DDI – Cross Domain Integration: Introduction

Contents

I. Overview 1

II. Contents of the Specification 1

III. Purpose 2

IV. Key Features of the Specification 2

A. Domain Independence 2

B. Datum-Oriented Data Description 3

C. Provenance and Process Description 3

D. Foundational Metadata 3

E. Interoperability, Sustainability, and Alignment with Other Standards 4

V. DDI - CDI and the Suite of DDI Specifications 4

I. Overview

The DDI – Cross Domain Integration (DDI – CDI) specification provides a model for working with a wide variety of research data across many scientific and policy domains. It provides a level of detail which supports machine-actionable processing of data, both within and between systems, and is designed to be easily aligned with other standards.

It focuses on the key elements of the data management challenges facing research today: an exact understanding of data in a wide variety of formats, coming from many different sources. Two elements are critical for dealing with these challenges: a flexible means of describing data that can reveal the connections between the same data existing in different formats, and a means of describing the provenance of the data at a detailed (but comprehensible) level: the processes which produced it must be made transparent.

DDI - CDI covers these areas in a fashion intended to make it optimally useful to modern systems, which often employ a variety of models, and comply with a range of related specifications for both functions related to data description and process/provenance. The model is designed to be easy to fit into such systems, by aligning with relevant external standards, and to be alignable with them into the future.

II. Contents of the Specification

The specification is separated into several documents and files, appropriate to the material covered. These are described in the document “DDI-CDI Overview PR 1.pdf.”

III. Purpose

The DDI - CDI specification describes a model and supporting elements for implementing it in the areas of data description and process/provenance. It is not intended to supplant existing specifications for these purposes, but to fill in the information which such specifications often do not capture. For data, this is the description of a single data point – a datum – which can be used to play different roles in different data structures and formats. For provenance and process, this is the packaging of specific machine-level processes, which may be described in many different ways, into a structure which relates them to the business processes described at a level understandable to human users.

In order to serve this purpose, the DDI - CDI specification uses a Unified Modeling Language (UML) formalization so that it can be mapped against other models within systems more easily. Several different syntax expressions of the model are made available to support implementation.

Several important features of the specification can be highlighted, to show how it serves this purpose:

- Domain-independence
- Datum-Oriented Data Description
- Provenance and Process Description
- Foundational Metadata
- Interoperability, Sustainability, and Alignment with Other Standards

Each of these will be addressed in more detail, and an outline of the specification documents is presented.

IV. Key Features of the Specification

A. Domain Independence

DDI - CDI is designed to be used with research data from any domain. In order to do this, it is fundamentally based on the structure and other generic aspects of the things it describes. It does not attempt to be a domain model of semantics, nor a model specific to the life cycle of a particular domain of science or research. [Historically, DDI has focused on the Social, behavioral, and Economic (SBE) sciences some types of health research – to see how DDI - CDI relates to other DDI specifications, see the last section in this document.)

DDI - CDI is intended to be complimentary to (and used in combination with) other standards and models which focus more on domain-specific aspects (such as semantics and life-cycle models). Such generic elements such as classifications and variables are given a detailed formal treatment but are agnostic as to semantics and concepts. It is left to the user to employ whatever semantics and concepts are demanded by the data with which they are working.

This feature of the specification makes it well-suited to combining data coming from more than one domain or system, to allow a description of it that supports systems which perform data integration, harmonization, and similar functions. Cross-domain data sharing is becoming increasingly common, and DDI - CDI is intended to provide support for this type of application.

B. Datum-Oriented Data Description

DDI - CDI embraces a form of data description which is based on its atomic components: individual datums. Any given datum can play different roles in different formatting of the same data set, depending on how it is processed and transformed. In order to retain the continuity of a given datum across different formats and throughout a series of processes, DDI - CDI allows it to be described playing different roles in different structures.

DDI - CDI provides four basic types of structural description for data sets: wide data, long data, dimensional data, and key-value data. These four types (and their sub-types) provide coverage for many common data formats today. While not comprehensive, they cover the majority of cases that the developers of this specification have seen. These include many of the newer forms of data such as streaming data, "big" data, registers, and instrument data. The underlying approach is one which could – and may be – expanded in future. By assigning appropriate roles to datum in each of these different formats, however, it is possible to understand how data passes from one form to another.

C. Provenance and Process Description

If we are to fully understand data, we also need to know how it has been processed and transformed. Given our ability to describe how a different datum can be used in different data sets, it becomes desirable to understand also how those data sets relate to one another in terms of the processes which use them. This can be understood as an important aspect of data provenance.

There are many different ways of describing process and provenance. Popular models include the Business Process Modelling and Notation (BPMN) standard and the PROV Ontology (from W3C). There are a multitude of syntaxes for driving data transformation, cleaning, and analysis in packages such as R, SAS, Stata, MATLab, SPSS, Python, and so on. There are also some emerging standard models for specifically describing such specific processes (eg, SDTL, VTL).

DDI - CDI attempts to do something which compliments the use of such models, by connecting specific processes interpretable by machines at the lowest level (described in a package-specific syntax or language) with the higher-level flows which combine these into human-readable documentation of business processes. Both traditional linear processing and the newer declarative processing approaches are supported.

D. Foundational Metadata

In order to formally describe data at a detailed level, there are many component elements which themselves must be modelled. Statistical concepts and their various uses – including as categories and variables – are a core part of this, but the range is broad. These components are included in DDI - CDI as "foundational metadata."

Terminology for such constructs varies widely across domains. DDI - CDI has attempted to provide common terms for these components, and to adopt common models from other standards where it seemed useful.

One area which deserves particular attention is the "variable cascade" – a model for how the different types of variables relate to each other, and how they reflect the way data is described at different points in its creation, processing, and use. While many different models have a “variable” of some form, the one presented in DDI - CDI reflects the experience of working with this important construct in many of the specifications and standards which have preceded it. It is a nuanced view of how variables relate and are are understood across different systems, and – although not simple – it is a powerful model which helps solve some of the commonly encountered problems in data description and management.

E. Interoperability, Sustainability, and Alignment with Other Standards

DDI - CDI is fundamentally a model which is intended to be implemented across a wide variety of technology platforms, and in combination with many other standards. Models, and specifications. To support this use, it is formalized using a limited subset of the Unified Modelling Language (UML). The model is provided in the form of Canonical XMI – an interchange format for UML models supported by many different modelling and development tools. Further, a syntax representation is provided in XML, so that direct implementation of the model is possible if needed.

The platform-independence of the model makes it more easily applicable across a broad range of applications and helps ensure that it will be sustainable even as the technology landscape evolves.

DDI - CDI builds on many other standard models and is aligned with them where appropriate. This is shown in the model itself, where formalizations from other models and specifications are refined, extended, or directly used. The specification includes a description of what these other standards and models are, and how they are used in DDI - CDI.

V. DDI - CDI and the Suite of DDI Specifications

DDI - CDI is a different type of specification than its predecessors. It is not a continuation of or replacement for earlier DDI specifications such as DDI Codebook or DDI Lifecycle. It is intended to be complementary to these specifications for those applications – mostly in the SBE sciences – where DDI is used.

DDI – CDI builds on the work of many years in the DDI 4 project and brings some of the strengths of that effort to light. In this, it shares many features with later versions of DDI Lifecycle, which has also incorporated some of that work. Notably, the “variable cascade” comes from earlier DDI 4 models, as does the overall approach to describing non-rectangular data.

The DDI - CDI Model is the first specification produced by the DDI Alliance which uses a conceptual model expressed in UML as its basis. It is intended to describe many of the types of data which earlier DDI specifications describe. Due to the way in which data today is increasingly used across traditional domain boundaries, however, DDI - CDI is also (and of necessity) capable of describing data from many related domains.

The purpose of the specification differs somewhat from the earlier DDI Codebook and DDI Lifecycle specifications. Due to changes in the way in which information technology is applied to research and statistics, some new features are emphasized. Notably, the diversity of data types analyzed in a given project has increased, and the range of sources for that data has grown, with corresponding changes in the technology used to manage it.

The functional goal of the specification is also different: where DDI Codebook was an XML representation of a data dictionary, and DDI Lifecycle a more complex model designed to support metadata from data conception and capture through publication and reuse, DDI - CDI is an attempt to describe data and its provenance independent of these contexts.

Both DDI Codebook and DDI Lifecycle combine the description of structure (e.g. a table of records) and the description of meaning. In both, the primary structural form is a table or a cube. A variable and a column are basically synonymous. DDI - CDI disentangles structural description from description of meaning. This allows description of structural forms like tall tables or key-value stores.

The growing demand for data from different sources, and from external domains, requires that some different types of data be described. The provenance of this data – that is, the processes by which it has been assembled for use – are of increasing importance in understanding what it is and how it can be used. While traditional SBE data was often collected using questionnaires, alternate sources of data such as registers and sensors are becoming increasingly common and have in some cases always been typical. Completely new types of data from social media and other “mined” sources are also increasingly used.

The DDI - CDI model applies the important features of the pioneering (but unreleased) “DDI 4” work to these functions: describing various types of data in a way which makes them subject to integration and transformation into useable forms, and providing the information needed to understand their origins and provenance.

Because the way in which such a model can be implemented is more variable than it is for traditional SBE data management systems, the emphasis in DDI - CDI is on a model, formalized in UML, and made available using the Canonical XMI format which supports the exchange of UML models between various tools, including both modelling and development environments. While XML is still supported, it is no longer the canonical format for the specification.

DDI - CDI is aligned with earlier DDI specifications, most notably DDI Lifecycle, as it is anticipated that it might be used as an integration model for systems based on these earlier specifications. The intention is that DDI - CDI be a tool which can supplement systems using earlier versions of DDI, enabling them to better handle new types of data.