## Scope

This document defines how the content of the City Data Model and Reference Architecture (CDMRA) is developed and managed.

### Scope of data considered

Smart City environments are divided into multiple subdomains, such as transportation, health, governance, power, etc. The CDMRA is envisioned to document data that is shared across domains and within individual domain areas. The management structure of the CDMRA is designed to accommodate the development of data definitions by those experts with a specific interest in the data without imposing extensive bureaucratic overhead from other domains. As a result, data that is produced and shared within a single domain area of Smart Cities can be produced by that domain; however, data produced and shared across domain boundaries require review from a broader range of stakeholders. The scope of the CDMRA includes both domain-specific and cross-domain data, with a particular interest in cross-domain data.

### Technical scope

The purpose of the CDMRA is to define the City Data Model (CDM) to provide unambiguous definition of data elements that can be shared across different domains of Smart City environments. While the CDM provides the formal data definitions, the encompassing reference architecture provides context to this data by identifying known examples of how the data is used to satisfy user needs through the Use Case View.

It should be recognized that the CDMRA does not claim to provide an exhaustive list of all possible uses of the data. The intent of the Use Case View of the CDMRA is to provide an initial justification for the data as well as examples of how the data is used within Smart Cities such that other potential users of the data have a fuller understanding of the context that the data was intended to exist within. The CDMRA formally defines the CDM; the other components of the CDMRA are informational and only developed to support the needs of the CDM.

### Protocol mapping

By adopting the definitions contained in the CDMRA, the component systems of a smart city will be able to more easily share information with one another. The CDMRA does not attempt to define how to share this data (i.e., the syntax); it merely assists in integrating data from disparate protocols by defining common data semantics to which all protocol data can be mapped.

It is envisioned that data transformations will be defined to translate data from specific protocols into their corresponding data elements contain in the CDM. Such transformation rules will facilitate the unambiguous translation of data among different protocols and avoid mistakes in translations. The CDMRA collaboration environment makes provisions to store this type of information but this information is not currently the focus of the CDMRA development.

## Understanding the Model

While the primary goal of the CDMRA is to formally defined the CDM, learning the contents of a data model by itself is a bit like trying to learn a language by reading a dictionary. In practice, data models are most useful when presented in small chunks that can be directly related to a practical use of the data.

To promote comprehension, the CDMRA adopts a use-case approach to defining data. The use cases are defined within the Use Case View of the CDMRA using a standard template. Each use case identifies the data required by the use case and typically includes one or more class diagrams that depict the relationships among this data.

The Use Case View also includes hyperlinks into the Information View, which provides additional details about the data, including:

1. Detailed definitions of each data concept required by the use case (the City Data Model)
2. Diagrams depicting how this data relates to other data (e.g., data that might not be used by the specific use case)
3. Links to each use case that uses each data element
4. Identification of specifications within the Specification View that define how the data is implemented in real-world interfaces
5. Links to outside resources that provide additional context for the data (e.g., external reference architectures)

The various parts of the CDMRA are summarized in Figure 1.



Figure : CDMRA Overview

## Contributing to the Model

The CDMRA is an open-source, collaborative product developed by and for the technical community that designs, develops, and maintains Smart City systems, including systems for any of the Smart City sub-areas, called domains. These domains include, but are not limited to:[[1]](#footnote-1)

* smart government
* smart transportation
* smart education
* smart health care
* smart home
* smart campus

### Approval Process

Submittals to the CDMRA can be made by any registered user of the cooperative environment, which is located at <https://www.smartdatamodel.org>. To register contact Mark Fox at the University of Toronto. Once registered, a user is able to select his or her domains of interest within the profile page of the website.

Registered users may submit:

* proposals for any ***collaborative architecture element***: these elements are fully documented and copyrighted as an integral part of the CDMRA;
* contributions of any ***catalogued architecture element***: these elements remain copyrighted by their original source and the CDMRA will often only include a reference to where the full definition can be found in the original source.

Catalogued architecture elements include external specifications and external resources. It is envisioned that catalogued elements from multiple sources will be submitted. Each catalogued element is subject to its own copyright statement. If the source allows for reproduction of content, the submitter may provide this content along with the copyright statement that applies to the content. Sources that do not allow for reproduction shall only be referenced using fair use of its content. As the content of these external elements are defined by external organizations, no CDMRA approval process is defined other than efforts to ensure that copyrights are respected.

Collaborative architecture elements include all elements of the CDMRA, except for the external specifications and external resources. Submittals of collaborative elements follow the approval process depicted in Figure 2 and described below.



Figure 2: Maturity Levels of Model Elements

An architectural element is initiated in the “formulation” state. Can the entry be viewed by others? Can others revise the content when in this state? Once the registered user has provided syntactically valid content for all mandatory fields for the element, the element will transition to the “tentative” state. Can the entry be viewed by others? Can others revise the content when in this state (or perhaps the originator can transfer ownership)? If others can see/edit, can they submit for review? Once the registered user is satisfied with the metadata entered for the architectural element in the “tentative” state, it may be submitted for review, whereupon it enters the “proposed” state. What happens if the user attempts to delete a mandatory item? Does the state revert to “formulation” or does the deletion fail (i.e., you allow change but not deletion)?

Upon each entry/re-entry into the “proposed” state, the revision number is incremented, and control of the element is passed to the appropriate domain data steward as defined by the domain and subdomain metadata for the element.

Version numbers shall be in the form of “<major version>.<minor version>.<revision>-<state code>”. Elements in the “tentative” state shall always be identified as version “1.0.0-T”; upon first entry into the “proposed” state, the element will be identified as version “1.0.1-P”.

The domain data steward will review the element to determine if the element is assigned to the proper domain and subdomain. At least once a month, each domain data steward shall provide a report to all other domain data stewards identifying the data concepts proposed within the domain and any recommendations to change these assigned domains. These reports shall be reviewed monthly by the domain data stewards. The domain data steward shall implement any change to the domain and subdomain as agreed to by a simple majority of the domain data stewards present.

Any registered users can review and submit comments on any collaborative architectural element within the CDMRA with a state of “proposed” or above. Any registered user can also respond to comments submitted by others. The preferred mechanism for submitting comments is to use the discussion forum on the page of the collaborative architectural element for which the comment applies, but comments can also be verbally submitted during domain working group meetings. The domain data steward will guide these discussions towards resolution and make the changes that the working group agrees to. Each revision (or group of revisions performed in one action) to an architectural element will cause the element to be “revised” and re-enter the “proposed” state, whereupon the revision number is incremented.

Once the domain working group has completed its review of the element, it can either decide to reject the proposed element, whereupon it transitions to the “closed” state and is archived with the state code of “C”, or can accept the element, whereupon it transitions to the “draft” state. The state code for the “draft” state is “D” (e.g., a draft element might have a version number of “1.0.13-D”).

Comments received on elements in the “draft” state are handled in the same manner as with elements in the “proposed” state. If revisions are made, the element returns to the “proposed” state (whereupon the revision number is incremented) and may either remain there or may be elevated again by a new decision by the domain working group.

If an element in the “draft” state includes a reference to at least one specification model, it automatically transitions to the “pendingApproval” state, with the state code of “pA” (e.g., version number “1.0.13-pA”). Comments received on elements in the “pendingApproval” state are handled in the same manner as with elements in the “proposed” and “draft” states.

Once a quarter, the domain data steward shall provide a report to the domain working group that identifies elements that are in the “pendingApproval” state for the entire Smart City Architecture. Once a specific revision of an element has been in the pendingApproval state for at least 180 days without any revisions, the working group may (but is not required to) promote the element to the ‘approved’ state, with the state code of “A” (e.g., version number “1.0.13-A”). Comments received on elements in the “approved” state are handled in the same manner as with elements in the other states; however, upon exiting the “approved” state, the minor version number is incremented and the revision number is reset to zero. NOTE: The revision number is then incremented upon entering the “proposed” state, so the version number after “1.0.13-A” would be “1.1.1-P”.

The domain data steward will assess whether revised architectural elements are backward compatible with their previously approved versions. For elements that are not backward compatible, the domain data steward will set a flag that indicates that the major version number needs set to the lowest previously unused major version positive-integer for the element; this act will automatically reset the minor revision number (e.g., “1.1.1-P” would likely become “2.0.1-P” unless a previous attempt to revise version “1.0.13-A” was unsuccessful – e.g., and resulted in “2.0.10-C” as an example – in which case, the version number would increment to the next unused major version – e.g., “3.0.1-P”).

The above should work pretty well, but we will have to consider how each correspondence rule might affect the revision process. For example, if I add a property to a class, does that count as a revision of the class? If I update a diagram to show the new element, does that affect every element that includes that diagram (or is the diagram its own separate element)? What happens if the data steward later removes the flag due to the decisions of the WG?

### Preferred Sequence of Content

While contributions may be made in any order, the mandatory fields of elements are based on the general flow described in Figure 3.



Figure 3: Process for Adding Content

The content of the CDMRAis intended to be based on use cases so that users of the architecture can understand the context in which data might be used by end systems. The use case is formally specified using the use case template as defined by the Use Case Specification Model Kind, which may include a use case diagram. The use case specification will eventually be supplemented with one or more Information Flow Class Diagrams but these might not be present upon initial submittal.

The use case specification includes the ability to link to external references to define the reference deployment architecture, which identifies the physical system components that exchange information to realize the use case and the descriptions of the flows between the components. This information can provide valuable context for how the use case is envisioned to be implemented, but it outside the scope of the CDMRAand the conventions used to document this information might vary by domain.

The next step in the process is to identify any existing specifications that have already defined data for this type of data or information flow. The goal of the CDMRAis to leverage existing efforts as much as possible, but there is recognition that in many cases, there are several existing standards that define the same core data. By identifying these various sources up front, we can gain a better understanding of where the entire industry is rather than basing our design off of a single effort.

Once we have identified various source specifications, we can begin the work of creating our City Data Model content. We start with defining the key terms (i.e., the classes within our model) and then continue to define the data elements (i.e., the properties of each class).

## Roadmap

The CDMRAis intended to specify the semantics of all data concepts that might be used within a Smart City environment, including data used within a single domain. This is a massive undertaking that will likely take decades to mature and will never be “done”. Managing such a large effort requires a management roadmap that identifies the major topics of current development and the major milestones along the way. This roadmap will be updated as the CDMRAmatures and is highly dependent on the volunteer efforts from each domain.

The current roadmap focuses on activities to achieve the following major milestones in order or priority:

1. Defining the architecture framework for the City Data Architecture. This document will define the viewpoints, model kinds, and correspondence rules that will be used to document the CDMRAper ISO 42010. It will also clearly identify the fields that are deemed to be mandatory prior to elevating an element to the “tentative” state.
2. Defining the governance rules for the CDMRAas proposed by this document, which how content is contributed and approved for the architecture.
3. Developing prototype content to demonstrate how architectural elements should be captured within the City Data Architecture. The current intent is that this will focus on use cases related to routing (e.g., real-time navigational routing, pre-trip navigational routing, routing of public transport lines, etc.).
4. Developing specific domains based on volunteer activity. Interest has already been expressed for the following topics:
	1. Transportation planning
	2. Kerbside management
	3. Parking management
	4. Work zone management

## Support

How does one get support in contributing to the architecture?

## Credits

The City Data Model is managed and hosted by the World Wide Web Consortium (W3C), but it represents a collaborative effort of many partners, which currently includes:

* ISO/IEC JTC1 WG 11
* ISO TC 204
* ISO TC 211
* OGC
* W3C

## Copyright

The content of the CDMRAshall by copyrighted under the ??? License.

1. The listed domains equate to the Smart Application Layer entities defined in ISO 30145-2. [↑](#footnote-ref-1)