Common Data Architecture for Life Sciences

Version 0.1 - May 1, 2024

Owner: Matt Robinson

Overview

The Common Data Architecture for Life Sciences (CDA.LS formally, CDA for short) is an industry standard for operational data in life science companies. It is a simple set of data structures that are small, easy to understand, and easy to implement.

The goal of CDA is to standardize operational data in the life science industry, helping software applications, data products, and people to talk to each other with greater consistency and accuracy. By establishing standard industry names, data types, and definitions, CDA creates a common understanding within and between organizations that increases speed, efficiency, and quality.

Kernels

CDA is composed of kernels. Each kernel defines a data structure for one or more related entities used in the life sciences industry, such as Healthcare Professional (HCP), Healthcare Organization (HCO), Study, Study Site, Product, Disease, etc. Kernels encapsulate the essence of entities in a compact form, encompassing only the bare essential definitions and attributes necessary for fundamental understanding and industry-wide interoperability. Some attributes include a list of industry standard picklist items.

Bare Essentials

It is crucial that each kernel remains small. If a kernel becomes too large, quality will suffer, complexity will increase, and adoption will be hampered.

The current approved set of CDA kernels is:

Kernel	Entities
CDA.LS.HCP	Healthcare Professional (HCP), HCP_Segment, and Address

Components

The definition of entities, their attributes, and their lists of picklist items are referred to as the components of a kernel. Each component has a name, English label and English description. Attributes also have a data type.

Entities

Each entity within a CDA kernel represents a type of person, place, or thing in the life sciences industry. Kernels can contain one or more related entity definitions. For example, the CDA.LS.Clinical kernel may contain entity definitions for Study, Study Site, and Participant.

Attributes

Attributes are the properties or characteristics of an entity that define and describe it. They represent the data that is collected and stored about an entity. Each attribute is designed to hold a piece of information (i.e. a value) that is significant for the entity it describes.

Each attribute consists of the following properties:

- **Entity**. The Name of the entity the attribute belongs to.
- **Name**. Alphanumeric, up to 50 characters, no spaces. An immutable and unique attribute identifier used for system interoperability in databases, APIs, etc.
- **Label**. Up to 100 characters. A user-facing display name for the attribute used in user interfaces like forms, reports, etc.
- **Data Type**. One of `Text`, `Number`, `Boolean`, `Date`, `DateTime`, `Picklist`, `Multivalue Picklist`, `Entity`. Determines the format of this attribute's value (see *Data Values* below).
- **Description**. Up to 1,000 characters. A description of what the attribute represents.
- **Picklist Items**. A list of items for attributes with the `Picklist` data type. Each item consists of three data elements.
    - **Name**. Up to 10 characters, no spaces. An immutable and unique attribute identifier used for system interoperability in databases, APIs, etc.
    - **Label**. Up to 50 characters. A user-facing display name for the item used in user interfaces like forms, reports, etc.
    - **Description**. (Optional) Up to 1,000 characters. A description of what the item represents.

Data Values

The value for each attribute in an instance of data should conform to a format based on the attribute's data type as follows. All data is assumed to be UTF-8.

- **Text.** Can range from 1 to 1,000 characters.
- **Number.** Up to 100 integral digits and 10 decimals with no thousands separators.
- **Date.** An ISO 8601 date value in the `YYYYMMDD` format or the `YYYY-MM-DD` extended format.
- **DateTime.** An ISO 8601 date or date and time value. Date values are expected to be in the `YYYYMMDD` format or the `YYYY-MM-DD` extended format. DateTime values are expected to be in the `YYYYMMDDTHHMMSS` format or the `YYYY-MM-DDTHH:MM:SS`

extended format. A time zone value can be added using the offset from UTC in the `+/-HHMM` or `+/-HH:MM` format, such as `-0800` or `-08:00` for Pacific Standard Time (PST, UTS-8). For example, `2024-06-15T09:30:00-08:00`. When no time zone designation is used, or a `Z`, `UTC+00:00` is assumed. For example, `2024-06-15T09:30:00Z`.

- **Boolean.** A `true` or `false` value.
- **Picklist.** A single option name such as `OPA`.
- **Multivalue Picklist.** A comma-separated list of option names such as `OPA,OPB,OPC`.
- **Entity**. Represents a relationship with another entity. Note that the CDA does not define how relationship keys should be structured. See *Entity Relationships* below.

Entity IDs

Instances of kernel entities often have unique identifiers in practice, such as a primary key. While CDA documents may make reference to such unique identifiers, they are not always standardized in the kernels. Some entities that represent data provided by Veeva have a unique VeevaID attribute. With the exception of VeevaID, it is up to application developers and data providers to choose how to handle support for unique identifiers.

Entity Relationships

Instances of kernel entities are often related to one another in practice. These relationships can be between entities within or across kernels. For example, an HCP is often related to one or more Addresses, Segments, and HCOs. Similarly, each Study, Study Site, and Participant usually has a relationship with one or more instances of these entities.

CDA documents do make reference to entity relationships and their cardinalities, and the kernels do specify attributes for these relationships. However, it is up to application developers and data providers to define how relationship keys are structured.

Common Attributes

Some attributes may be identical across kernels. This will happen infrequently. In these cases, duplication may occur and we strive for consistency of attribute names when this happens.

Internationalization

All kernel component labels and descriptions are defined in English. For each kernel, translated labels are provided for the following languages in a separate translation file:

1. German
2. French (France)
3. Spanish (Spain)
4. Italian (Italy)

5. Chinese (Simplified)
6. Japanese
7. Korean
8. Portuguese (Brazil)

It is up to application developers and data providers to choose how to handle support for different languages and locales.

Change Management

CDA kernels are treated as a Data API, with a specific support and deprecation policy similar to developer APIs:

- The names of components will not change after they are published.
- Component labels and descriptions may change to provide clarity or correct errors.
- New components may be added in any new version.
- Announcement of component depreciation can be made at the start of a calendar year, initiating a 3-year notice period. After this period, deprecated components will be removed.

Implementing CDA-Compliant Solutions

While Veeva does not prescribe how CDA should be implemented in software and data solutions, we do recommend some best practices to maximize interoperability.

Best practices:

1. **Name tables using entity names.** Where possible give your tables names that match or closely align with CDA entity names.
2. **Name fields using attribute names.** We recommend adding "_cda" to the end of field names to clearly distinguish CDA fields.
3. **Name picklist items using picklist item names.** Where possible picklist item names in the CDA are based on broadly accepted standards such as those from ISO. Therefore, we recommend using these as is.

Change History

As new versions of CDA are released, a summary of changes will be included here.

Veeva Public CDA License

Veeva provides the common data architecture ("CDA") and associated documentation at www.veeva.com/cda for no charge.

Right to CDA documentation. Veeva grants to you a perpetual right to copy, print, and distribute CDA documentation so long as you retain all copyright notices and include a copy of this Public CDA License Agreement with such documentation. In addition, you may create derivative works of CDA documentation as necessary to implement, or facilitate the implementation, of CDA.

Implementing CDA. Veeva grants to you a perpetual right and license under its intellectual property rights to implement CDA in your software or database, ("Your Implementation") provided that you:

1. Include the following attribution in Your Implementation in the same manner as you attribute other copyright or trademark notices: "Built using the Common Data Architecture (CDA) by Veeva available at www.veeva.com/cda".
2. Do not use CDA or associated documentation to create an alternative standard that is intended to compete with or otherwise create confusion with CDA.
3. Do not file or participate in any intellectual property lawsuit against another based solely on an implementation of CDA.
4. Do not place any new or different limitations or restrictions on use of CDA on downstream users of Your Implementation.

Changes to CDA. Veeva retains control over the content of CDA and makes it available to the public at no charge. Accordingly, if you suggest any changes or modifications to CDA or its associated documentation, then you agree to waive all rights in such suggestions and agree that Veeva may use or incorporate such suggestions (in whole or in part) into CDA or associated documentation without any further obligation to you.

No other rights. No other rights except those expressly stated in this promise shall be deemed granted, waived or received by implication, exhaustion, estoppel, or otherwise.