

However, these two terms do provide a context in which the calendar_date is used. Hence they may be reference data that is used to classify the use of calendar_date in a template.

There is no specific PLCS reference data class termed 'date_delivered', but there is one which refers to 'date_actual', and the definition is close to that required. However, for the business usage, the definition needs to make reference to the fact that it refers to when the item was actually delivered. Thus, although there are several other subclasses of 'date_actual', a new reference data class termed 'date_actual_delivery' is created as a subclass of 'date_actual'.

The basic steps involved in developing the TLSS Reference Data are as follows:

1. Apply the TLSS data exchange specification development methodology [8] in the context of one, or more, of the TLSS Information Flows to select the subset of the ISO 10303-239 (PLCS) Information Model necessary for the corresponding data exchange. Identify and report gaps in ISO 10303-239 (PLCS) itself.
2. Once the subset of the ISO 10303-239 (PLCS) Information Model is defined, then the scope of the required Reference Data is known - the Reference Data will be subclasses of the classifiable entity types included within that scope.
3. Once the scope of the Reference Data is identified, the existing OASIS PLCS standard Reference Data for that scope is evaluated for use and the relevant OWL classes are identified.
4. Once the scope of the Reference Data is identified, additional sources for Reference Data that may be adopted or adapted are investigated and the applicable Reference Data is identified.
5. For each new or existing Class to be added to the Reference Data, where it fits in the ISO 10303-239 (PLCS) schema or an existing class of which it is a subclass must be identified. It is possible that a class may have more than one superclass.
6. Given the superclass(es) and sibling classes of the class in question, the definition of the class is written such that its usage is clear and helps users decide when to use the class and when not to use the class.
7. The class is added to the TLSS developer's ontology and the tasks involved in managing the process are applied. Each class must also have the required annotation created at this point in order to support the management of the TLSS Reference Data Library as a whole.

The tasks involved in managing the development process are as follows:

1. Individual modellers develop reference data as part of the appropriate DEX or development team but within separate OWL files. This involves developing classes including their definition and relevant annotation properties. The class needs to be placed in the correct place in the Reference Data class hierarchy. This involves making classes subclasses of the EXPRESS OWL classes or of an existing Reference Data class - or identifying and reporting gaps in the appropriate ISO 10303-239 (PLCS) schema.
2. The classes developed by the modellers are then peer reviewed by other TLSS Reference Data developers. This review involves checking that:
    - the required annotation properties are used;
    - the class definition is understandable and unambiguous;

- the class is a subclass of the appropriate ISO 10303-239 (PLCS) schema classes (directly or indirectly through other Reference Data classes);
- the class overlaps appropriately with the ISO 10303-239 (PLCS) schema classes;
- the class is a subclass of all appropriate Reference Data classes;
- the class does not replicate other classes *within the same context*. The context of an exchange is typically two partners. However, in some cases multiple partners may be the target of a data exchange and those partners may require the use of specific Reference Data that is similar to, overlaps or duplicates other Reference Data. In these cases, the same data in an exchange file can be classified multiple times to support each of the target partners so that a single exchange file can be produced rather than a separate one for each partner. For example, both DEF STAN 00-60 and NATO standards may result in classes that are subclasses of the ISO 10303-239 (PLCS) entity type "product" and those classes may overlap. However, in that case, the classes that overlap are not considered to be "within the same context".

3. The results and the rationale of the review are recorded in an issue log with the classes either deleted, or accepted as TLSS Reference Data. Note that detailed issues may be recorded against particular Reference Data using the tools during its development. However, general issues are likely to be better managed using a document, spreadsheet or database. Issue logs are good resources for work planning, and therefore should be managed by whatever means make them most useful in that process.
4. If appropriate, a similar review cycle then occurs but on a wider scale involving the same roles but from the OASIS PLCS development community. This supports the harmonization of Reference Data across the community.
5. The results and the rationale of the broader review are recorded in an issue log with the classes either deleted, or accepted as TLSS Reference Data – or perhaps migrated to the OASIS PLCS Reference Data that is being standardized through the OASIS processes.
6. Upon completion, the TLSS Reference Data is frozen and the new version of the ontology is made available for TLSS data exchange implementers.

It is often the case that the review and consolidation steps are performed in a workshop. A process of recording the results of the above steps should be adopted for the workshop so that decisions are not lost. Unless a minor change is made, these should be recorded outside the Reference Data itself, or at least only on a local copy of the Reference Data used only for the workshop. Trying to apply change to an ontology during the workshop may affect other issues and decisions in a workshop are often overturned upon deeper review.

6.1 Reference Data Usage of OWL Constructs

Each TLSS Reference Data element[3] is an OWL class (or an individual of a class) defined within an OWL ontology. An OWL class is identified by a Uniform Resource Identifier (URI). The following sections specify the details of the OWL classes, related constructs and identification of Reference Data.

6.1.1 Reference Data Class Identity and Versioning

The semantic concept behind a Reference Data Class is that the class has a set of, perhaps currently unknown, members. Over time, as the classes evolve, changes to the definitions and/or relationships between classes may change. For the purpose of identity and versioning of classes, if a “new version

[3] In the sense that an element could be either a class or an individual.

of a class" has a different membership than the original class had, then it is not a version of the class but in fact a new class in its own right – one that is often a superclass or subclass of the original class which may result in the original class being deprecated. For most Reference Data Classes, having a different class membership is the result of changing the criteria for membership (e.g. changing the criteria of the "HeavyTool" class from over 50 kilograms to over 100 kilograms).

Stability and reliability of the Reference Data Class URIs is critical to the Reference Data Library users [4] and are to be maintained as long as the semantic intent of the class is maintained. Changes in description of the class, including most changes of definitions and simple additions or changes in term[4] relationships[5], should not qualify as semantic changes requiring a change in a term URI. In general, non-semantic changes might be:

1. Additions of broader, narrower or related terms (i.e. subclasses and superclasses)
2. Changes in definition for clarification, correction of typos or grammar, etc.
3. Addition of definition or scope note when none is present
4. Change in term status[6]
5. Addition of other information (references, etc.)

The OASIS PLCS Reference Data guidelines specify the following concerning the identification of classes:

The class names must be unique within the context of the entire Reference Data Library. Given that the practice in the ontology community is to give meaningful names (i.e. URI fragment identifiers) for classes, the PLCS RD class names follow that practice.

As the class names are also part of a URI in the OWL language, they may not contain spaces or special characters. The convention for the name is that the first character of the first word is upper case and all other characters are lower case. Words in the class name are separated by the underscore character.

Note – The OWL class name that is part of the URI is not the same as the human interpretable label(s) that may be applied as annotation properties.

There is no technical reason for the requirement of each TLSS Reference Data class local identifier (referred as the class name above) to be unique as long as the full class URI is unique. However, supporting the use of the same term in a different context would require that the TLSS Reference Data be made up of multiple ontologies – the URI of a class includes the URI of the ontology.

The scope of TLS is vast and the TLSS Reference Data will be created and maintained over a period of time. The OWL language also only allows the import of an entire ontology, not individual classes, so multiple ontologies provide better support for reuse as well. Based on these facts and the issue of class name uniqueness, the recommended approach is to create TLSS ontologies for different business/process areas and create one full TLSS ontology that imports those as well as the OASIS PLCS Reference Data Library. This also allows much better change management and control of portions of the TLSS Reference Data by different organizations. Figure 10 shows the concept of the TLSS Reference Data being split into multiple ontologies as a trivial example of where Land, Air and Sea ontologies are managed separately. It also shows that project-specific

[4] Term is used here to represent the conceptual class being discussed.
[5] Such as disjoints, synonyms etc.
[6] e.g. if moved from developer-draft to committee-draft status.

Reference Data might import the complete TLSS Reference Data and extend it further. Care should be taken in determining how to split the TLSS Reference Data as the ontology determines part of the identifier, or URI, of the Reference Data class.

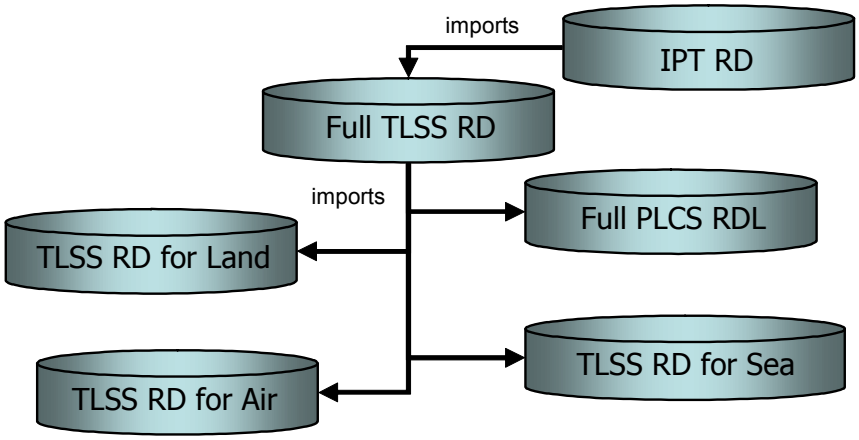
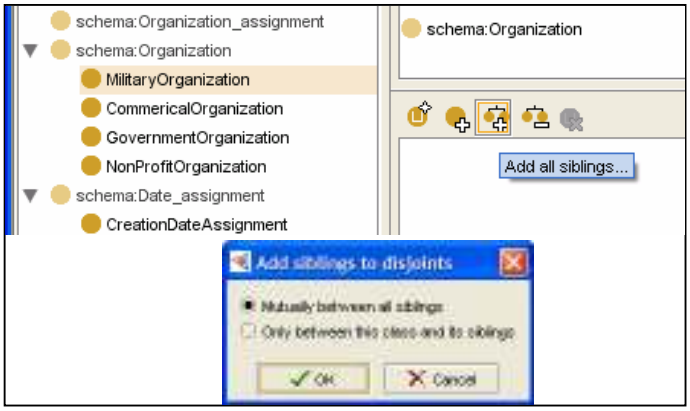


Figure 10: Example of TLSS Reference Data as Multiple Ontologies.

6.1.2 Reference Data Class Relationships and Constraints

Each TLSS OWL class must be related, either directly or indirectly, to an OWL class representing one of the ISO 10303-239 (PLCS) Information Model entity types specified in the OASIS PLCS Reference Data. Note that TLS concepts that are not within the current scope of the TLSS/PLCS Reference Data should be recorded outside the TLSS Reference Data OWL ontologies[7] for possible inclusion in a future edition of these standards, perhaps as part of an overall issue log for the TLSS project itself.

Each TLSS OWL class that has sibling classes (i.e. other classes with the same, but not necessarily immediate, superclass) should be defined as OWL disjoint with other sibling classes that do not overlap in membership (i.e. are mutually exclusive). Figure 11: Defining disjoints in and Figure 12: Disjoints shown in , illustrate these disjoint topics.



[7] For example as part of the Nato Item of Supply Codification work, it was found that codification reference data required by TLSS needs to be a direct representation that is synchronised with the reference data managed and published by the Codification Authority.

Figure 11: Defining disjoints in Protégé

A TLSS OWL class may be disjoint with ISO 10303-239 (PLCS) Information Model entity types and/or other OASIS PLCS TC and TLSS Reference Data classes.

NOTE - disjoint is **not** the OWL default so the Reference Data developer must take the time to set these constraints properly.

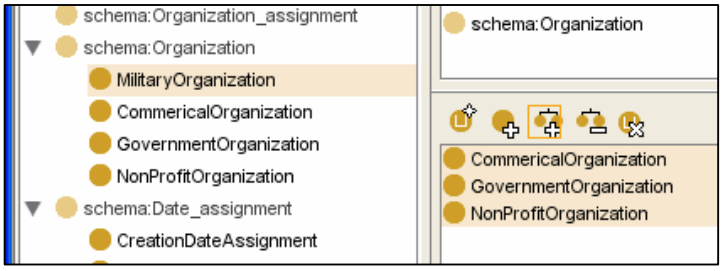


Figure 12: Disjoints shown in Protégé

A TLSS OWL class may be an overlap (i.e. not mutually exclusive) of other TLSS or OASIS PLCS OWL classes if the subclasses can co-exist within the class without semantically being contradictory. In these cases, no disjoints are defined.

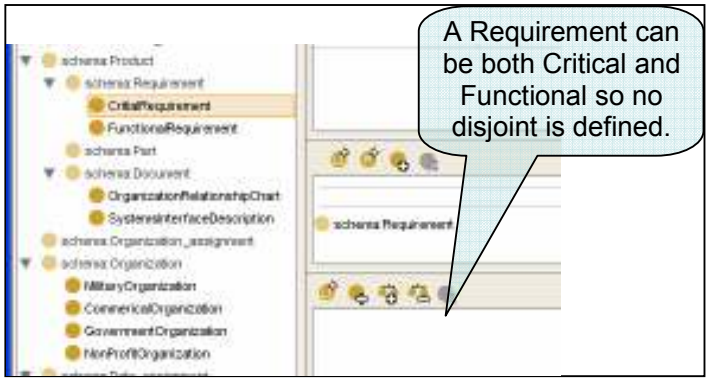


Figure 13: Overlapping Sub-classes; Not Disjointed

A TLSS OWL class may be a subclass of one or more other TLSS or OASIS PLCS OWL class(es) if it is a subset of the superclass(es) (i.e. if every member of the subclass is also a member of the superclass).

6.2 Reference Data Testing Approach

Reference Data is based in set theory. As Figure 14 shows, the relationships between Reference Data classes can grow to be complex. For this reason, testing using automated tools is important.

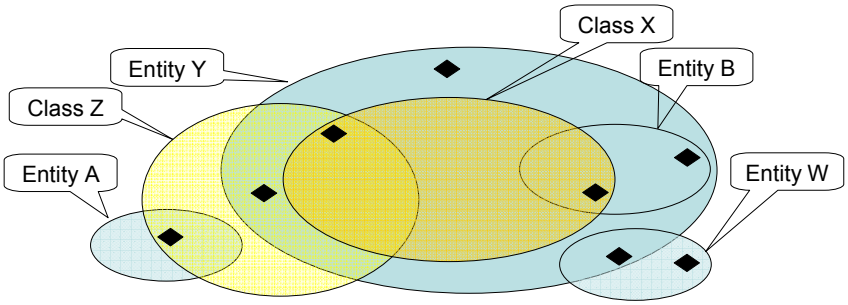


Figure 14: Complex Reference Data relationships

The TLSS Reference Data can be tested without creating any data exchange software. A data exchange pre- or post-processor would have to do two checks to validate the assignment of TLSS Reference Data in an exchange file:

1. Check whether the assigned class is related to the ISO 10303-239 (PLCS) entity type as a subclass at some level, and
2. Check whether the assigned class is disjoint with the ISO 10303-239 (PLCS) entity type or with any additional reference data assignments for the ISO 10303-239 (PLCS) data instance.

If the class is a subclass of the entity type or the class is not disjoint with the entity type, then the Reference Data assignment is valid. Testing that the specification of the Reference Data itself includes the proper constraints on applicability to a particular ISO 10303-239 (PLCS) entity type can be performed while staying in the OWL environment through the use of an OWL-capability software application called a "reasoner" or "classifier". This is equivalent to checking constraints in an EXPRESS tool or comparing an XML file against an XML Schema or DTD.

The testing process can be described as follows:

1. Create the OWL class extending the ISO 10303-239 (PLCS) entity type and add the relevant disjoints as required.
2. Create an instance of the OWL class representing the ISO 10303-239 (PLCS) entity type to which the Reference Data might be applied.
3. Using the OWL tool, assert that the ISO 10303-239 (PLCS) data instance is also of the type that is the Reference Data class.
4. Start a reasoner and request that it check the consistency of the ontology that includes both the data instance and the complete set of Reference Data.
5. The reasoner will report any inconsistencies found in the ontology (i.e. any improperly assigned Reference Data).

For example, if TLSS had a need for a Document (a subtype of Product), and needed to allow (only) a Document to be additionally asserted as either a class of "AsIs" or "ToBe[8]", then the TLSS classes of "AsIs" and "ToBe" would need to be defined in such a way to eliminate other types of Products from mistakenly being classed. In order to specify this partial overlap between PLCS and TLSS classes, all the *other* subtypes of Product would need to be specified as disjoint[9] with the TLSS classes above. This is shown in the Figure 15 below and equates to step 1 above.

[8] AsIs and ToBe are popular assertions for documents which describe a 'before' and 'after' scenario.
[9] Thereby *implying* that Document was the only legal overlap allowed.

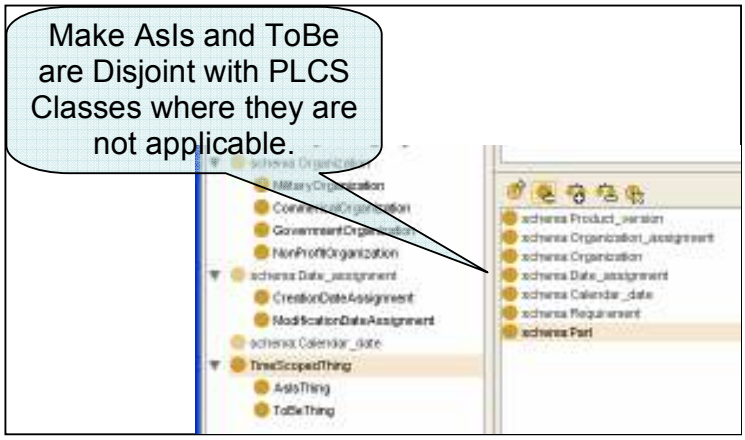


Figure 15: Overlapping PLCS and TLSS Class

Figure 16 shows how an individual of a class (Part in this case) is created (step 2), whilst Figure 17 shows how reference data can be asserted to that individual (step 3).

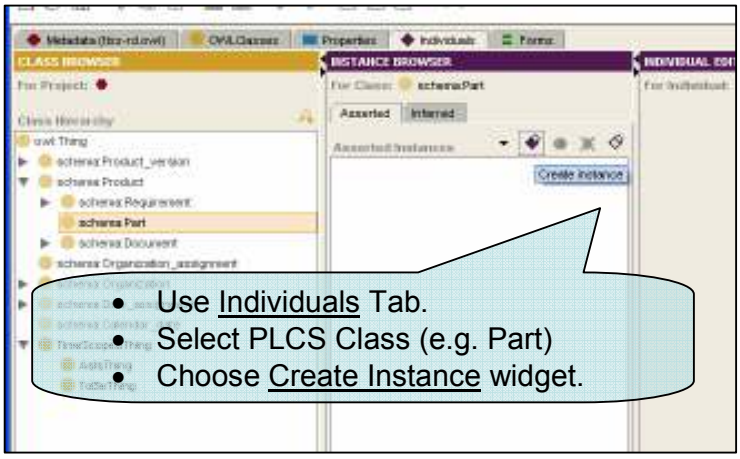


Figure 16: Creating Instances/Individuals in Protégé

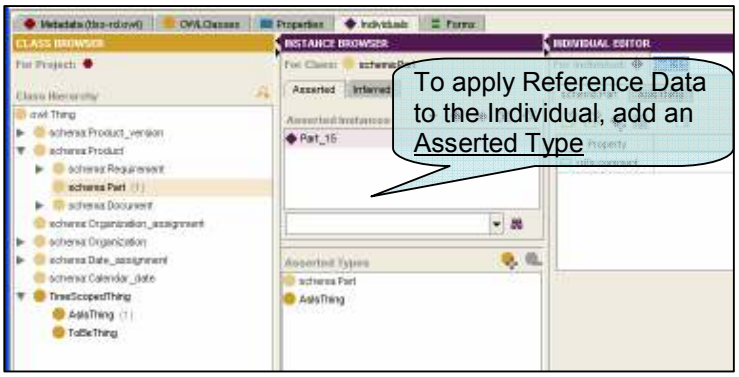


Figure 17: Asserting Reference Data to an Individual

Figure 18 shows how a reasoner is then used to validate the assertions made (step 4) and detects an error (step 5).

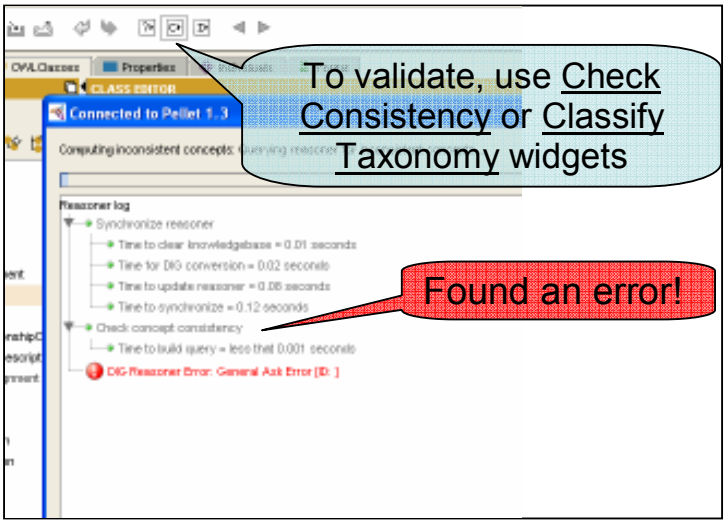


Figure 18: Validation using Protégé & Pellet

Both positive and negative testing is supported by this methodology. The above was an example of a negative test. To correct this and to perform a positive test, we would delete the invalid instance/individual of Part, create an instance of Document, assert that the Document(n)[10] is also a class of "ToBeThing" and re-run the reasoner.

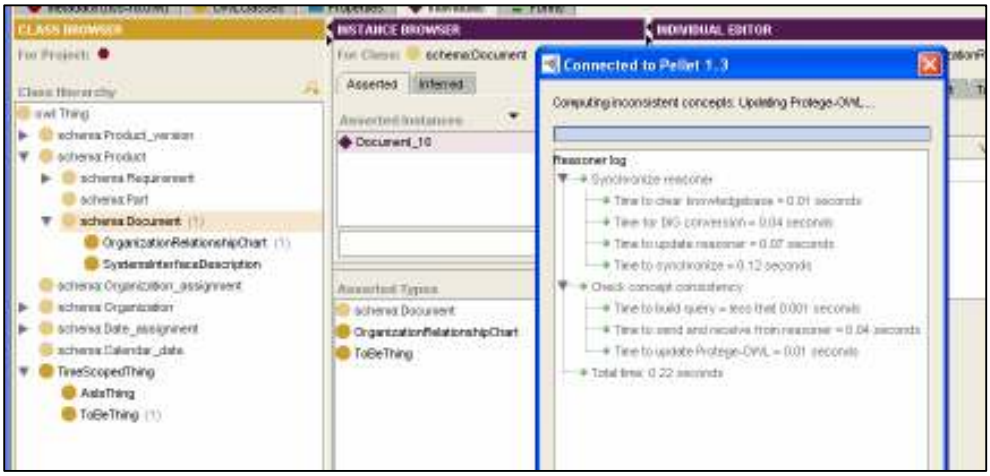


Figure 19: Positive Test using Pellet with Protégé

During the course of development, this testing should happen quite regularly as OWL classes are created, deleted, moved, subtyped or made members of other classes because of the difficulty in debugging problems. After only a handful of such changes, testing should be performed. However, editing the Dublin Core or OWL-based annotation properties (e.g. Dublin Core "creator") does not require the use of this testing approach at all. In fact, the reasoners ignore the annotations in an OWL ontology.

[10] Each individual of a class is numbered in Protégé

6.3 TLSS Reference Data Documentation and Annotation

The OASIS PLCS community has adopted a mix of the built-in OWL and Dublin Core documentation capabilities. Documentation can be embedded directly in the OWL files using what are called **annotation properties**.

The Dublin Core is an industry standard for data about Web resources managed by the Dublin Core Metadata Initiative (DCMI). Quoting from their Web site:

"*The Dublin Core Metadata Initiative is an open forum engaged in the development of interoperable online metadata standards that support a broad range of purposes and business models. DCMI's activities include consensus-driven working groups, global workshops, conferences, standards liaison, and educational efforts to promote widespread acceptance of metadata standards and practices."*

The Dublin Core elements have also been standardized as ISO 15836-2003 [3]. TLSS and the OASIS PLCS TC make use of these and additional elements not yet part of the ISO standard.

For annotating TLSS Reference Data elements, only a subset of the Dublin Core Elements and Terms are required. Further, not every Dublin Core Element or Term is applicable to every TLSS Reference Data element. For OASIS PLCS Reference Data Library, and therefore for TLSS as well, annotation properties are used as part of the process of managing the library itself.

For classes defined in the TLSS Reference Data, the following Table 1 and Table 2 show which annotation properties are to be used and how. The Dublin Core elements are prefixed by **dc** and the OWL elements are prefixed by **owl** or **rdfs**.

Note – "rdfs" is the prefix being used for the Resource Description Framework Schema, a language that underlies OWL.

Annotation Property	**Brief Description**	**Class Requirement**
rdfs:comment	Definition of item in English.	Required, set "en" language
rdfs:label	Human interpretable label in English (spaces and special characters are allowed in labels).	Required, set "en" language
owl:versionInfo	Version identifier n.nn (e.g. 0.1 or 1.01). Will be automatically set to 1.0 upon official publication of full TLSS Reference Data so use 0.n until that point.	Optional
dc:creator	Person and/or organization	Required
dc:source	Policy, standard or resource from which element was taken, use multiple dc:source annotations if required.	Optional
dc:created	Date created YYYY-MM-DD	Required
dc:references	Refers to any standard or other document referenced in the	Optional

	definition of the Reference Data item. Use multiple dc:references annotations if required.	

Table 2: The Dublin Core annotation properties for a Reference Data class

The annotation for the individual OWL ontologies is shown in the following table.

Annotation Property	Brief Description	Ontology Requirement
dc:creator	Person and organization	Required
dc:modified	Last date modified YYYY-MM-DD	Required
dc:created	Date created YYYY-MM-DD	Required
dc:abstract	Brief summary of content	Required
dc:title	Formal name	Required
dc:source	Standard or resource from which element was taken, use multiple dc:source annotations if required.	Optional
owl:versionInfo	Version identifier n.nn (e.g. 0.1 or 1.1 or 2.43). Will be automatically set to 1.0 upon official publication of full RDL.	Required
dcterms:rightsHolder	Used with rights to define any IPR.	Optional
dc:rights	IPR	Optional

Table 3: The Dublin Core annotation properties for a Reference Data ontology

More detailed explanations of these elements follow.

rdfs:comment : The OWL term "comment" is the annotation that is the human interpretable definition of the Reference Data element. Given the superclass(es) and sibling classes of the class in question, the definition of the class is written such that its usage is clear and helps users decide when to use the class and when not to use the class. The class definition needs to be understandable and unambiguous. A useful convention is "a <superclass> that <specify distinguishing characteristics here>". For example, the class SerialNumber might be defined as "an identification code that is a unique number that is one of a series assigned for identification which varies from its successor or predecessor by a fixed discrete integer value". Use the language identifier to specify the definition is English so that other language translations can be included later, if required.

rdfs:label : The OWL term "label" is the annotation that provides human interpretable names for Reference Data elements. Labels may contain spaces and punctuation and may be in multiple languages (e.g. Cat in English and Chat in French) may be defined so use the language identifier to specify the definition is English so that other language translations can be included later, if required. Note that this does not imply that the OWL Class identifier (see 6.1.1) need not be human interpretable as well; it does however have a more limited available character set.

dc:creator : The Dublin Core term "creator" is the annotation to be used to define the person and organization who added the class to the Reference Data Library. The form of the dc:creator value should be first name, last name, comma, and then organization name (e.g. John Smith, DCMA).

dc:source : The Dublin Core term "source" is the annotation to be used to define any standard or organization that originally created the class and is responsible for publishing it and controlling its development. A unique name for any dc:source must be agreed by the Reference Data developers. If the Reference Data is entirely new, then the dc:source should be set to read "TLSS RDL n.n" where "n.n" is the version of the TLSS Reference Data Library in which the Reference Data first appeared (e.g. TLSS RDL 1.0).

dc:created : The Dublin Core term "created" is the date the Reference Data was created in the format YYYY-MM-DD. There is also a Dublin Core term for "modified" but that need not be used until a second version of a TLSS Reference Data element has been published.

dc:modified : The Dublin Core term "modified" is the annotation to be used to define the date of the most recent change to the Reference Data item (e.g. YYYY-MM-DD).

dc:references : The Dublin Core term "references" specifies an identifier for some standard, policy, document, etc., upon which the definition of the Reference Data element (i.e. its dc:comment) depends.

owl:versionInfo : The OWL term "versionInfo" is used as part of the management/review process for TLSS Reference Data. Use 0.1 and increment until reaching 1.0 for the official publication of the first edition of the TLSS Reference Data. The version identifier for a Reference Data class does not change when a new version of its containing OWL ontology is published. Note that if a file-based version manager is used during development, then many versions may be created during development without causing an issue. However, once the ontology is published, version control needs to maintained more carefully (i.e. you don't want to publish hundreds of versions of an ontology).

dc:rightsHolder : The Dublin Core term "rightsHolder" is the annotation to be used to define the organization or person who owns the Intellectual Property Rights (if any) related to an RD item. It is used in conjunction with "rights". For the TLSS Reference Data, it is assumed that the rightsHolder is the TES ILS Policy and Co-ordination Group (i.e. the owning organization).

dc:rights : The Dublin Core term "rights" is the annotation to be used to define the Intellectual Property Rights related to the RD (e.g. "Access limited to MoD staff" or the URL of a Terms and Conditions statement). It is used in conjunction with "rightsHolder".

dc:title : The Dublin Core term "title" is the annotation to be used to provide the title for an OWL ontology that is part of the RDL.

dc:abstract : The Dublin Core term "abstract" is the annotation used to provide a summary of the content of the OWL ontology that is part of the RDL.

7 TLSS Reference Data Development Tools and Usage

With the requirements and methodology for TLSS Reference Data Development as input, an evaluation of the current toolset used for OASIS PLCS Reference Data development was performed. No reason to switch from in-use tools was identified. This section describes how to use the Protégé-OWL editor [5] for the development of TLSS Reference Data. Protégé OWL Release 3.2.1 is the currently available release.

The open-source Pellet reasoner [6] can be used in conjunction with the open-source Protégé-OWL editor to perform the kind of testing referred to above. Figure 20 shows a screen capture of the assignment of Reference Data (e.g. the assertion that an instance of Part has the type AsIsThing).

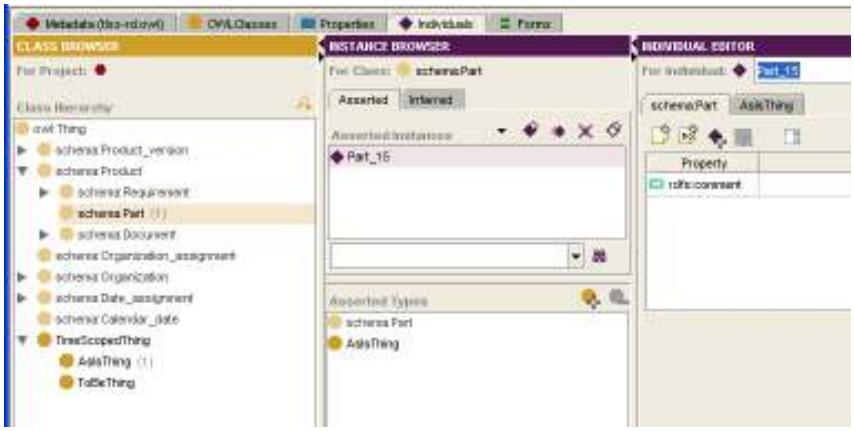


Figure 20: Using Protégé-OWL to Test Reference Data.

Other tools are likely to support testing, but they were not tested as part of the writing of this development methodology.

In order to make the TLSS Reference Data development process simpler, the initial TLSS OWL ontologies will be set up and provided to MoD. These TLSS ontologies will import the OASIS PLCS ontologies, as required. In order to allow sub-setting for performance and simplicity, TLSS developers should start by taking a local copy of the OASIS PLCS OWL files. Unnecessary classes can be removed from the local copy of the OASIS PLCS Reference Data given the particular scope, or information flow, as necessary. This includes removing the OASIS PLCS classes that represent the ISO 10303-239 (PLCS) Information Model for areas not in the current scope and OASIS PLCS Reference Data classes that are subtypes of those representing the elements of the ISO 10303-239 (PLCS) Information Model that are removed. For example, given that a DEX is (by definition) a subset of the overall PLCS Information Model, only those entities in the scope of the DEX need be used as the basis of the OWL file. This eliminates one potential cause of errors and to avoid introducing errors should be done using the Protégé OWL editor itself.

From the Protégé-OWL documentation [6]:

Protégé-OWL uses a mechanism based on the notion of "ontology repositories" to determine where an ontology should be loaded from. An ontology repository can be regarded as a "place" where ontologies can be obtained from. Given an ontology URI, a repository can be checked to see whether or not it contains the ontology that is identified by the URI. If the repository does contain the required ontology, then it acts as a gateway for loading and possibly saving the ontology. Protégé-OWL maintains a list of such repositories and searches them when attempting to import an ontology.

Traditionally, a Reference Data developer would create their own OWL file within the developer-draft folder. The TLSS project has the option to retain this structure and to additionally create a plcs-rdl-tlss.owl file which then imports from the developers on the project. A plcs-rdl-tlss.owl file is

effectively a shell at the beginning of the project, only importing the Reference Data developed by the developers. In this sense it is similar to one of the OASIS drafts. TLSS-specific Reference Data should be copied to this file at the end of the project and imports to individual developers removed[11].

Subsequent TLSS Reference Data projects should create a new local folder/directory that is their "developer repository" and copy the OASIS PLCS OWL files[12] and TLSS OWL files (if relevant) to that folder and import these into a new TLSS development file. The management, creation and deletion of ontology repositories is done via the ontology repository manager, which is accessible from the "Ontology repositories..." item on the OWL menu in Protégé-OWL. The detailed instructions are not repeated here as they available on the Stanford University Protégé-OWL Web site [7] and may be updated occasionally.

The local folder for TLSS should thus contain individual developer draft files for those developing the reference data. Repository files and project files are created by Protégé. Given that OASIS has reference data that could be at any of five stages of publication, the TLSS team has the option to import only the Committee Specification (CS) reference data, the CS and Committee Draft (CD) reference data, or the CS, CD and Public Review Draft (PRD) reference data. However, even though the filenames within the different folders may have the same name, no files will ever have the same class definitions. This is because there is a dependency between them[13]. Hence, importing the PRD, will automatically include the contents of both the CD and CS reference data. Importing the CD will automatically include the CS, whilst importing the CS will not include the others.

A typical project will thus need to define which of the OASIS files should be imported.

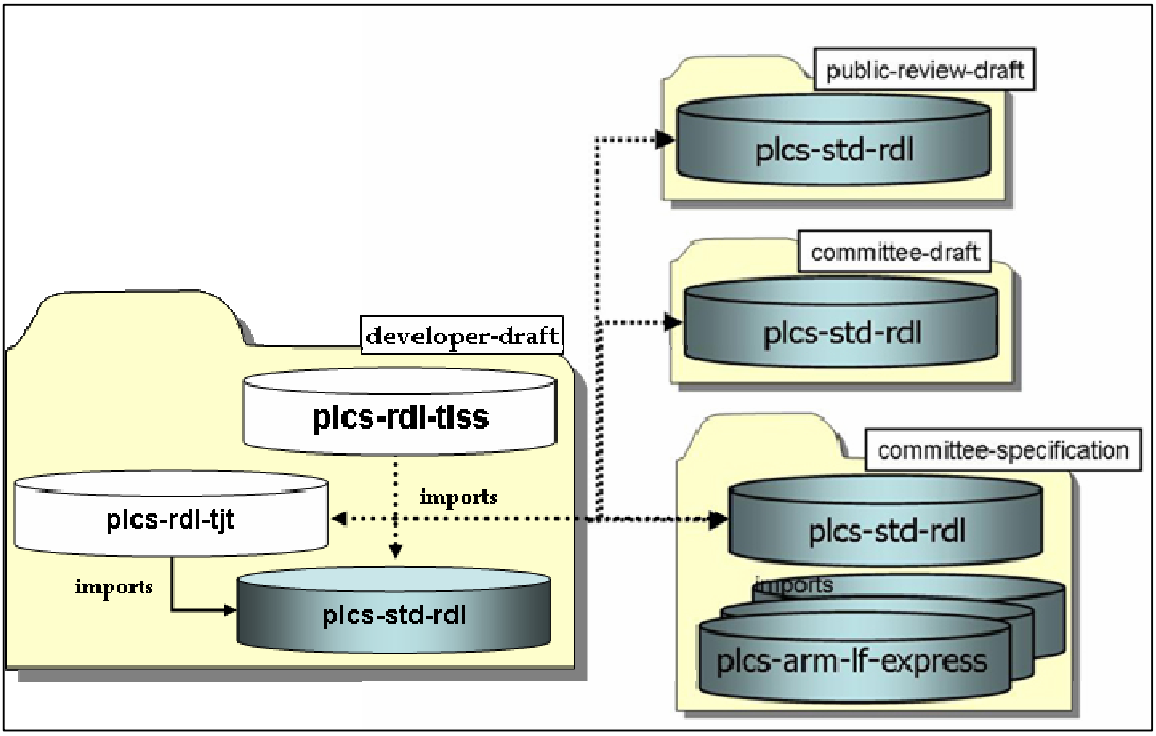


Figure 21: Typical Project Configuration Options

[11] Imports of standardized OASIS Reference Data should be kept.
[12] Such as relevant developer files – emptied of previous Reference Data, in dexlib\data\refdata\developer-draft
[13] Not shown on this diagram, but is described below

One major difference for TLSS Reference Data development from the typical use of Protégé-OWL, is that the Protégé-OWL Dublin Core files are not to be used. Instead use the OASIS PLCS TC-provided OWL ontology[14] for the Dublin Core subset used for Reference Data development. Figure 22 shows the related import to delete.

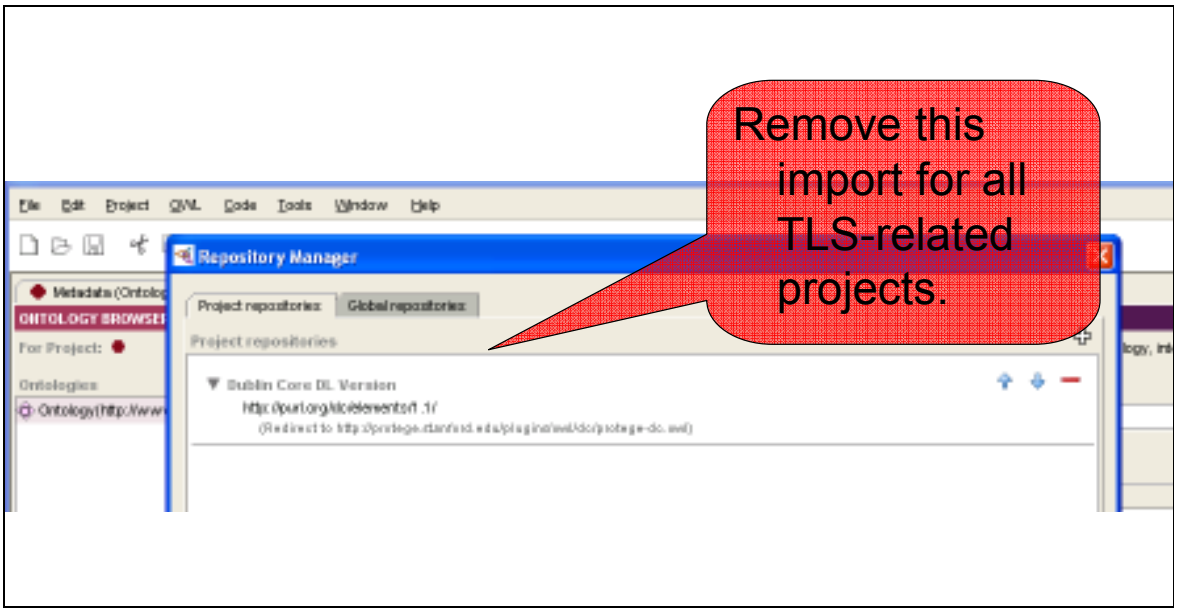


Figure 22: Do Not Use Protégé-OWL Default Dublin Core Import.

Editors Note – This methodology does not specify that the TLSS Reference Data is to be managed using the OASIS PLCS TC DEXLib-related tools. A policy decision must be made by MoD as to whether TLSS Reference Data is public or not, will be standardized using the OASIS PLCS TC or not and whether internal servers/applications will be utilized. If MoD has determined that the OASIS PLCS TC and DEXLib are to be used, then simply follow the guidelines available[15]– they are not repeated here.

8 Feedback to OASIS PLCS TC and ISO PLCS

If TLSS Reference Data developers need to provide formal feedback to the OASIS PLCS TC, feedback on OASIS work is collected through electronic mail and is publicly archived [9]. To subscribe, send a blank email message to plcs-comment-subscribe@lists.oasis-open.org. Once you confirm your subscription, you may post messages to plcs-comment@lists.oasis-open.org at any time. When posting to the comment list, please include specific information in the **Subject:** line identifying the topic, especially in communications that provide input on TC specifications.

If TLSS Reference Data developers find gaps in ISO 10303-239 (PLCS), they may use the ISO Standard Enhancement and Discrepancy System (SEDS) to submit a SEDS report. The SEDS report

[14] Located in dexlib\data\refdata\committee-specification\plcs-dc-elements-1.1.rdf-xml.owl file
[15] See http://www.plcs-resources.org/plcs/dexlib/help/dex/dvlp_refdata.htm

form [10] is available on the Web and should be emailed to seds@tc184-sc4.org. An overview of the process is shown in Figure 23.

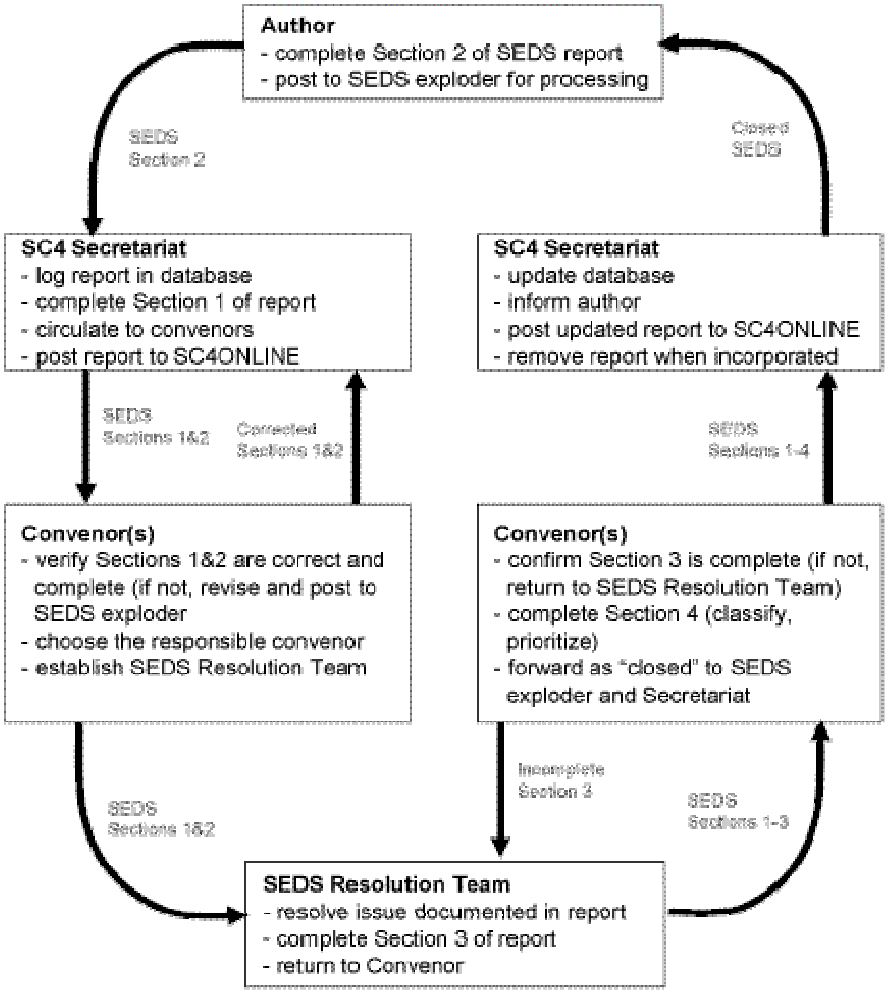


Figure 23: The ISO SEDS Process.

9 Acronyms

DC	Dublin Core
DEX	Data Exchange Set
ISO	International Organization for Standardization
MoD	Ministry of Defence
OAGIS	Open Applications Group, Inc. Specification
OASIS	Organization for the Advancement of Structured Information Standards
OMG	Object Management Group
OWL	Web Ontology Language
PLCS	Product Life Cycle Support
RDFS	Resource Description Framework Schema
RD	Reference Data
RDL	Reference Data Library
SEDS	Standards Enhancement and Discrepancy System
TC	Technical Committee
TLS	Through Life Support
TLSS	Through Life Support Standard
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URN	Uniform Resource Name
URL	Uniform Resource Locator
W3C	World Wide Web Consortium

References

1. Through Life Support Standard – Tailoring Methodology report (LSC Group Ref C/ZU/02/40 and Eurostep.ESUKPC09.130).
2. Through Life Support Standard - TLSS Tailoring Tool Software Architecture Requirements Specification, LSC Group Ref: C/ZU/02/40 and Eurostep.ESUKPC09.000129
3. ISO 15836:2003(E) Information and documentation — The Dublin Core metadata element set
4. National Science Digital Library Metadata Registry. Available on the Web at http://metadataregistry.org/
5. The Protégé-OWL editor. Available at http://protege.stanford.edu/overview/protege-owl.html
6. Pellet: An OWL DL Reasoner. Available at http://pellet.owldl.com/
7. Managing imports in protégé-owl. Available at http://protege.stanford.edu/doc/owl/owl-imports.html
8. Through Life Support Standard - TLSS Data Exchange Specification Development Methodology, LSC Group Ref: ECSMODTLSS0601 and Eurostep.ESUKPC09.000136
9. Providing Feedback to Members of the OASIS Product Life Cycle Support (PLCS) TC. Available at http://www.oasis-open.org/committees/comments/index.php?wg_abbrev=plcs
10. ISO TC184/SC4 SEDS Report Form. Available at http://www.tc184-sc4.org/SC4_Open/SC4_Standards_Developers_Info/documentation.cfm?bk=1862
11. The Open Applications Group, Incorporated Specification. Available at http://www.openapplications.org/