OSCD Design Document

This document contains information on the development process of the **O**ntology for **S**ource **D**ata **C**hange (**OSCD**).

1. Tasks

You are asked to complete the following tasks:

1. Review the feedback questionnaire which will provide an indication of the requested feedback: https://forms.gle/gCSHZn79eTAFVCrz8
2. Review this document while considering the design methodology followed by OSCD in comparison to the state of the art.
3. Please complete the feedback questionnaire after you have reviewed the document.

Contact Alex Randles (alex.randles@adaptcentre) if you have any questions.

2. Design Methodology

The design of the OSCD followed best practices as recommended by the semantic web community. Ontology design practices were reused from the most prominent ontology design methodologies. The methodologies included the NeON methodology [19], UPON Lite [8], Ontology development 101: A guide to creating your first ontology [9] and LOT: An industrial oriented ontology engineering framework [12].

1. **Identification of aims, objectives, scope:** The design process commenced with the identification of the aims, objectives and scope of the ontology, which are outlined in Table 1 of this document. The template used for the table was retrieved from the methodologies and used to define the ontology requirements specification document. The document outlines requirements and among other things, the aims, objectives and scope of the ontology.
2. **Identify and analyze relevant information:** A review of publications in the state of the art was conducted to identify relevant information. Publications within the state of the art which related to topics within the defined scope were reviewed to facilitate the retrieval of relevant information. Thereafter, the retrieved information was used to formalize competency questions. Table 2 includes references to publications which inspired the creation of each competency question.
3. **Create use-cases and competency questions:** Competency questions were created during the design process of the ontology. The questions define the functional requirements of the ontology and were iteratively refined until an accurate representation of the requirements and objectives was conceived. The final iteration of the questions is

shown in Table 2. Use cases were devised in order to refine the requirements of the ontology and were retrieved from the RML test case files[1]. The test case files provided a diverse set of source data in formats such as XML, JSON, relational databases and CSV as well as respective RML mappings. In addition, the R2RML test case files[2] were used, however, the source data is only represented in relational format. The test cases facilitated the creation of use cases through the generation of graphs defined in OSCD when changes were detected between the file versions. The use case has been documented in a publication [14]. In addition, the ontology was applied to a network management use case [15]. A use case graph generated by the RML test cases is available (https://tinyurl.com/hyrmr9aa).

4. **Identify concepts and relationships:** Concepts and relationships were identified through the state of the art review and the researchers previous experience in the creation of linked data (LD). The concepts and relationships were iteratively defined until the information modeling provided by the ontology satisfied each of the competency questions. In addition, concepts and relationships were reused from existing vocabularies as recommended by the methodologies and the W3C recommendation on Data on the Web Best Practices [6]. **Reused ontologies** included an ontology for Linking Open Descriptions of Events (LODE) [17] which is designed to model events. LODE was extended to model changes as specialized events which have occurred in source data. The Rei ontology [3] is designed to model policies for various domains and was reused to represent the details of the notification policy, which is used to inform maintainers of changes. The DUL ontology [1] was reused similar to LODE to represent the agents involved in the activities. The reuse in OSCD is demonstrated in the competency questions and ontology documentation.
5. **Progressive iterations:** Steps 2-4 were iteratively repeated until the point when the proposed concepts/relationships provided information which satisfied each requirement defined in the form of a competency question.
6. **Create Ontology:** The ontology was implemented in OWL 2 Web Ontology Language [7]. Concepts and relationships which were defined in the previous step were constructed using Protégé ontology development tool [24]. In addition, semantic reasoners were also utilized to detect and remove logical inconsistencies within the ontology.
7. **Evaluate:** The ontology was evaluated for sufficiency to provide information to fulfill each competency question. The usage of a semantic reasoner within Protege ensured logical inconsistencies were identified. In addition, OOPS! Pitfall Scanner [13] was used to detect common ontology design issues. The quality of metadata and documentation was evaluated through presentation in peer reviewed publications. Feedback received from reviewers allowed areas for improvement to be identified. Peer reviewed publications related to OSCD are outlined in Section 5. Further expert feedback was received in a previous user evaluation where they were asked to provide feedback on the design and application of the ontology. In addition, the ontology was presented to a

[1] https://rml.io/test-cases/
[2] https://www.w3.org/2001/sw/rdb2rdf/test-cases/

panel of experts at the semantic interoperability conference (SEMIC2022)[3] organized by the European commission. Each graph generated in the use cases were assessed with the RDFUnit [5] quality assessment framework which provides test driven validation of RDF data.

8. **Publication:** Ontology documentation (https://w3id.org/OSCD) was created using WIDOCO [2] which is a tool designed to use ontology metadata to create HTML documents listing its classes and properties. Thereafter, the ontology and human readable documentation were published with a permanent identifier as a FAIR resource including an open and permissive license. The documentation contains information about the creation, design, usage, class interaction diagrams and provides various serializations.

3. Background

The following section provides information related to the requirements, design and purpose of OSCD.

3.1 Description

OSCD provides an ontology for expressing information related to changes which occur in source data used by mappings to create a LD dataset. The information will allow notification of changes to data maintainers which will enable them to make appropriate changes to the mappings and data in order to maintain alignment between them. In addition, it is hoped the information will benefit the maintenance and reuse of mappings. The ontology was designed to resolve the gap in the state of the art in relation to an ontology which represents information related to changes in the source data of LD mappings.

3.2 Requirements

The development of the ontology follows best practices in ontology development methodologies, such as those mentioned. Creating a specification for the ontology provided additional guidance during the development phase. The requirements have been derived from state of the art review and application of the ontology within a framework and use cases. **Table 1** shows the requirements document for OSCD.

Table 1: Ontology requirements specification document [20]

Ontology Requirements Specification Document	
1	**Purpose**

[3] https://joinup.ec.europa.eu/collection/semic-support-centre/semic-conference

	Capturing information related to changes which have occurred in the source data used by mappings to produce LD. The information is expected to positively impact the quality of mappings and datasets by preserving alignment with the underlying data sources as well as facilitate the reuse and maintenance of those mappings.
2	**Scope**
	In scope: • Source data used by mappings • Changes within source data • Agents related to the source data • Notifications of changes **Out of scope:** • Actions to improve alignment • Resulting published dataset
3	**Implementation Language (optional)**
	OWL 2 Web Ontology Language
4	**Intended End-Users (optional)**
	Agents involved in the transformation of data into LD representation.
5	**Intended Uses**
	Capturing information associated with the agents and activities involved in the generation and maintenance of LD.
6	**Ontology Requirements**
	1. Non-Functional Requirements

	Allow the users of the ontology to identify and be notified of changes which have occurred in the source data of LD mappings.
	2. Functional Requirements: Lists or tables of requirements written as Competency Questions and sentences
	Competency questions in Section 4 (next section).
7	**Pre-Glossary of Terms (optional)**
	1. Terms from Competency Questions
	Competency questions in Section 4.
	2. Terms from Answers
	Competency questions in Section 4.
	3. Objects
	Competency questions in Section 4.

4. Competency Questions

Ontology Competency questions define design requirements in natural language form. These questions state information which should be provided by the ontology. The fulfillment of the questions is accomplished by providing a concept/relationship which represents the required information. Most questions were inspired by literature discovered in the state of the art review. However, certain questions were defined through application to use cases (**DTA**) and feedback from the experts (**DTF**). The answers to each question are structured as <Subject, Relationship, Concept> which represent an RDF triple. **Table 2** shows the final iteration of competency questions created for OSCD.

The **answer** to each question is structured as <Subject, Relationship, Concept> which represent an RDF triple. A description of each concept and relationship used is available[4].

Table 2: OSCD Competency Questions

#	Question	Relationship	Concept	References
Subject: cdo:ChangeLog A grouping of changes which have occurred in a source data.				
1	Who maintains the source data?	oscd:hasMaintainer	dul:Agent	[11,17,21]
2	What data is represented in the source data?	oscd:hasCurrentSource oscd:hasPreviousSource	rdfs:Resource	DTA
3	What changes were detected in the source data?	oscd:hasChange	oscd:InsertSourceData …… cdo:DeleteSourceData	[4,11,16,23,25]
4	What notification policies are associated with the source data?	oscd:hasNotificationPolicy	rei-policy:Policy	[10,11,21,23,25]
5	What thresholds are associated with the notification policy?	oscd:hasThrehsold	xsd:integer	[10,11]
6	When did the change detection process begin?	oscd:hasDetectionStart	xsd:dateTime	[10,11,18,22]
7	When did the change detection process finish?	oscd:hasDetectionEnd	xsd:dateTime	[10,11,18,22]
Subject: oscd:[<Action>]SourceData The change which has occurred. <Action> represents one of the change types.				
8	Who is responsible for the change?	oscd:wasChangedBy	dul:Agent	DTF
9	What data was changed as a result of a specific change?	oscd:hasChangedData	xsd:string	[4,11,18,21,22]
10	When did the change occur?	lode:atTime	xsd:dateTime	[4,11,17]
11	What was the original value?	oscd:hasPreviousValue	xsd:string	DTF
12	What data is associated with the change?	oscd:hasStructuralReference, oscd:hasDataReference	xsd:string	DTA

[4] https://drive.google.com/file/d/12tQrdzb3xKDmt2VTlKhIzs9Y5FWtkhew/view?usp=share_link

SPARQL query answers to the competency questions are available[5]. Further information on the graph used to execute the queries can be found in the “Description” section of the ontology documentation.

5. Publications

The following peer reviewed publications related to the design and usage of OSCD[6].

1) *Randles, A. and O'Sullivan, D., 2022, October. Modelling & Analyzing Changes within LD source data. In MEPDaW 2022-8th Workshop on Managing the Evolution and Preservation of the Data Web.*

In this publication we presented a description of the design process followed by OSCD. Furthermore, we outlined the application of the ontology within a use case and our quality improvement framework, named the Mapping Quality Improvement (MQI) framework.

2) *Randles, A., O’Sullivan, D., Keeney, J. and Fallon, L., 2022, September. Applying a Mapping Quality Framework in Cloud Native Monitoring. In Proceedings of the 18th International Conference on Semantic Systems (SEMANTiCS).*

In this publication we briefly described the application of OSCD in a network management use case. The change detection component of the MQI framework was applied to data which represented metrics used in network management.

Bibliography

[1] Victor Charpenay, Sebastian Käbisch, and Harald Kosch. 2016. Introducing Thing Descriptions and Interactions: An Ontology for the Web of Things. In *SR+ SWIT@ ISWC*, 55–66.

[2] Daniel Garijo. 2017. WIDOCO: a wizard for documenting ontologies. In *International Semantic Web Conference*, 94–102.

[3] Lalana Kagal and others. 2002. Rei: A policy language for the me-centric project. (2002).

[4] A. M. Khattak, Z. Pervez, W. A. Khan, A. M. Khan, K. Latif, and S. Y. Lee. 2015. Mapping evolution of dynamic web ontologies. *Inf. Sci. (Ny).* 303, (May 2015), 101–119. DOI:https://doi.org/10.1016/j.ins.2014.12.040

[5] Dimitris Kontokostas, Patrick Westphal, Sören Auer, Sebastian Hellmann, Jens Lehmann, Roland Cornelissen, and Amrapali Zaveri. 2014. Test-driven evaluation of Linked Data quality. In *WWW 2014 - Proceedings of the 23rd International Conference on World Wide Web*, Association for Computing Machinery, Inc, 747–757. DOI:https://doi.org/10.1145/2566486.2568002

[6] Bernadette Farias Lóscio, C Burle, and N Calegari. 2016. Data on the web best practices. W3C recommendation. *World Wide Web Consort.* (2016). Retrieved from https://www.w3.org/TR/dwbp/

[7] Deborah L McGuinness, Frank Van Harmelen, and others. 2004. OWL web ontology

[5] https://drive.google.com/file/d/1rd-xMDkMcPfPry28vmG4MsKuvnXjb6xn/view?usp=share_link

[6] As a note the ontology was previously called the Change Detection Ontology (CDO) in the publications

language overview. *W3C Recomm.* 10, 10 (2004), 2004.

[8] Antonio De Nicola and Michele Missikoff. 2016. A Lightweight Methodology for Rapid Ontology Engineering. *Commun. ACM* 59, 3 (February 2016), 79–86. DOI:https://doi.org/10.1145/2818359

[9] Natalya F Noy, Deborah L McGuinness, and others. 2001. Ontology development 101: A guide to creating your first ontology.

[10] Alexandre Passant and Pablo N Mendes. *sparqlPuSH: Proactive notification of data updates in RDF stores using PubSubHubbub*. Retrieved from http://www.ldodds.com/blog/2010/04/rdf-dataset-notifications/

[11] Niko Popitsch and Bernhard Haslhofer. 2011. DSNotify - A solution for event detection and link maintenance in dynamic datasets. *J. Web Semant.* 9, 3 (September 2011), 266–283. DOI:https://doi.org/10.1016/j.websem.2011.05.002

[12] María Poveda-Villalón, Alba Fernández-Izquierdo, Mariano Fernández-López, and Raúl García-Castro. 2022. LOT: An industrial oriented ontology engineering framework. *Eng. Appl. Artif. Intell.* 111, (2022), 104755. DOI:https://doi.org/https://doi.org/10.1016/j.engappai.2022.104755

[13] María Poveda-Villalón, Asunción Gómez-Pérez, and Mari Carmen Suárez-Figueroa. 2014. OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation. *Int. J. Semant. Web Inf. Syst.* 10, 2 (2014), 7–34.

[14] Alex Randles and Declan O'Sullivan. 2022. Modeling & Analyzing Changes within LD Source Data. In *8th Workshop on Managing the Evolution and Preservation of the Data Web (MEPDaW) co-located with the 21st International Semantic Web Conference (ISWC 2022)*, Virtual Event.

[15] Alex Randles, Declan O'Sullivan, John Keeney, and Liam Fallon. 2022. Applying a Mapping Quality Framework in Cloud Native Monitoring. In *18th International Conference on Semantics Systems (SEMANTiCS)*.

[16] Julio Cesar Dos Reis, Cédric Pruski, Marcos Da Silveira, and Chantal Reynaud-Dela\^\itre. 2015. DyKOSMap: A framework for mapping adaptation between biomedical knowledge organization systems. *J. Biomed. Inform.* 55, (2015), 153–173.

[17] Ryan Shaw, Raphaël Troncy, and Lynda Hardman. 2009. Lode: Linking open descriptions of events. In *Asian semantic web conference*, 153–167.

[18] Anuj Singh, Rob Brennan, and Declan O'Sullivan. *DELTA-LD: A Change Detection Approach for Linked Datasets*.

[19] Asunciónand Fernández-López Mariano Suárez-Figueroa Mari Carmenand Gómez-Pérez. 2012. The NeOn Methodology for Ontology Engineering. In *Ontology Engineering in a Networked World*, Asunciónand Motta Enricoand Gangemi Aldo Suárez-Figueroa Mari Carmenand Gómez-Pérez (ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 9–34. DOI:https://doi.org/10.1007/978-3-642-24794-1_2

[20] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, and Boris Villazón-Terrazas. 2009. How to write and use the ontology requirements specification document. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, 966–982.

[21] J ¨ Urgen Umbrich, Michael Hausenblas, Aidan Hogan, Axel Polleres, and Stefan Decker. *Towards Dataset Dynamics: Change Frequency of Linked Open Data Sources*. Retrieved from http://code.google.com/p/pubsubhubbub/

[22] Vânia Vidal, Narciso Arruda, Matheus Cruz, Marco Casanova, Carlos Brito, and Valéria Pequeno. 2017. Computing changesets for RDF views of relational data. In *Workshop on Managing the Evolution and Preservation of the Data Web (MEPDaW 2017)*, 43–58.

[23] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. 2009. *Discovering and Maintaining Links on the Web of Data*. Retrieved from http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/

[24] Protégé. Retrieved January 1, 2022 from https://protege.stanford.edu/
[25] 2010. Dataset Dynamics (dady) Vocabulary. Retrieved June 14, 2022 from http://purl.org/NET/dady