

The Canadian Heritage Information Network's Target Model and Semantic Paths Specification: Facilitating the Understanding of CIDOC CRM for Canadian Museum Institutions

Canadian Heritage Information Network (CHIN)
Philippe Michon
Semantic Web Analyst

49th joint meeting of the CIDOC CRM SIG and
ISO/TC46/SC4/WG9
March 11th 2021



Links shared during the meeting

- [Collections Model Website:](https://chin-rcip.github.io/collections-model/)
<https://chin-rcip.github.io/collections-model/>
- [Collections Model Github Repository:](https://github.com/chin-rcip/collections-model)
<https://github.com/chin-rcip/collections-model>
- [CRITERIA Live Demonstrator:](http://chinrcip.pythonanywhere.com/)
<http://chinrcip.pythonanywhere.com/>
- [CRITERIA Github Repository:](https://github.com/chin-rcip/CRITERIA)
<https://github.com/chin-rcip/CRITERIA>
- [diagrams.net Libraries Repository:](https://github.com/chin-rcip/diagrams.net_libraries)
https://github.com/chin-rcip/diagrams.net_libraries

CHIN's mandate

Special Operating Agency within the Department of Canadian Heritage

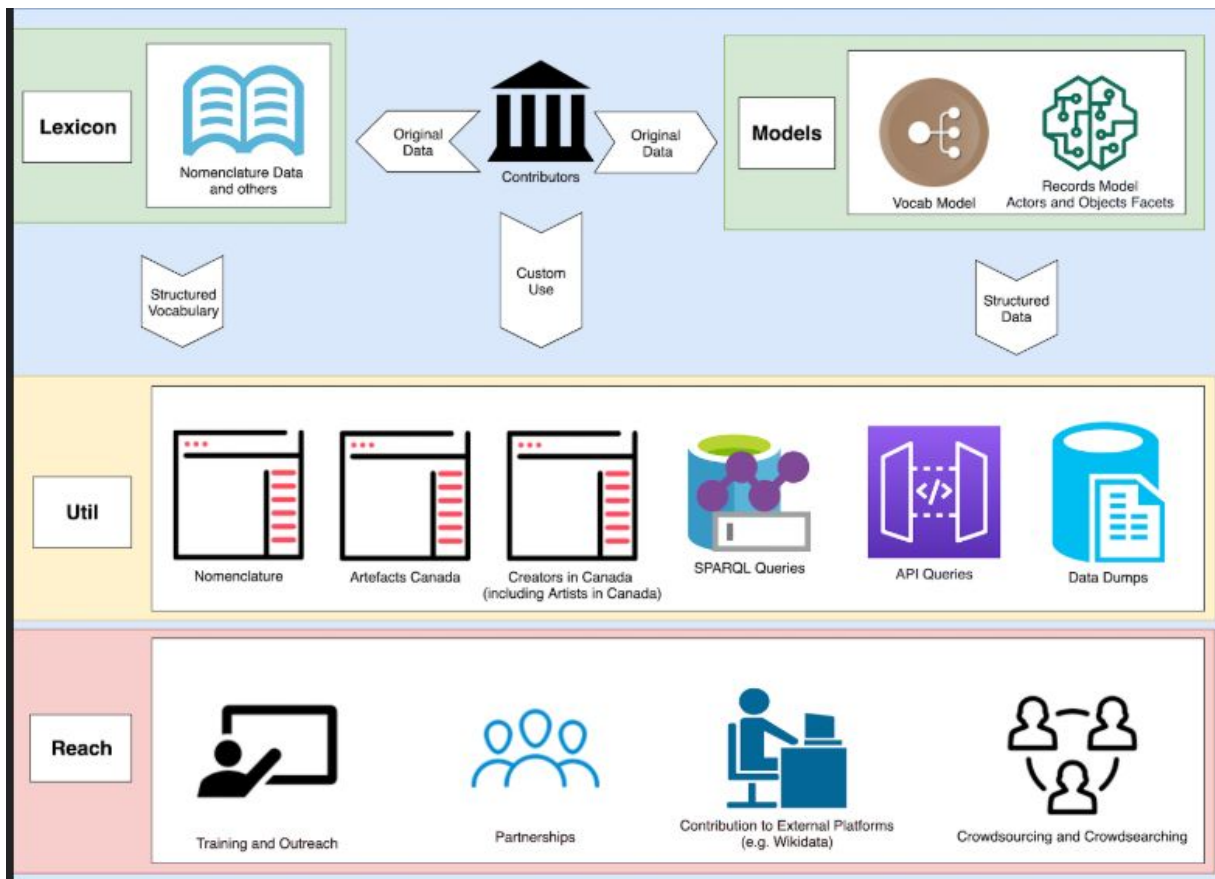
Assists Canadian Museum in documenting, managing, and sharing information about their collections

Areas of activity:

- Maintain an online point of entry to Canadian collections (Artefacts Canada)
- Carry out research and development on collections documentation tools and standards
- Provide guidance and training to cultural institutions on managing collections information



Operational Activities



Target Model

Aims:

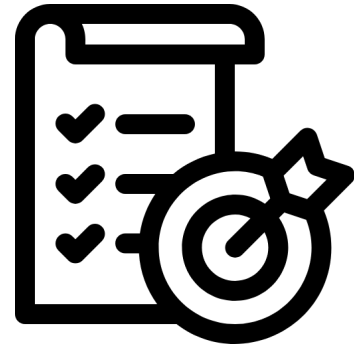
- Define semantic patterns based on CIDOC CRM
- Identify entry points for museums
- Promote the cooperative development of the model
- Present the current state of the research

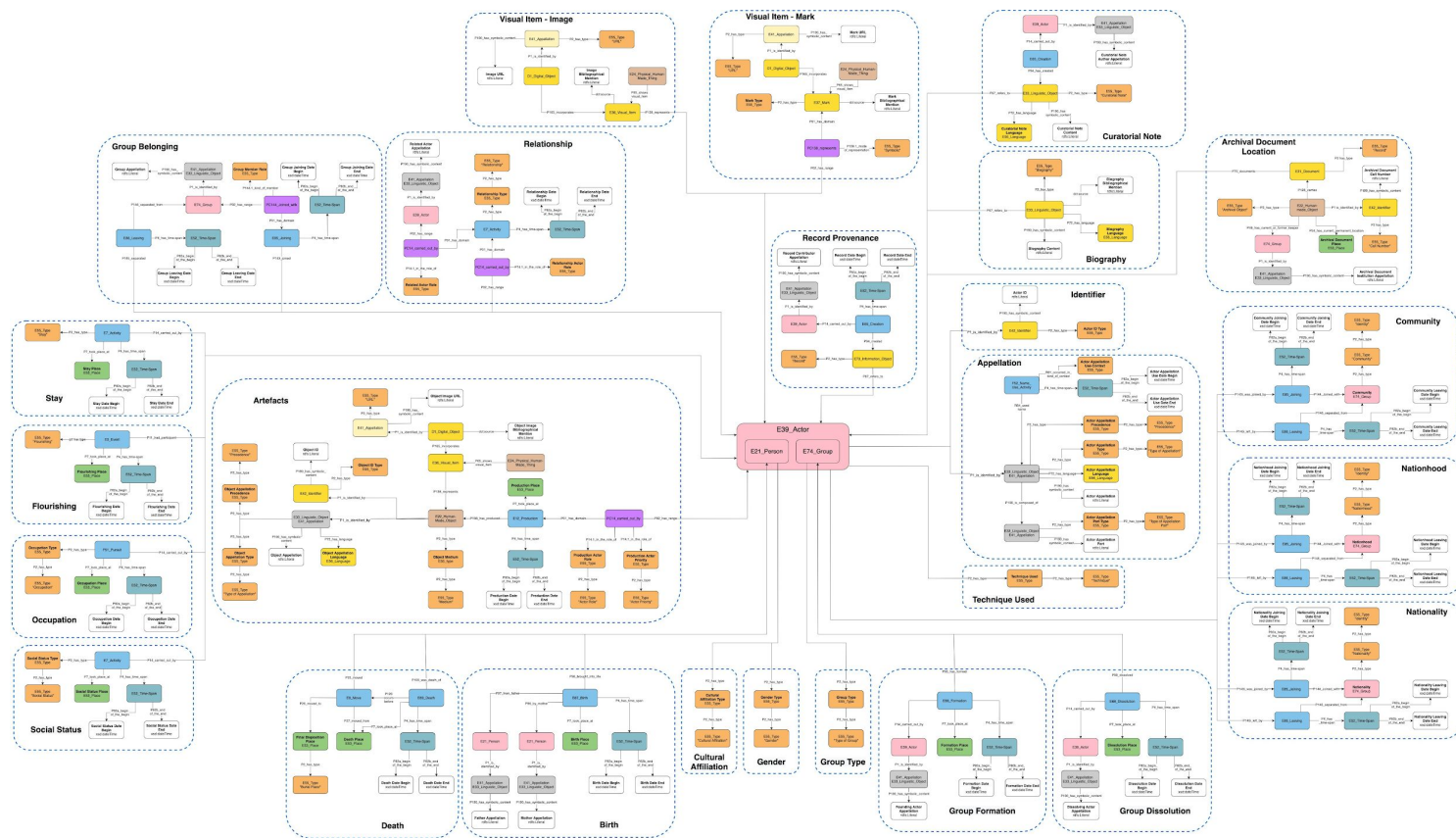
Approaches:

- Limit the creation of new classes and properties
- Provide space for less structured data

Intended Audiences:

- Ontologists
- Researchers





Target Model



E2_Temporal_Entity

E18_Physical_Thing

E28_Conceptual_Object

E53_Place

E39_Actor

E52_Time-Span

E54_Dimension

E55_Type

PC classes

Literal

Identifiers and Appellations (Example and Issue)

💡 Example 1:

Jean Paul Riopelle's IDs could be:

Different Identifiers

URI: mic.ca/uri/actor/1234

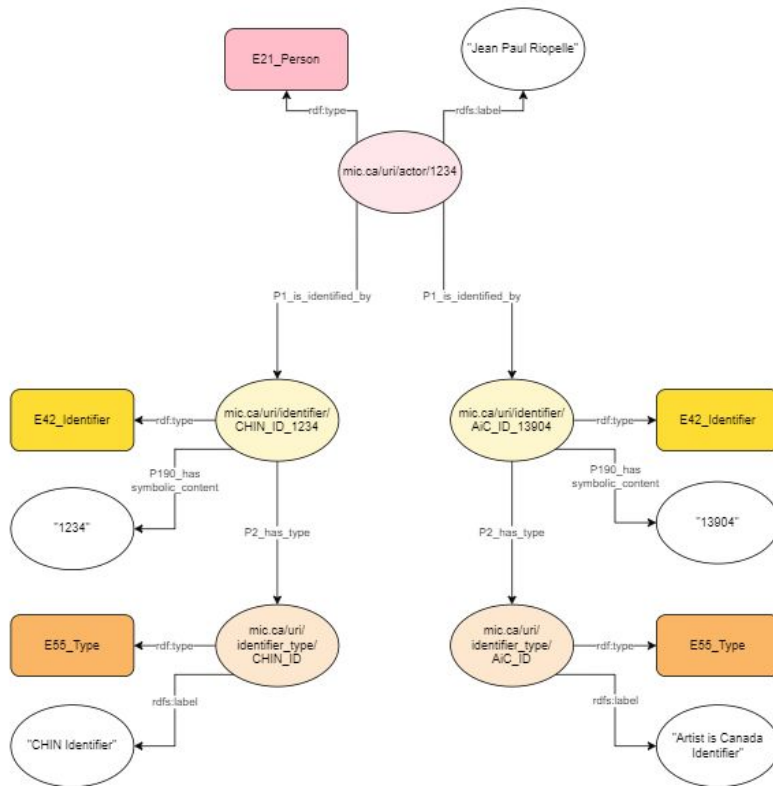
CHIN ID: 1234

Artist in Canada ID: 13904



Related Github Issue

This topic is discussed in [Issue #25](#)



Semantic Paths Specification

Aims:

- Define all our Entry Nodes
- Describe the semantic paths that will be generated from the input value
- Propose quality thresholds
- Present typical and edge cases
- List the relevant vocabularies

Approaches:

- Be precise in the description
- Emphasize the limitation of the model

Intended Audience:

- IT professionals



Actor Appellation Part Type

| | |
|--------------------------|---|
| Scope | This field qualifies the Part of the Actor Appellation and conceptually characterizes Parts of the Actor Appellation in order to facilitate identification, belonging to larger entities (such as families or brands) and placement in the Appellation. |
| Generated Bond(s) | E55_Type |
| Dependency(ies) | Actor Appellation Part |
| Related SQN(s) | Type of Appellation Part: Definition |
| Full Path | crm:E39_Actor -> crm:P1_is_identified_by -> crm:E41_Appellation + crm:E33_Linguistic_Object -> crm:P106_is_composed_of -> crm:E41_Appellation + crm:E33_Linguistic_Object -> crm:P2_has_type -> crm:E55_Type (-> <i>crm:P2_has_type -> crm:E55_Type["Type of Appellation Part"]</i>) |

Semantic Paths Specification

| | |
|-----------------------------|--|
| Target Model View(s) | Identifiers and Appellations |
| Semantic Valuation | <p><i>Low:</i> The value provided is a string that cannot be parsed automatically by reconciliation tools but still contains the required information.</p> <p><i>Medium:</i> The value provided is a string that can be automatically cleaned and reconciled using controlled vocabularies specified by CHIN. In the event that the value is adequately structured but does not reconcile with any terms from the controlled vocabulary, and provided that it is deemed relevant, CHIN will add the term to the vocabulary. Accepted Value Type(s): String</p> <p><i>High:</i> The provided value is a string that can be automatically reconciled with the controlled vocabularies specified by CHIN. Accepted Value Type(s): String, URI</p> |

| | |
|------------------------|--|
| Typical Case(s) | The record of a painting in a collection by the painter Emily Carr entitled <i>Kitwancool</i> made in 1928. “Emily” is the first name actor appellation part type and “Carr” is the last name actor appellation part type. |
| Edge Case(s) | <p>Middle names in general are hard to situate within an Actor Appellation. For example, Jean Paul Riopelle could be considered to be “Jean” (Actor Appellation Part Type “First Name”), “Paul” (Actor Appellation Part Type “Middle Name”) and “Riopelle” (Actor Appellation Part Type “Last Name”) just as it could be “Jean Paul” (Actor Appellation Part Type “First Name”) “Riopelle” (Actor Appellation Part Type “Last Name”). How to determine the order and display of parts will thus be a challenge.</p> <p>The same applies to organisations’ appellations that might have multiple parts such as “Randy & Berenicci”, which would be composed of three Actor Appellation Parts (“Randy”, Actor Appellation Part Type “Root”; “&”, Actor Appellation Part Type “Affix”; “Berenicci”, Actor Appellation Part Type “Root”). In this case how to determine the order of the Actor Appellation Parts could not be automatised and would likely be problematic. In addition, whether to classify an artistic duo (or trio etc.) as an organisation (using an Actor Appellation) or a group (using a Group Appellation) has not yet been determined. Provided the duo considers itself to be an autonomous artistic entity, it would not suit a Group Appellation considering Group belonging revolves around the multiplicity of entities.</p> <p>How to display and order affixes in general will be a challenge as well as some are typically displayed prior to the name (e.g. Sir) whilst others (e.g. Ph.D.) typically come after.</p> |

| | |
|-----------------------------|--|
| Value Origin(s) | Actors Checklist: Actor Appellation Part Type: Provider Data |
| Controlled List/Term | TBD |
| Potential Error(s) | TBD |
| Comment(s) | Any part of the Actor Appellation can be typified here, including elements that are not typically thought of as names such as manners of address or legal affixes. |
| Reference(s) | (Le Boeuf et al. 2015, sec. E55 Type) , (Wikipedia 2019h) , (Wikipedia 2019af) , (Wikipedia 2019i) |

Checklist

Aims:

- Facilitate the fields/models alignment
- Share simpler definitions
- Analyze fields' content

Approaches:

- Hide CIDOC CRM
- Offer a known format

Intended Audience:

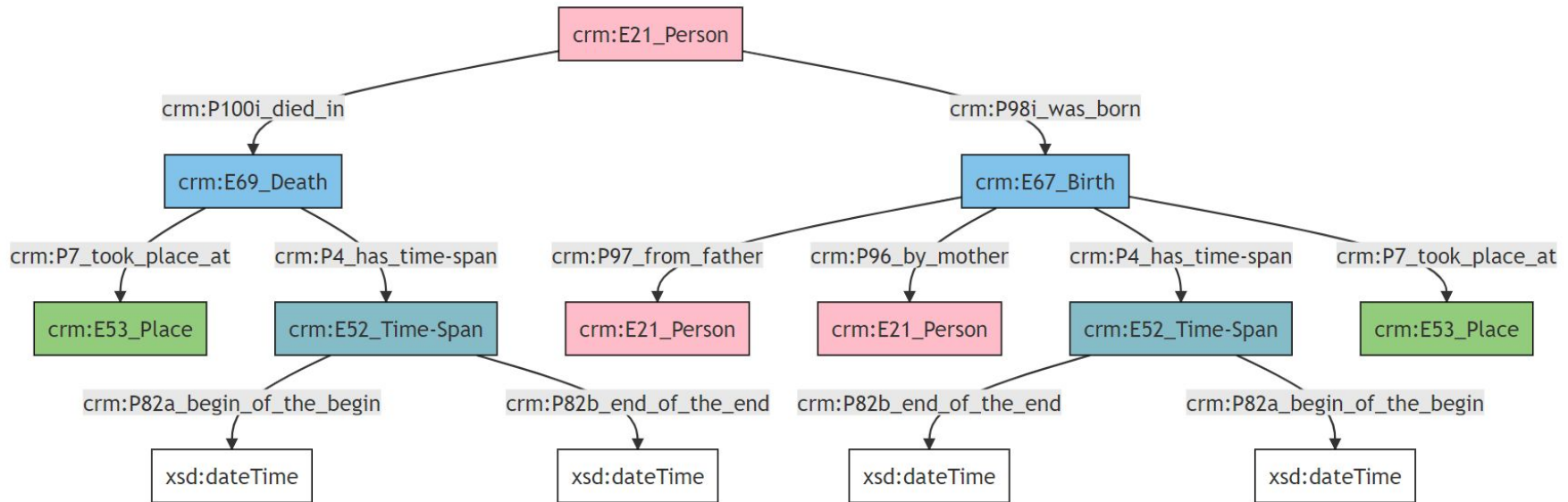
- Cataloguers



| CHIN's Elements | | | | | | |
|-------------------|---|---------------|--------------------|----------------------|--|---------------------|
| Field Name | Field Description | Type of Value | Field Requirements | Example Data | Comments | Provider Field Name |
| Actor Appellation | Word(s), number(s), symbol(s) or code(s) by which an entity (personal or corporate) is identified and referred to. This will be the information in 'Artist/Maker' or 'Manufacturer' in AC DD. | String | UTF-8 | "Jean Paul Riopelle" | This information will normally be found in a CMS 'creator' field. In AC DD, this is Artist/Maker or Manufacturer. **we need to resolve how we deal with unknown artists/culture. CCO recommends putting the culture name in the artist/maker field. Eg. Creator display: Mandinka People (West Africa, 19th century) This is how we deal with western artists (eg. unknown Italian). AC DD also has Artist Other Names for non preferred names the artist is known by. Is it not preferred to have controlled vocabulary in here where possible? For example if they are listed in ULAN? | |

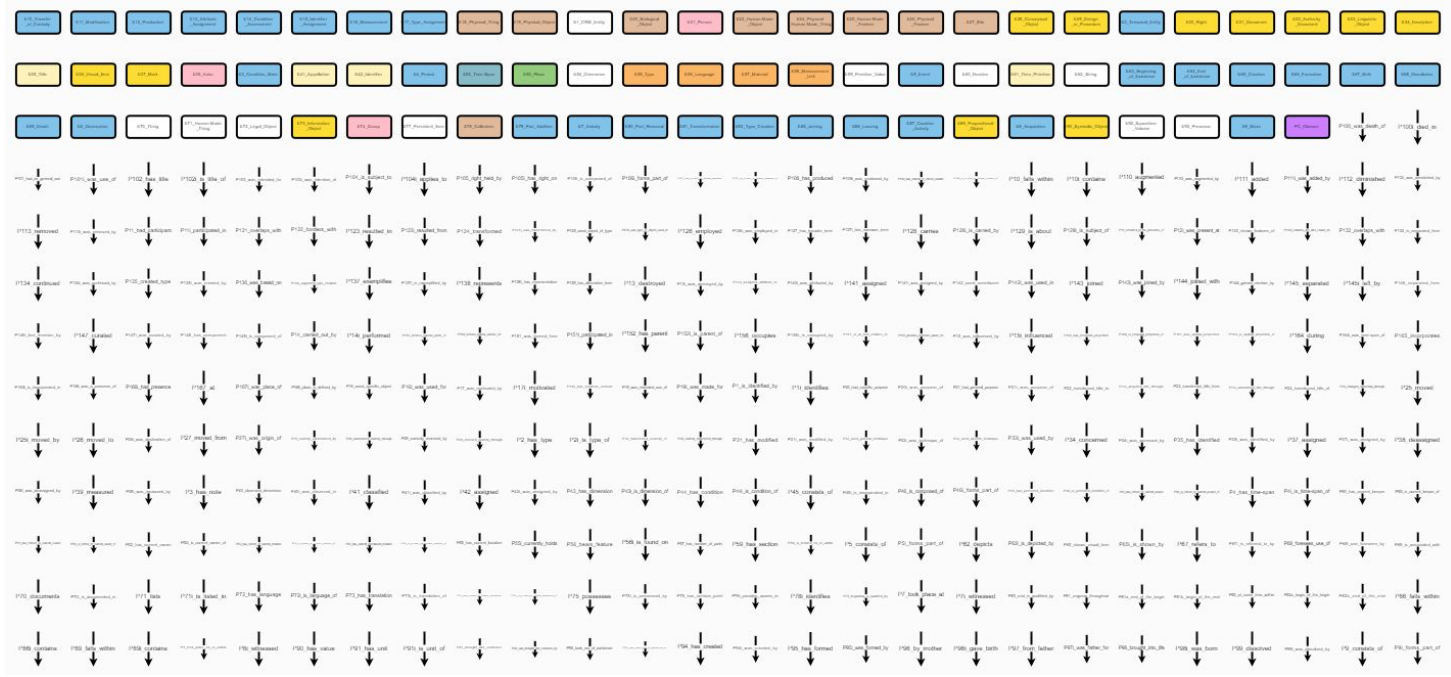
CRITERIA

- Python script to create Mermaid Diagrams from RDF files
- Visualization of the instances (examples) or the ontology
- Available script and Live Demonstrator



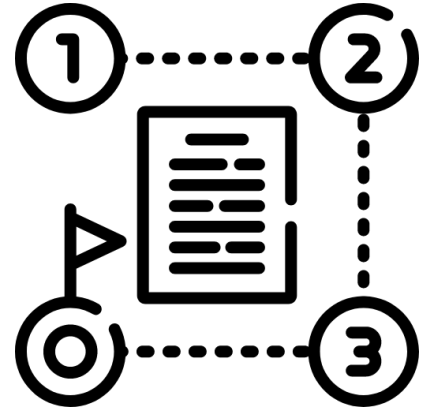
diagrams.net Libraries

- diagrams.net Libraries for CIDOC CRM and a few extensions
- All the classes and properties



Next steps

- Generic patterns
- Pattern approval stamps
- Perfect dataset
- RDF serialization
- Semantic Valuation Enhancement
- Stakeholders' datasets testing
- Object Facet Documentation
- Automated processes
- How-tos
- Even more collaboration!



Questions?

philippe.michon@canada.ca