JORD

TSP VERIFICATION AND VALIDATION

JORD (Joint Operational Reference Data) Project
enhancing the
PCA Reference Data Service (RDS) Operation
in partnership with FIATECH

Summary –

The current document is the third in a series of reports from JORD on Template Pattern Specifications (TSP) and their implementation in the PCA RDL. These reports include specification of TSP (the first report), an overview of TSP content (the second report), and methodology for verification and validation ofTSP conformance and compliance (the current report).

This report outlines criteria and methodology for evaluating and enhancing the conformance and compliance of templates.It has been developed and tested against the template content in the PCA endpoints, as anintroductory guide to enhancing the current and adding newtemplates. As such, it can hopefully be developed into a practical tool for maintenance and improvement of the template content in the PCA RDL

Rev	Date	Description	By	Check
Ver. 0.3	Aug. 2013	Initial index based on previous project work	TCHR	
Ver. 0.4	Nov. 2013	Reviewed, updated and added structure	TCHR	
Ver. 0.5	Dec. 2013	Added textonverification and validation	TCHR	
Ver. 1.0	June 2014	Updated, corrected and completed content	TCHR	

Contents

1 Introduction

The deliverables for phase 2 of JORD Breakdown A: Compliance, Validation & Methodology are as follows:

- specifications of a set of Template Signature Patterns (TSP) described as Project Part A3,
- enhancement of the RDL to accommodate the specified TSP, described as Project Part A4,
- procedures to validate TSP Content and Business Interfaces, described as Project Part A5.

This document addresses the *third* of those deliverables by outlining criteria for conformance and compliance, and methods and tools for verifying and validating against those criteria. It builds on work completed in the first phase of the JORD project/program:

- JORD Compliance Specification a technical definition of compliance.
- JORD Mapping Methodologya methodological basis for ensuring compliant usage.

These documents recognize the need for templates and tools (procedures) which are sufficiently practical for business users and sufficiently rigorous for modelling and implementation experts. Development of such additional tools and technology are part of JORD Breakdown B.

In order to clarify terms and avoid confusion, wedistinguish between the words conformance and compliance. We *verify conformance* of Templates against the technical requirements of the ISO 15926 standard, and *validate compliance* of Templates against stated needs of end-user business applications.

The aim of the report is to (1) outline the criteria and categories forverifying and validating content,developed in Phase 1 of JORD, (2)summarize the status of the Template content in the PCA RDL against those criteria, and (3) document tools (SPARQL queries)that can be used to carry out the verification and validation.

Chapter 2 describes criteria to verify conformance of data entries with ISO 15926.

Chapter 3 describes criteria to assess conformance of template entries with technical requirements.

Chapter 4 describes criteria to assess compliance of template entries with business requirements.

Chapter 5 summarizes the conformance and compliance of the PCA RDL based on the Compliance Specification from JORD 1

Chapter 6outlinesthe queries for verifying technical structure, validating information contentand fixingRDL content.

Appendix A contain SPARQL queries to verify conformance.

Appendix B contain SPARQL queries to add new RDL content.
Appendix C contain SPARQL queries to enhance existing RDL content.

2 Verifying Conformance to technical requirements

In order to conformto the requirements of the JORD TSP Specification (new and existing) RDL content must satisfy requirements to syntax and structure from *ISO 15926 Part 2*:*Data Model*, *ISO 15926 Part 4*: *Reference Data Library* and *ISO 15926 Part 6*:*Methodology for development and validation of reference data*.

This chapter gives an overview description of the criteria for technical and standards conformance, as anextension of the requirements listed in the Compliance specificationfrom JORD Phase 1.

2.1 Verifying encoding structure and representation

Technical conformance of Templates must be verified against specifications from relevant standards bodies such as ISO and W3C. Most of these tests can be automated, and only require manual intervention when errors and omissions are discovered and reported.

2.1.1 Verify encoding and structure

The first set of checks that must be carried out is to ensure machine-readable encoding and structure. This involves checking

- Characters: names and meta-data can only contain the 26 letters of the English alphabet, and numerals 1 through 9.
- Language: All reference data names (and meta-data) must be in American English.
- Validity: All representation of reference data must follow valid XML syntax.
- Consistency: All reference data must be represented in (or translated to) valid RDF consistent with the chosen profile for OWL.

2.1.2 Verify representation

The next set of checks is to verify that the representation of data conforms to the ISO-15926 Part 2 Data Model and its Part 12 representation in OWL. This means checking

- Classification: All reference data must be classified (using RDF:Type) by a Part 12 class that represents a valid Part 2 entity type.

2.1.3 Verify content

The third set of checks is to verify against ISO 15926 Part 4 and the existing Reference Data Library. This means checking

- Duplication: the names of new reference data items must not be duplicates of any existing reference data item (within the same RDL).

- Specialization: All reference data must have a superclass that is (either) a valid RDL class (or a specialization from a valid RDL class).

- Classification: Any RDL class may be classified (RDF-Type) by (one or more) valid RDL Classes.

- Hierarchy: The classifiers of a superclass and the classifiers of it subclasses must either be equal, or have corresponding specialization relations. This means that

  If MyNewClass is represented as a subclass ofMyExistingClass,
  and entity type of MyNewClass is different from entity type of MyExistingClass,
  then entity type of MyNewClass must be a subclass of entity type of MyExistingClass

2.1.4 Verify meta-data

The fourth and final set of checks is to verify the correctness and coverage of meta-data against ISO 15926 Part 6.

- Definition: All entries must have an associated definition, which starts with the pattern "A 'superclass' that …"

- History: All entries must have a minimum of selected meta-data (Creator, Registrar, Submitting Organization and Creation Date).

2.2 Verifyingmodeling and referencing

In order to comply with requirements proper modeling and use of reference dataall (new and existing) template entries and data entities must contain (or be part of) a meaningful ontology, with content and structure relevant to stated business requirements. In general, these tests cannot be automated, but require manual definition, execution and interpretation in order to validatecompliance with the Compliance Specification from JORD Phase 1 as outlined below.

2.2.1 Semantic modeling category

The semantic modeling category is a matter of how the information, *as understood* by the business, is related to the information modeled *as intended* by the ISO-15926 lifecycle information model. This aspect is the entire subject of the JORD Mapping Methodologyand may be argued to be the *most technically significant* category in ISO-15926-specific terms.

The information at interoperability interfaces must be expressed in terms of valid Template Signatures, and may be *expanded* into explicit Part 7 Templates and *lifted* into explicit Part 2 Entity representation. The Mapping Methodology describes the way in which information at interoperability interfaces shall be represented as valid Template Signatures, by a process of selecting Template Signature Patterns and mappings to relevant reference data items. Definition and description of a basic set of Template Signature Patterns (TSP) that can be used as the basis for specifying Templates is given in the JORD TSP Specification.

The distinct compliance levels in the Semantic Modeling category are as follows:

(i) ***Dictionary & Typing Level*:** This is the minimum level of semantic compliance. As a minimum, all objects in the interface satisfy the *Registration* Template Signature Pattern (for identification, typing, classification and/or specialization templates) according to the JORD Mapping Methodology and TSP Specification.

(ii) ***Short-Cut Relations Level***: In addition to the requirements of the Dictionary & Typing Level, all data elements, properties and/or attributes shall, as a minimum, be related using Template Signature Patterns representing Part 2 relationships (Proto Templates).

(iii) ***Full Ontology Level:*** This is the highest level of semantic compliance. All data elements, properties and/or attributes shall be related using Template Signature Patterns templates according to the JORD Mapping Methodology and TSP Specification.

2.2.2 Referencing technology category

A large part of the modeling of the interoperability interface in terms of template signatures (above) involves populating template roles with valid reference data items. Compliance levels within the referencing technology category reflect the way in which reference to those reference data items are captured in the interface.

(i) ***Local Naming Level:*** The interface includes the names of reference data items already resolved explicitly from external reference data library sources.

(ii) ***URI Referencing Level:*** The interface includes URI's, which require access to on-line (or cached) reference data libraries in order for the reference data items to be resolved.

2.2.3 Representation technology category

Levels within the representation technology category reflect the choice of representation technology in the interface.

(i) ***No Explicit XML Schema Level:*** The interface format is defined according to implicit or explicit document type definition or tabular format, but without an explicit XML-Schema.

(ii) ***Explicit XML Schema Level***: The interface format is defined according to a registered XML-Schema (other than RDF/OWL).

(iii) ***RDF/OWL Schema Level***: The interface format is defined according to a registered RDF/OWL compliant XML-Schema. (e.g., as defined in Part 8.)

2.2.4 Interface technology Category

Levels within the interface technology category reflect the choice of interface paradigm for information exchange with other systems.

(i) ***File Exchange Level***: - The interface involves file exchange.

(ii) ***API Query Level***: - The interface provides an API or supports querying other than SPARQL.

(iii) ***SPARQL Query Level***: - The interface supports querying through a SPARQL-based Façade (e.g. as defined in Part 9).

3 Validatingcompliance to business

The categories to consider when validating business compliance with relevant parts of ISO 15926include industrial standardization, payload content, meta-data and change-management functionality as defined in the Compliance specificationfrom JORD Phase 1. In addition comes validation of business compliance, use case consistency and business suitability.

3.1 Validating content and meta-data

The content of Templates and reference data must be checked for business compliance against stated business objectives. Most of these tests cannot be automated, but require manual definition, execution and interpretation.

3.1.1 Industrial standardization category

JORD reference data operations recognize that sharing of industrial reference data will occur at different levels. It therefore supports a persistent globally unique reference data URI / RUID ID Specification (which can be downloaded from the PCA JORD web page), and the federation of reference data libraries at different levels of authority and across different domains of management responsibility.

(i) ***Local Sandbox Level***: - Reference data is managed and validated at and individual organization level within the business community using the interoperability interface.

(i) ***Global Industrial Level***: - Reference data is managed and validated at a level beyond the business community using the interoperability interface.

(iii) ***PCA / JORD Level***: - Reference data is managed and validated by the JORD Reference Data Operations.

(iv) ***ISO Level***: - Reference data is managed and validated by or on behalf of ISO.

3.1.2 Payload content category

It is necessary for an interoperability interface to declare the business domain content scope supported. The options possible include:

(i) ***Generic Level***: - Within the technical compliance category levels above, the interface is provided by otherwise generic tool(s) not dependent on the payload scope.

(ii) ***Explicit Scope Level***: - The interface compliance is limited to an explicit scope of payload data. That explicit scope needs to be defined and registered as a reference data item. There are

several proposed bases for defining such scopes – per defined business interface(s), per object information model(s), per application type(s).

3.1.3 Change-management meta-data category

The focus for modeling asset lifecycle information has been the payload scope representing the objects comprising the asset. For practical interoperability and integration over business process lifecycles, meta-data necessary to manage change must also be included in interoperability interfaces. The categories proposed are

(i) ***Identity Only Level:*** - As a minimum requirement, all versioned objects have business identifiers as meta-data content in the interoperability interface.

(ii) ***Version Level:*** - Business identifiers for versioned objects shall also include explicit version identity as meta-data content in the interoperability interface.

(iii) ***Status Level:*** - Each identified version (above) shall have explicit business process status information as meta-data content in the interoperability interface.

3.1.4 Change-management functionality category

Most interoperability interfaces in business will involve multiple transactions, with the potential for two-way flow, and the feedback of changes made one side of the interface to the other. If the content crossing the interoperability interface is subject to change then it is important that applications making changes behind the interface are also responsible for management of version and status meta-data. The following levels of change-management functionality are proposed:

(i) ***Export Level***: -The ability of a system to deliver compliant content across an interoperability interface, independent of data scope and interface technology.

(ii) ***Import Level***: -The ability of a system to receive compliant content across an interoperability interface, independent of data scope and interface technology.

(iii) ***Seeding Level***: - The ability of a system to persist imported data, independent of existing data, if any.

(iv) ***Consolidation Level***: -The ability of a system to consolidate and/or maintain relationships between added, deleted, modified data and relationships, on the basis of system *internal* keys and meta-data per 2.7 above.

(v) ***Reconciliation Level***: -The ability of a system, in addition to consolidate content based on internal keys and meta-data (iv above), to correlate and maintain *external* keys and meta-data (per 3.2.3 above).

3.2 Validating consistency and suitability

The final set of tests focus on validating the content and scope of the Templates and other data, and the suitability of the use case behind the data. These tests will (probably) always involve human interpretation the results from execution of the model data.

3.2.1 Validate use case consistency

In order to validate the functionality of a template entry for use in stated applications, the representation its information content should be evaluated against the following categories of questions

- Complete scope: Check relevant data library for missing templates relative to business needs.
- Complete content: Checkrelevant data library to make sure all required data objects exist.
- Complete reference: Check reference from relevant sources for all required data objects.
- No overlap: Check to avoid overlap with existing reference data or existing domain concepts.
- Domain expert help: Assistance from domain experts to ensure correct terms and definitions.
- Admin expert help: Assistance from standards experts to maintain (the status of) data items.

3.2.2 Validate business suitability

In order to validate suitability of a template entry for business usage, the representation of information should be evaluated to establish its contribution to achieving metrics for use, reuse, timesaving and quality improvement

- Creation benefit: Link between established KPIs and addition of new template instances.
- Usage benefit: Link between established KPIs and use of existing template instances.
- Usage consistency: Consistency of data usage between template populations used across applications and projects.
- Cost effectiveness: Comparison of cost for template development and operation against perceived and achieved benefit.

4 Assessing conformance and compliance of the PCA Templates

This chapter contains an overview assessment of conformance and compliance of the current Template implementation in the PCA RDL (note that this evaluation was defined for assessing software interfaces, and that some of the criteria may not be useful for the current evaluation of RDL Template content).

4.1.1 Verifying technical requirements

Semantic Modeling

The various templates listed in chapter 2 are all represented as instances of the ISO 15926 Part 2 entity type Class of Multidimensional Object, and all individuals created from these Templates to describe individuals therefore become instances of Multidimensional Object.

For all templates in the PCA RDL there is also a defined class hierarchy of super and subclasses, as well as defined membership for instances of these templates covering the full set of relationships in the ISO 15926 Part Data Model.

The defining axioms of each template is listed in Prover9 syntax, but without full representation of the underlying Data Model entities (Relationships and Roles). Thus, the templates in the PCA RDL have conformance at the ***Short-Cut Relations Level***.

Referencing Technology

All reference data items have URI's, which require access to on-line (or cached) reference data libraries in order for the reference data items to be resolved. Thus, the templates in the PCA RDL have conformance at the ***URI Referencing Level***.

Representation Technology

All Templates in the PCA RDL have been implemented using an (XML-serialized) RDF graphs. Thus, the Template representation in the PCA RDL has conformance at the ***RDF/OWL Schema Level***.

Interface Technology

Through the PCA RDL endpoint, (http://posccaesar.org/endpoint) user may use SPARQL to query for any set of data items, and see the results as Linked Pages, or in Turtle or RDF/XML format. Thus, the PCA RDL hasconformance at the ***SPARQL Query Level.***

4.1.2 Validating business compliance

Compliance of Templates must be validated against usage by end-users and applications to solve "real-world" problems. So far, the various Templates in the PCA RDL have not yet been used to a sufficient degree to allow meaningful validation, and the assessment of maturity levels given below is only meant to be indicative and informative as input to required future work.

Industrial standardization

All of the templates implemented in the PCA RDL use only data items from the PCA RDL itself: all Roles and Parameters are defined as reference data and the template implementation has maturity at the ***PCA/JORD Level.***

Payload Content

The templates in the PCARDL are all basic Core, Base or Proto Templates, which represent generic information, and the PCA RDL is not specific for any particular type of payload content and has maturity at the ***Generic Level***.

Change Management Meta-data

Template representation in the PCA RDL includes ISO 15926-2 Entity Type (Class of Multi-dimensional Object), standard meta-data (status, note, definition, creation date and creator) Roles, Superclasses, Subclasses and Classifiers. However, the meta-data applies to the data item as such and has no specific mechanisms for any particular version or status of the data item. Thus, the PCA RDL is at the ***Identity Only Level.***

Change Management Functionality

The interfaces to the PCARDL (and the P8 endpoint) includes facilities for both importing data (through either SPARQL Update or endpoint load) and exporting data (through XML files), but has no particular functionality for populating empty instances with new data or reconciliation of external identifiers. The PCA RDL maturity can be said to be at a (tentative) ***Export Level*** and ***Import Level***.

4.2 Summary checklist of maturity levels

The JORD Compliance Specificationprovides a checklist against which technical conformance and business compliance can be assessed. The checklist example in figure 3.3 below summarizes the above verification and validation of the representation of Templates in the PCA RDL.

Compliance Categories		Compliance Levels per Compliance Specification	*MATURITY LEVEL CHECKLIST SUMMARY*	User Required	Provider Claimed	JORD Validated
Technical	*Semantic Modeling*	2.1 (i)	**Dictionary & Typing Level** - Identification, Specialization & Classification template signatures only.			
		2.1 (ii)	**Short-Cut Relations Level** - As Dictionary Level plus *CoRwS* or other (eg *Gellish*) "Short-Cut" template signatures.		X	
		2.1 (iii)	**Full Ontology Level** - Any / all valid template signatures supported.			
	Referencing Technology	2.2 (i)	**Local Naming Level** - RD URI's resolved and naming self-contained in schema representation.			
		2.2 (ii)	**URI Reference Level** - Dependency on RD URI's being resolvable.		X	
	Representation Technology	2.3 (i)	**No Explicit XML Schema Level** - Implicit / document / formatted / tabular / non-XML schema.			
		2.3 (ii)	**Explicit XML Schema Level** - registered XML Schema			
		2.3 (iii)	**RDF/OWL Schema Level** - eg Part 8		X	
	Interface Technology	2.4 (i)	**File Exchange Level**			
		2.4 (ii)	**API or Query Level** - other than Part 9 / SPARQL			
		2.4 (iii)	**SPARQLQuery Level** - eg Part 9 Façade		X	
Business	*Industrial Standardization*	2.5 (i)	**Local Sandbox Level** - Community or individual organization with no externally certified RDL management.			
		2.5 (II)	**Global Industrial Level** - externally certified RDL			
		2.5 (III)	**PCA/JORD Level**		X	
		2.6 (iv)	**ISO Level**			
	Payload Content	2.6 (i)	**Generic Level** - Tool capability independent of payload.		X	
		2.6 (ii)	**Explicit Scope Level** - Scope per BIDG or otherwise defined			
	Change-Management Meta-Data	2.7 (i)	**Identity Only Level** - all data elements & sets identfiable / explicitly addressable		X	
		2.7 (ii)	**Version Level** - identification of succeeding / superceding versions of data elements & sets explicit			
		2.7 (iii)	**Status Level** - business status explicitly attributed / associated with each identified & versioned data element & set.			
	Change-Management Functionality	2.8 (i)	**Export Level** - Component interface publishes or permits read / query of internal content		(X)	
		2.8 (ii)	**Import Level** - Component interface accepts write to internal content, or reads external content.		(X)	
		2.8 (iii)	**Seeding Level** - Component populates empty instance with imported content losslessly			
		2.8 (iv)	**Consolidation Level** - Component populates existing instance with new imported content losslessly, correctly handlng versions and consolidating duplicates.			
		2.8 (v)	**Reconciliation Level** - Component maintains reconcilliation of external identifiers when updating existing instance internally.			

Figure 3.3: Summary checklist of maturity levels for Templates in the PCA RDL

5 Enhancing conformance and compliance

This chapter lists“tools”forchecking and improving theconformance of reference data and compliance of templates. These tools use SPARQL Query for exploring the RDL Data and Template content, SPARQL Insert for adding content, and SPARQL Update to fix and enhance content.

5.1 Discovering errors and omissions in data

In order to search for incorrect and missing data items SPARQL Select queries can be used to

- Select classes missing or incorrect superclass
- Select classes with inconsistent classifiers
- Select classes with incomplete meta-data

Appendix A gives a set of examples of relevant queries.

These queries can be set up to run automatically, and report all errors in an auto-generated machine-readable (summary and detailed) online verification reports. Thereport system can distinguish between “hard” and “soft” errors (where only the former reports trigger alarms).

5.2 Adding new data

In order to add new templates (and other data entities) we can use SPARQLInsert statements to create Templates (Class of Multidimensional Object), Template Instances (Multidimensional Object), Roles (Role) and Parameters (specified by role restrictions).

- Insert classes and meta-data (including Templates and Roles)
- Insert relationships between specified classes

Appendix B contains a set of examples of relevant queries.

5.3 Fixing errors in existing data

In order to fix (update and upgrade) existing templates, we can use SPARQL Update, with a combination of Delete and Insert queries.

- Update inconsistent sets of Templates relative to specifications
- Updateinconsistent and missing Role names and restrictions
- Update Cardinality of Roles
- Updateinconsistent and missing meta-data

Appendix C contains a set of examples relevant queries.

Appendix A: SPARQL queries to find incorrect and missing data

The main SPARQL endpoint can be found at http://posccaesar.org/endpoint/
The new SPARQL endpoint can be found at http://data.posccaesar.org/rdl/
-- --

The set of predefined namespaces:
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX fn: <http://www.w3.org/2005/xpath-functions#>
PREFIX afn: <http://jena.hpl.hp.com/ARQ/function#>
PREFIX p2: http://rds.posccaesar.org/2008/02/OWL/ISO-15926-2_2003#
PREFIX pca: <http://posccaesar.org/rdl/>
(use /dev/ for development, /stg/ for staging instead of /rdl/)
-- --

A.1 Find all data items without P2 entity type

```
SELECT ?rds
WHERE
{
FILTER NOT EXISTS { ?rds a ?part2Entity }
}
```
-- --

A.2 Find all data items without definition

```
SELECT ?rds ?label
WHERE
{
?rds a ?part2Entity;
rdfs:label ?label .
FILTER NOT EXISTS {?rds RDL:hasDefinition ?y }
}
```
-- --

A.3 Find all classes without superclass

```
SELECT ?rds ?label
WHERE
{
?rds rdfs:label ?label
FILTER NOT EXISTS {?rel p2:hasSubclass ?rds }
}
```
-- --

A.4 Find all instances of Class of Multidimensional Object

```
SELECT ?rds ?label
WHERE
{
?rds a p2:ClassOfMultidimensionalObject; rdfs:label ?label .
?rel p2:hasSuperclass ?rds .
}
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

A.5 Find all classes that have different entity type than the superclass

```
SELECT ?subclass ?sublabel ?subentity ?superclass ?superlabel ?superentity
WHERE
{
?subclass rdfs:label ?sublabel; a ?subentity .
?superclass a ?superentity; rdfs:label ?superlabel .
?rel p2:hasSubclass ?subclass; p2:hasSuperclass ?superclass .
FILTER (?subentity != ?superentity)
}
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

Appendix B: SPARQL queries to add new data

The main SPARQL endpoint can be found at http://posccaesar.org/endpoint/
The new SPARQL endpoint can be found at http://data.posccaesar.org/rdl/
-- --

The set of predefined namespaces:
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX fn: <http://www.w3.org/2005/xpath-functions#>
PREFIX afn: <http://jena.hpl.hp.com/ARQ/function#>
PREFIX p2: http://rds.posccaesar.org/2008/02/OWL/ISO-15926-2_2003#
PREFIX pca: <http://posccaesar.org/rdl/>
(use /dev/ for development, /stg/ for staging instead of /rdl/)
-- --

B.1 Insert single class with id, designation and metadata

```
INSERT DATA { GRAPH http://posccaesar.org/ontologies/devrdl   {
        pca:PUMP a p2:ClassOfInanimatePhysicalObject ;
        rdfs:label $label ;
        pca:hasCreationDate $date ;
        pca:hasCreator $creator ;
        pca:hasDefinition $definition ;
        pca:hasDesignation $designation ;
        pca:hasIdPCA $pcaid ;
        pca:hasNote $note ;
        pca:hasStatus $status . } }
```
-- --

B.2 Insert Specialization relationship between specified classes

These functions connect new relationships between existing classes

```
 INSERT DATA { GRAPH <http://posccaesar.org/ontologies/pcardl> {
pca:RDSR1211 a p2:Specialization ;
pca:hasIdPCA "RDSR1211" ;
p2:hasSubclass pca:RDS11 ;
p2:hasSuperclass pca:RDS12 . } }
```
-- --

B.3 Insert Classification relationship between specified classes

```
INSERT DATA { GRAPH <http://posccaesar.org/ontologies/devrdl> {
pca:RDS7951704
a p2:Classification ;
pca:hasIdPCA "RDS7951704" ;
p2:hasClassified pca:RDS1193984 ;
p2:hasClassifier pca:RDS7950676 .} }
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

Appendix C: SPARQL queries to fix existing data

The main SPARQL endpoint can be found athttp://posccaesar.org/endpoint/
The new SPARQL endpoint can be found athttp://data.posccaesar.org/rdl/
-- --

The set of predefined namespaces:
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX fn: <http://www.w3.org/2005/xpath-functions#>
PREFIX afn: <http://jena.hpl.hp.com/ARQ/function#>
PREFIX p2: http://rds.posccaesar.org/2008/02/OWL/ISO-15926-2_2003#
PREFIX pca: <http://posccaesar.org/rdl/>
(use /dev/ for development, /stg/ for staging instead of /rdl/)
-- --

C.1 Modify data (and/or metadata) of a given class

This function changes the status of a specified class from its current value to status RETIRED. Note that this change makes use of a delete and insert command (and that the same commands can be used to change any data item).

```
WITH <http://posccaesar.org/ontologies/devrdl>
DELETE { ?id  pca:hasStatus ?status  }
INSERT { ?id  pca:hasStatus "Retired" }
WHERE { ?id pca:hasIdPCA "d" ; pca:hasStatus ?status }
```
-- --

C.2Change the name (designation and label) of a given class

```
WITH <http://posccaesar.org/ontologies/devrdl>
DELETE { ?id pca:hasDesignation ?designation; rdfs:label ?label }
INSERT { ?id pca:hasDesignation $designation; rdfs:label $label}
WHERE { ?id pca:hasIdPCA $pcaid; pca:hasDesignation ?designation; rdfs:label ?label}
```

We can use an input variable only once, and have to explicitly add designation and label (even if identical).
-- --

C.3 Change the note of a given class

```
WITH <http://posccaesar.org/ontologies/devrdl>
DELETE { ?id pca:hasNote ?note }
INSERT { ?id pca:hasNote $note }
```

```
WHERE { ?id pca:hasIdPCA $pcaid; pca:hasNote ?note }
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

C.4 Retire class by changing Status variable

```
WITH <http://posccaesar.org/ontologies/devrdl>
DELETE { ?id pca:hasStatus ?status }
INSERT { ?id pca:hasStatus $status }
WHERE { ?id pca:hasIdPCA $pcaid; pca:hasStatus ?status }
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

C.5 Delete Status of specified class

```
WITH <http://posccaesar.org/ontologies/devrdl>
DELETE { ?id pca:hasStatus ?status }
WHERE { ?id pca:hasIdPCA $pcaid; pca:hasStatus ?status }
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

C.6 Delete single class with ID, Designation and Metadata

```
WITH <http://posccaesar.org/ontologies/devrdl>
DELETE { ?rd ?property ?value }
WHERE { ?rd ?property ?value ; pca:hasIdPCA $pcaid}
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
```