Towards semantic representation of machine-actionable Data Management Plans

João Cardoso1, Leyla J. Garcia3, Fajar J. Ekaputra2, Marie C. Jacquemot4, Tomasz Miksa2, and José Borbinha1

1 Universidade de Lisboa, Instituto Superior Técnico & INESC-ID, Lisboa, Portugal
2 TU Wien & SBA Research, Vienna, Austria
3 ZB MED Information Centre for Life Sciences, Cologne, Germany
4 DMP OPIDoR & Inist-CNRS, Nancy, France

Abstract. The concept of Data Management Plan (DMP) has emerged as a fundamental tool to help researchers through the systematical management of data. The Research Data Alliance DMP Common Standard (DCS) working group developed a core set of universal concepts characterising a DMP, in the pursuit of having making a DMP as a machine-actionable information artefact. This paper reports on the community effort to create the DMP Common Standard Ontology (DCSO), a serialisation of the DCS core set of universal concepts, giving particular focus to detailing the components of that ontology.

Keywords: Data Management Plan · Machine-Actionable Data Management Plan · Semantic Technology.

1 Introduction

With ever growing quantities of research data, researchers are now faced with the challenge of systematically managing large quantities of data. Data Management Plans (DMP) have emerged as a tool to aid researchers in tackling this challenge. The DMP, produced as a document, describes techniques, methods and policies on how data is to be managed throughout the data life cycle [21]. Additionally, it also establishes associations between data management activities and the actors responsible for their execution [10]. The concept of DMP has evolved into that of the machine-actionable DMP (maDMP), which aims to address some of the limitations that have been identified with the DMP. [15]. For example the level of details of a DMP can vary according to the design choices and awareness of its creators. Additionally having information expressed in free-form text can lead to the DMP not having the necessary detail, thus offering less accurate information. The main goal of maDMP is to have DMP represented in a format that makes its information consumable by both humans and automated systems. This machine-actionable representation will allow: for the maDMP to be used as an interchange format between automated systems; to be fully integrated into
existing data management workflows; and to support automated machine-process pipelines regarding data management policies.

The production of a machine-actionable representation of a DMP, implies the necessity for standardisation. To tackle that issue, the Research Data Alliance (RDA)\textsuperscript{5} DMP Common Standards (DCS) working group created a set of universal elements characterising a DMP \cite{12, 11, 13}. The next step of the DCS working group was to provide reference serialisations of this DCS core set of terms. At the time of release, the DCS core set of terms was accompanied by a JSON serialisation\textsuperscript{6}, with the responsibility of developing other serialisation formats left to the community.

This paper reports on the creation of the DMP Common Standard ontology (DCSO), a community effort to a semantic technology based serialisation of the DCS set of universal terms, and details the resulting ontology and its components.

The remainder of this paper is organised as follows. Section 2 reports on the creation of the DCSO, with particular focuses on the requirements and methodology followed to achieve that goal. Section 3 introduces DCSO core together with an extended model as well as a validation layer. Finally section 4 offers a summary of what was achieved and lists a set of future goals.

2 The ontology engineering process

One of the primary goals of the DCS working group was to establish a core set of terms that characterised a DMP. The intention was that this set of term would form an application profile, which by definition is a metadata design specification that uses a selection of terms from multiple metadata vocabularies, with added constraints, to meet application-specific requirements\textsuperscript{7}. However, the DCS application profile is still a work in progress, for not all of the concepts selected to characterise a DMP, are associated with established metadata vocabularies.

Nonetheless, providing serialisations for the DCS core set of terms was one of goals listed in the initial proposal of the DCS working group. The motivation behind this goal was to allow any tools or systems engaged in research data processing not only to consume data but also to add data to maDMPs, thus automated data interchange.

The initiative of creating the DMP Common Standard Ontology (DCSO), is to provide a serialisation of the DCS core set of terms. This serialisation is not intended to be a replacement of existing serialisations, but to add a serialisation with distinct features to the existing set of serialisations. The DCSO represents information using semantic technologies, (e.g., standardisation and interoperability), which are in agreement to the advantages pursued by the concept of maDMP. Ontologies allow for the representation of a shared conceptualisation of knowledge through the usage of formal semantics \cite{19}. However, despite their

\textsuperscript{5} https://www.rd-alliance.org/rda-europe
\textsuperscript{6} https://github.com/RDA-DMP-Common/RDA-DMP-Common-Standard/tree/master/examples/JSON
\textsuperscript{7} https://www.dublincore.org/resources/glossary/application_profile/
traditional perception of a highly formal means of knowledge representation, ontologies and have been proven to be suitable for the creation of Linked Open Data (LOD) [1, 6]. Ontologies are by design easy to extend, as concepts can be matched or relations can be established between ontologies covering distinct domains. This characteristic enforces the suitability of ontologies for the representation of the DCS core set of terms, for it is also designed with modularity in mind. Additionally, ontologies enable reasoning, and thereby knowledge inference from the information explicitly represented [9]. It is important to mention that throughout the process of creating the DCSO, none of the established ontology engineering methodologies for the creation of ontologies [14, 20, 8] were applied in full. The methodology for the creation of the latest version of the DCSO (version 3.0.2) is described in section 2.2, and draws inspiration from the established methodologies in ontology creation and integration.

2.1 DCSO version 2.0.2

The first attempts to develop a stable version of the DCSO resulted in DCSO version 2.0.2. While it was a good starting point, this version had several major issues described hereafter.

Constraints. The DCS core set of terms defines multiplicity and type constraints. In the DCSO version 2.0.2 these constraints were represented through the usage of OWL constraints. This is a viable means of constraint representation. However, OWL does not allow to check that data is compliant with specification and therefore, validation rules should be expressed using another language, such as Shapes Constraint Language (SHACL) [7] or Shape Expressions (ShEx) [2].

Controlled vocabularies. The DCSO version 2.0.2 defines a set of custom literal datatypes to accommodate the usage of several controlled vocabularies, (see section 3.2). This was a deliberate choice, made out of a necessity to simplify the creation process. However, defining controlled vocabularies though custom literal datatypes does not follow the best practices in ontology engineering, and furthermore, does not add value to the ontology.

Ontology Reuse. In the process of creating the DCS application profile, number of terms and fields from standardised vocabularies were specified, (e.g., DCAT and DC). However in DCSO version 2.0.2 these terms and fields are redefined, as opposed to the best practice of reusing terms by means of integration of existing domain ontologies. This diminishes the overall value of the DCSO version 2.0.2.

Non persistent namespaces. In the DCSO version 2.0.2, the namespace URI is not persistent. The ontology uses the link to the GitHub repository as the
ontology namespace. To this end, the current state of practice is to use W3ID\textsuperscript{13} and redirect it to the hosting location of the ontology.

### 2.2 DCSO version 3.0.2

Whilst version 2.0.2 of the DCSO proved to be an adequate starting point, it was evident that a new version that followed the best practices in ontology engineering was necessary. The opportunity to materialise this new version presented itself through the RDA Hackathon on Machine-Actionable Data Management Plans\textsuperscript{14}. The objective of the Hackathon was to promote the usage of the maDMP concept by the research community. Participants were encouraged to submit topics and assemble teams that would collaborate for two days in order to tackle those topics.

The creation of a version 3.0.2 of the DCSO\textsuperscript{15} was one of the topics submitted in the context of the RDA Hackathon on Machine-Actionable Data Management Plans. This version had to comply with the existing requirements (i.e., provide a serialisation of the DCS core set of terms), and address the identified issues. The development process was divided into three iterative stages.

**First stage.** The objective of this stage, was to first create an ontology that would fully serialise the DCS core set of terms, and would integrate terms from selected domain ontologies. It was decided not to reuse previously existing versions of the DCSO. The new version would be expressed using the Terse RDF Triple syntax (Turtle)\textsuperscript{16}. The outcome of this stage was the creation of the DCSO Core (see section 3.1).

**Second stage.** The next stage of the process was to incorporate the usage of controlled vocabularies into the existing ontology (see section 3.2). In parallel to this effort, a constraint validation layer was also developed using ShEx (see section 3.3).

**Third stage.** In the last stage of the process human-readable descriptions for all of the created resources were added (i.e., adding rdfs:comment descriptions to classes, data properties and object properties). Additionally the default namespace of the new version of the ontology was provided by the W3ID. This is in compliance with the best practices of having the namespace URI being a persistent identifier. Finally, the GitHub repository where the ontology is published was revised, its structure was adapted to suit the new version of the DCSO, and its documentation was created to facilitate community adoption.

### 3 The DMP Common Standard Ontology

This section details the results of the development process of the latest version of the DCSO. The DCSO is organised into two ontologies: (1) The DCSO Core; and

\textsuperscript{13} https://github.com/perma-id/w3id.org
\textsuperscript{14} https://github.com/RDA-DMP-Common/hackathon-2020
\textsuperscript{15} https://github.com/RDA-DMP-Common/RDA-DMP-Common-Standard/blob/master/ontologies/core/dcso.3.0.2.ttl
\textsuperscript{16} https://www.w3.org/TR/turtle/
(2) the DCSO Extensions (DCSX). The DCSO Core represents the core elements of the DCS core set of terms. It includes terms from the W3C DCAT Specification [5], DCMI Metadata Terms [3], and FOAF Vocabulary Specification 0.99 [4]. The DCSX is a support to the DCSO core. As it collects into a single ontology the controlled vocabularies that are in use in the DMP Common Standard.

### 3.1 DCSO Core

The DCSO core represents the core set of universal elements defined by the DCS characterisation of a DMP. The DCSO Core comprises of 26 classes, as can be seen in Figure 1a. Of the 26, 13 are one to one matches of the terms in the DCS core set of terms. The remainder 13 classes can be divided into two categories: (1) identifier classes; and (2) external classes.

![Diagram of DCSO Core](image)

![Diagram of DCSX](image)

**Fig. 1:** The class structure of the DCSO
The reasoning behind the identifier classes was to represent the multiple "_id" nested structures that are present in the DCS core set of terms. As such, all of the identifier classes, (e.g., deso:ContributorId, deso:DMPId, deso:DatasetId, etc.), are subclasses of the deso:Id class. All classes share the same data properties, (i.e., identifier and type), only diverging on the accepted values for each data property and respective property domain.

External classes are classes that are reused from imported ontologies. The DCSO Core imports 4 ontologies: (1) the W3C DCAT Specification: classes deso:Dataset and deso:Distribution are subclasses of dcat:Dataset and dcat:Distribution, respectively; (2) the FOAF Vocabulary Specification 0.99: classes deso:Contact and deso:Contributor are sub-classes of the foaf:Agent class; (3) the DCSX, (see section 3.2); and (4) the DCMI Metadata Terms, where both annotation and data properties were reused (e.g., terms:description, terms:title, terms:format, terms:issued, etc.);

In the DCS core set of terms, relations between concepts are not named, with only their multiplicity being declared. In the DCSO Core relations between classes are represented through object properties. Their names are always typed in camelCase, and comprise of the prefix "has" followed by the name of the class to which they pertain, as can be seen in Figure 2. Concepts in the DCS core set of terms also have multiple fields. These fields are represented in the DCSO Core as data properties. However, any field that complied with a standardised
controlled vocabulary was represented using individuals from the DCSX ontology (see section 3.2), with object properties being used to represent the relation. An example of this particular scenario can be found in Figure 2, where the individual \textit{dmp}\_1 of the class \textit{dcso:DMP} has a object property relation (\textit{hasLanguage}) with the individual \textit{en} of the class \textit{dcsx:Language} from the DCSX.

### 3.2 DCSX: DCSO Extensions

The DCS core set of terms specifies controlled vocabularies for some of its terms. The Host class is an example of this, with the geo\_location field describing the physical location of the data by using the ISO 3166-1 country code [17]. In early versions of the DCSO, controlled vocabularies were represented through the creation of custom Datatypes. However, this solution did not follow the best practices in ontology representation (see section 2).

The DCSX ontology was created to address the DCS core set of terms that require the usage of standardised controlled vocabularies. It comprises of a set of individuals associated with one of three classes, and the necessary object properties required to express the relations between the three classes and the DCSO Core.

The class structure of the DCSX can be seen in Figure 1b, with each of the classes representing a standardised controlled vocabulary. The \textit{dcsx:Country} class represents the ISO 3166-1 country codes, the \textit{dcsx:CurrencyCode} class represents the ISO 4217 currency codes [18], and the \textit{dcsx:Language} class represents the ISO 639-3 language codes [16]. All three controlled vocabularies were represented through the creation of a set of individuals associated with each respective class.

### 3.3 Constraints Validation Layer

The DCSO constraints validation layer makes it easier for users to realise whether or not their maDMP complies to the underlying model. It also contributes to data completeness and consistency. Currently, three validation approaches are prevalent: JSON schema is particularly useful for data expressed in JSON-LD while Shape Expression (ShEx) [2] and Shapes Constraint Language (SHACL) [7] can be used with other formats as well. None of the mentioned approaches is more broadly used than others when it comes to semantic data validation; therefore, choosing one or the other becomes more a personal choice. ShEx was selected for the DCSO constraints validation layer due to expertise and familiarity with the format. A ShEx schema establishes conditions regarding existence (e.g., mandatory or optional), cardinality (e.g., zero or one, one or more), and how elements should be combined with each other. The DCSO constraint validation layer ShEx schemas follow the guidelines provided by the DCS core set of terms. Two distinct schemas are provided. The first one targets the DMP together with elements regarding identifiers, contacts, contributors, costs and projects while the second one, still in progress by the time of writing, deals with datasets contained in the DMP.
4 Conclusions and Future Work

This paper reported on both the DCSO as well as the process that led to its creation. The DCSO is a semantic technology based serialisation of the DCS core set of terms, and is structured into two codependent ontologies: (1) The DCSO Core (see section 3.1), which is used to represent the core concepts of the DCS DMP characterisation, and reuses terms from established domain ontologies, in accordance to the overall objective of the DCS of creating an application profile (see section 2); and (2) The DCSX (see section 3.2), which provides support to the DCSO Core by providing representations of the three standardised controlled vocabularies referred in the DCS core set of terms. The last component of the DCSO that was reported is the constraints validation layer. It is a set of ShEx constraints that allow any DMP serialised with the DCSO to be validated against the DCS core set of terms.

The DCSO is now a matured ontology that follows the best practices in ontology engineering. However, it must continuously be updated and upgraded. Currently the team in charge of the development and maintenance of the DCSO has five main goals that are either being currently tackled, or are to be tackled in the near future. They are: (1) Reassess the need for the definition of individuals for the DCSX; (2) Further integration of terms from established ontologies into the DCSO (e.g., the Data Integration for Grant Ontology (DINGO)\textsuperscript{17}), thus achieving the goal of creating a serialisation of the DCS application profile, that has controlled vocabularies associated to all of its concepts; (3) The DCSO serialisation should be interchangeable with the current JSON serialisation of the DCS core set of terms. As such, an effort to establish the rules for the conversion between the DCSO and JSON is underway; (4) Performing semantic validation of DMP documents represented using the DCSO would be an ambitious but useful feature; and (5) As a result of the RDA Hackathon on Machine-Actionable Data Management Plans, there is an ongoing effort to create an extension to the DCS core set of terms. Any changes brought upon by this planned extension must be replicated by extending the DCSO as well.

Acknowledgements

The authors would like to thank the RDA for establishing the community that made this collaboration possible. This research was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UIDB/50021/2020, by project DRE, Imprensa Nacional - Casa da Moeda S.A. (INCM), and Portugal 2020. It was also carried out in the context of the Austrian COMET K1 program and publicly funded by the Austrian Research Promotion Agency (FFG) under grant 877389 (OBARIS) & 871267 (WellFort), as well as the Wirtschaftsagentur Wien (VasQua).

\textsuperscript{17} https://dcodings.github.io/DINGO
References