CIHI

Pan-Canadian Health Data Content Framework

Semantic Common Data Model Implementation Guide

Version 3, September 2026

Production of this document is made possible by financial contributions from Health Canada and provincial and territorial governments. The views expressed herein do not necessarily represent the views of Health Canada or any provincial or territorial government.

Unless otherwise indicated, this product uses data provided by provinces and territories in Canada.

All rights reserved.

The contents of this publication may be reproduced unaltered, in whole or in part and by any means, solely for non-commercial purposes, provided that the Canadian Institute for Health Information is properly and fully acknowledged as the copyright owner. Any reproduction or use of this publication or its contents for any commercial purpose requires the prior written authorization of the Canadian Institute for Health Information. Reproduction or use that suggests endorsement by, or affiliation with, the Canadian Institute for Health Information is prohibited.

For permission or information, please contact CIHI:

Canadian Institute for Health Information
350 Albert Street, Suite 1400
Ottawa, Ontario K1R 1A4
Phone: 613-241-7860
Fax: 613-241-8120
cihi.ca
copyright@cihi.ca

© 2026 Canadian Institute for Health Information

How to cite this document:
Canadian Institute for Health Information. *Pan-Canadian Health Data Content Framework — Semantic Common Data Model Implementation Guide, Version 3, September 2026*. Ottawa, ON: CIHI; 2026.

Cette publication est aussi disponible en français sous le titre *Guide de mise en œuvre du modèle de données sémantique commun du Cadre pancanadien de contenu des données sur la santé, Version 3, septembre 2026*.

Table of contents

Introduction 4
- What is the Semantic Common Data Model? 4
- Benefits of the Semantic Common Data Model 4
- Audiences 5
- Development 6
- SCDM relationship with HL7 FHIR 6
- Foundational PCHDCF characteristics 7
- Architecture patterns 29
- Implementation and alignment methods 36

Introduction

The Canadian Institute for Health Information (CIHI) is developing the Pan-Canadian Health Data Content Framework (PCHDCF), which defines, standardizes and models the health data required to enable connected care in Canada.

At the core of this implementation guide is the Semantic Common Data Model (SCDM) architecture product. The SCDM serves as a common data model for all implementers to use and provides the shared semantic foundation essential for consistent and interoperable implementations across pan-Canadian health care systems.

There are several different ways that individuals and organizations can adopt the SCDM. This document is not a rulebook for how each implementation should be structured; rather, it is a guide to point out some of the nuances and considerations in the architecture that may not be immediately apparent.

What is the Semantic Common Data Model?

The SCDM is a semantically rich model that represents and standardizes key health concepts and supporting information concepts, as well as describing how they relate to each other. This model, which is integrated with the Pan-Canadian Health Data Content Framework's Data Content Standard, will be a foundational component for data management and data exchange.

Benefits of the Semantic Common Data Model

The following benefits can be realized if systems adopt and align with the SCDM as the shared semantic authority.

Standardized meaning

- Establishes a single, authoritative definition of meaning for core health care concepts.
- Core concepts are defined once and reused consistently.
- Facilitates terminology alignment by anchoring standardized value sets.

Strong governance and interoperability

- Provides official, versioned definitions to support business glossaries, data standards, policy interpretation and catalogs.
- Enables semantic interoperability at scale by standardizing all core semantics — across jurisdictions, systems and over time.
- Enables reuse and serves as a foundation for interoperability components outside of the PCHDCF, such as APIs, policy-based data sharing and federated identity integration.

Scalability and flexibility

- Can be scaled up and down to meet the needs of any organization or use case.
- Can be extended and flexible without losing shared meaning, even as systems, standards and requirements evolve
- Accelerates implementations by providing a reusable semantic model that leads to deduplication of design efforts.

Research and analytics

- Allows AI models and analytics to benefit from trustworthy data that has been standardized using a common data model with consistent features.

Overall, the SCDM, along with other components of the PCHDCF, provides systems with a robust semantic foundation which enables alignment that will drive the standardization of comprehensive health records, ultimately advancing connected care and improving patient care in Canada.

Audiences

The audiences for the Common Data Model Implementation Guide are decision-makers, policy-makers, researchers, staff at health care institutions, IT staff, clinical solution vendor technical staff, experts involved in standards development and data architecture, software developers, those interested in e-health and digital health modernization initiatives, as well as anyone who would like more information about the Pan-Canadian Health Data Content Framework.

Development

An environmental scan was conducted for existing health standards produced by Canadian organizations, international organizations and various other health authorities. A model structure was established that contains key concepts from those standards, which are also documented in the Pan-Canadian Health Data Content Framework's Data Content Standard, on which the model is based. Other sources that the PCHDCF referenced can be found in the Bibliography of the Semantic Common Data Model artifact or the Business Context Model artifact. The model's structure was validated by internal and external stakeholders and was iteratively refined based on the feedback received.

The model strives to reflect both the current realities of Canada's health care systems and how they could be improved upon in a future ideal state. It remains technology-agnostic, with the inclusion of only generic data types and business metadata. Technical attributes, such as primary and foreign keys to link tables and databases, are not included; implementers will need to add further detail as required by their physical system.

SCDM relationship with HL7 FHIR

HL7 FHIR and the SCDM address two complementary aspects of interoperability. A useful analogy is that FHIR provides the suitcase used to transport the data, while the SCDM provides the shared understanding of what is inside it.

FHIR is designed to enable data exchange between systems. Through standardized APIs, resources and vocabularies, it provides a common structure for securely and reliably moving health care data. Its modular and extensible design allows it to support a wide range of systems, workflows and implementation contexts. This flexibility is one of FHIR's strengths and has contributed to its broad adoption.

However, flexibility can also introduce variability. Many FHIR elements are optional, multiplicities may allow multiple implementation choices and the same resource may be populated or interpreted differently across systems. FHIR Implementation Guides and shared profiles can reduce this variability, but they require consistent governance, adoption and alignment. As a result, systems may be technically capable of exchanging data without interpreting that data in the same way.

The SCDM addresses this semantic interoperability gap. It defines the meaning of healthcare concepts, including their attributes, relationships, value sets and governing rules. By establishing shared semantics, the SCDM helps ensure that data is interpreted consistently across systems, organizations and time. It also makes relationships explicit, allowing information to be connected across concepts and assembled into coherent longitudinal views.

FHIR and the SCDM therefore serve distinct but mutually reinforcing purposes:
FHIR moves the data; the SCDM ensures that the data is understood consistently.

Canada Health Infoway is developing the CA Core+ FHIR profiles to implement the expectations of the SCDM. These profiles constrain and guide the use of FHIR within the Canadian context, reducing unnecessary implementation variability while retaining FHIR's flexibility and extensibility. CIHI and Infoway are working together to establish precise mappings between CA Core+ and the SCDM (starting from the Canadian Core Data for Interoperability [CACDI]), ensuring that the structure used to exchange data remains aligned with its intended meaning.

Foundational PCHDCF characteristics

Association

Plain lines in the Semantic Common Data Model represent association relationships between classes. Unified Modelling Language (UML) notation is used to depict multiplicities in these relationships. This includes 0 to 1, 1 to 1, 0 to many, and 1 to many relationships.

Any such multiplicity applies to the class that is the nearest to it in a diagram. Every relationship thus has 2 multiplicities, one at either end.

The direction of a relationship is depicted by an arrow near its name.

Generalization

Generalization relationships have an arrow that points from the child to the parent.

 Generalization relationship from child to parent

This means that the child class is a special case (subclass) of the parent class.

Usage

Usage relationships have an arrow with a dotted line that points from the clinical class to the class in which this information will be stored and exchanged.

 Usage relationship from clinical concept to an class

The business view classes in periwinkle are business representations of clinical concepts and are not part of the core Semantic Common Data Model. They are there for illustration purposes.

Composition and aggregation

Composition relationships have a black diamond on the parent side of the relationship. In composition relationships, the child class cannot exist independently of the parent class.



Aggregation relationships have a white diamond on the parent side of the relationship. In an aggregation relationship, the child class can exist independently of the parent class.



Class

A class is an encapsulation of data recognized by a domain expert as a representation of a discrete concept. Examples include Person and Health Care Service Event.

1. Green classes: Classes detailed in the subject area diagram



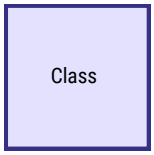
2. Orange classes: Classes detailed in another subject area diagram



3. Pink : Classes to be detailed in a future release

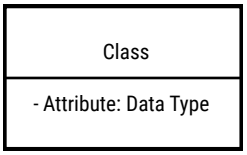


4. Periwinkle classes: Classes that are business representations of clinical concepts and are not part of the core Semantic Common Data Model.



Attribute

An attribute is a characteristic of a class, providing more specific information to distinguish it from other classes. Each attribute will have a data type and multiplicity (multiplicity = 1 will be hidden on the diagram).



Naming conventions

The definitions of attributes and classes in the SCDM describe the data concepts represented in the model and are linked to 1 or more terms in the Business Glossary.

A standard naming convention is applied across the SCDM to ensure clear understanding of classes and attributes. A few details are outlined below to aid reader understanding.

Class

- Names of association classes and 2-way intersection classes (i.e., regular, non-association classes that represent the intersection of 2 other classes) will be composed of the names of both classes they connect.

Attribute

- Attributes are typically prefixed by class-specific characteristics (generally but not necessarily the name of the class) to distinguish them from other similar attributes. However, such prefixing applies only when the meaning of an attribute involves the class to which it belongs.
  - For example, in the Patient Allergy-Intolerance class, the attribute name Patient Allergy-Intolerance Resolution Date contains the prefix “Patient,” which is a class-specific characteristic, since different patients may have different resolution dates for the same allergy or intolerance.
  - On the other hand, the attribute Allergy-Intolerance Code in the same class does not contain the class prefix (which would necessarily begin with “Patient”) since the meaning of an allergy-intolerance — product or substance — does not involve patients having that condition.

It follows from the above that

- Attributes with the same name have same meaning.

General

Long class and attribute names may be abbreviated, generally by dropping context prefixes, when the meaning is clear from the context (e.g., Health Care Provider Organization Affiliation Type Code abbreviated to Provider Organization Affiliation Type Code).

SCDM and value sets

Attributes in the SCDM that are assigned the Codeable Concept data type have their allowed values defined in one or more curated value sets.

A value set member has a code unique within a code system as well as a name and description, possibly bilingual.

The code systems are usually managed external to the implementer. Depending on the completeness of the assigned value sets, the value set reference in the attribute note may take the following forms:

- Value set: <URL1> <URL2> …
- Value set: <URL1> Additional concepts to be developed: <comma-separated list>
- Value set: To be developed: Further development is required to specify the value set (as of the time of SCDM publication).

SCDM and business rules

Business rules are explicit, specific and actionable statements that define or constrain aspects of data or its behaviour within a data persistence system.

These rules ensure data quality, patient safety, interoperability and compliance by enforcing real-world constraints and operational requirements.

All business rules defined in the framework must be followed in any fully conformant implementation. Business rules may be defined either in the UML diagrams (e.g., multiplicities, constraints) or in the Business Rules artifact. These rules are purely semantic, meaning they represent business requirements rather than technical details.

Business rules are depended on:

- Business strategy/objectives
- Business policies and/or government regulations
- Business requirements
- Business terms (business glossary)
- Concept structure as represented in the SCDM

In scope

1. Security and Privacy Rules
2. Time Based Rules
3. Data Quality
4. Aggregation
5. Cohort

Out of scope

Workflow rules

1. Admission workflow
2. Discharge approval
3. Medication ordering process
4. Referral routing
5. Appointment scheduling
6. Claims adjudication
7. Clinical decision-support workflows

SCDM and Business Glossary

One of the key products included in the PCHDCF is the Business Glossary, which provides terms, their definitions and their semantic relationships for clinical and architecture concepts used across the Pan-Canadian Health Data Content Framework. The glossary establishes a standard vocabulary to support clear communication and common understanding of terms across Canada's health systems.

While definitions from the Business Glossary underpin terms used across the framework, each class and attribute within the SCDM may also include a data dictionary term. These terms provide additional information around how a data element should be used, implementation considerations and other information.

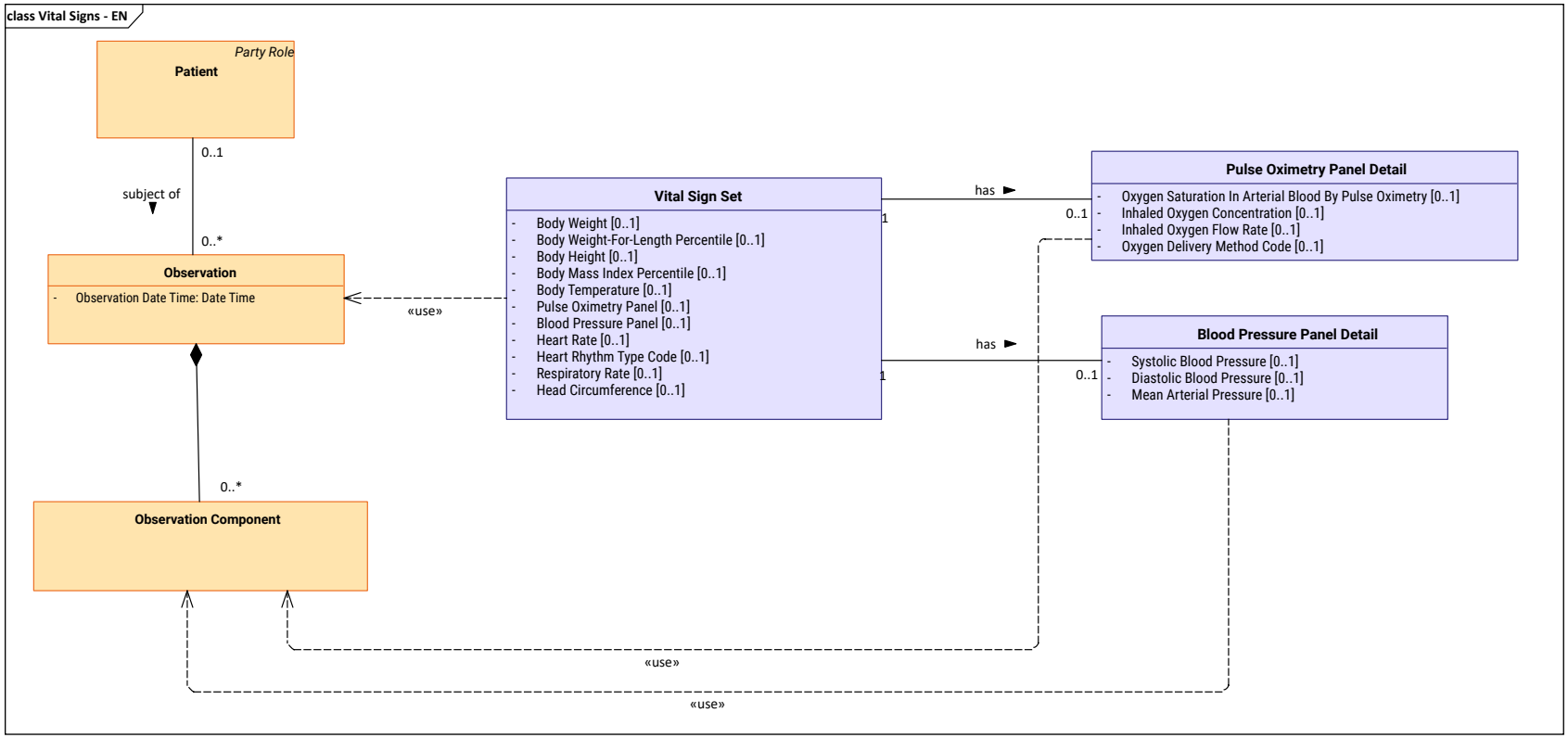
Business view

As of publication, the SCDM includes 4 subject areas that provide a business view of the PCHDCF data elements:

- Vital Signs
- Social Determinants of Health (SDOH) Assessment
- Substance Use
- Perinatal Health

These subject areas are primarily intended for a business audience, as they visually illustrate how some business concepts may share common logical classes during implementation and/or data exchange. Some details for Vital Signs, SDOH Assessment, Substance Use and Perinatal Health could be implemented using the Observation and Observation Component classes in the Observation subject area.

Figure 1 Sample of business view classes within the Vital Signs subject area



For example, in Figure 1, concepts such as Body Weight, Body Height, Waist Circumference and Blood Pressure are all considered Observation concepts and therefore can be implemented using the Observation and Observation Component classes on the left side of the diagram.

In summary, the classes to the right, in periwinkle (Vital Sign Set, Blood Pressure Detail and Pulse Oximetry Detail) are intended to serve only as a reference for these concepts. Using the example above, they help clarify which vital signs are within scope of the framework; however, they are not meant to be implemented as classes within a system.

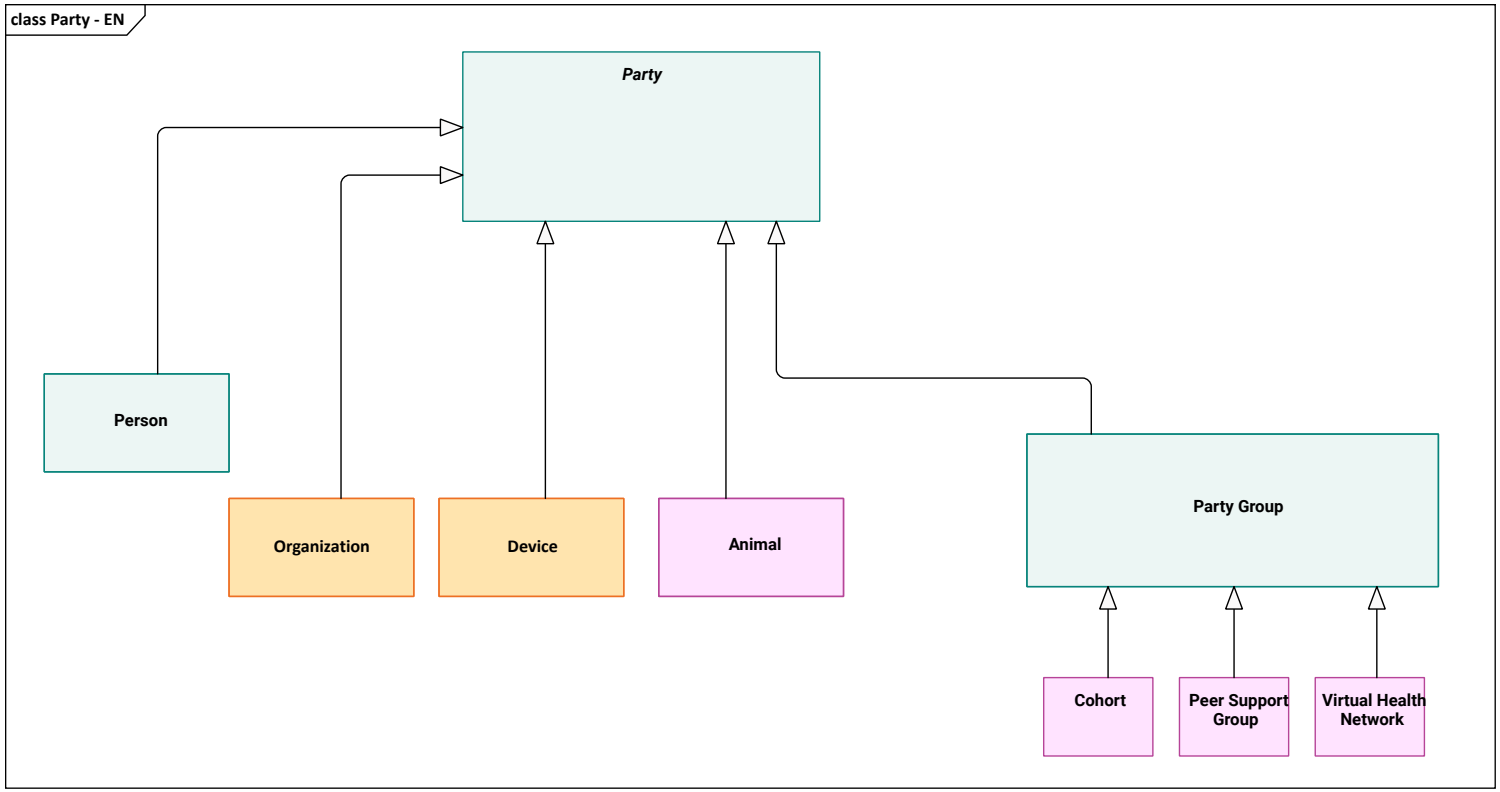
Abstract class

An abstract class is a superclass (i.e., parent of generalization relationship, whose children inherit its properties) that is not to be instantiated. It follows that the set of its subclasses in the SCDM is complete, at the time of the model's release: no other subclasses are known at that time. Thus, for each instance of the abstract class, exactly one instance of the child class is to be instantiated (inheriting the attributes and relationships of the superclass, other than the relationship from that child to said superclass), unless that child is itself an abstract class, etc.: ultimately, leaf classes (always concrete) and non-leaf concrete classes shall be instantiated.

A concrete class is one that is not abstract, i.e. that is to be instantiated. Any class that is not a superclass, i.e., not a generalization relationship parent (not a superclass) is a concrete class by definition. For a concrete superclass, the set of its child classes is incomplete, i.e., a concrete superclass may have instances that do not correspond to any of its child subclasses. Therefore, a concrete superclass must have discriminator attribute(s), such as <class name> Type Code: Codeable Concept (preferably) or equivalent (e.g., name and purpose for party groups: see diagram below) that categorize instances not corresponding to any child class.

In Sparks EA, abstract classes names are shown in italics.

Figure 2 Sample of abstract and concrete classes within the Person Information subject area



- Party is an abstract class (name displayed in italics): in this release of the SCDM, it is assumed that it has no child subclasses other than the ones shown in the model: i.e., Person, Organization, Device, Animal, Party Group.
- Party Group is a concrete class: in this release of the SCDM, it is assumed that not every party group is necessarily one of the children shown in the model: i.e., Cohort, Peer Support Group or Virtual Health Network.

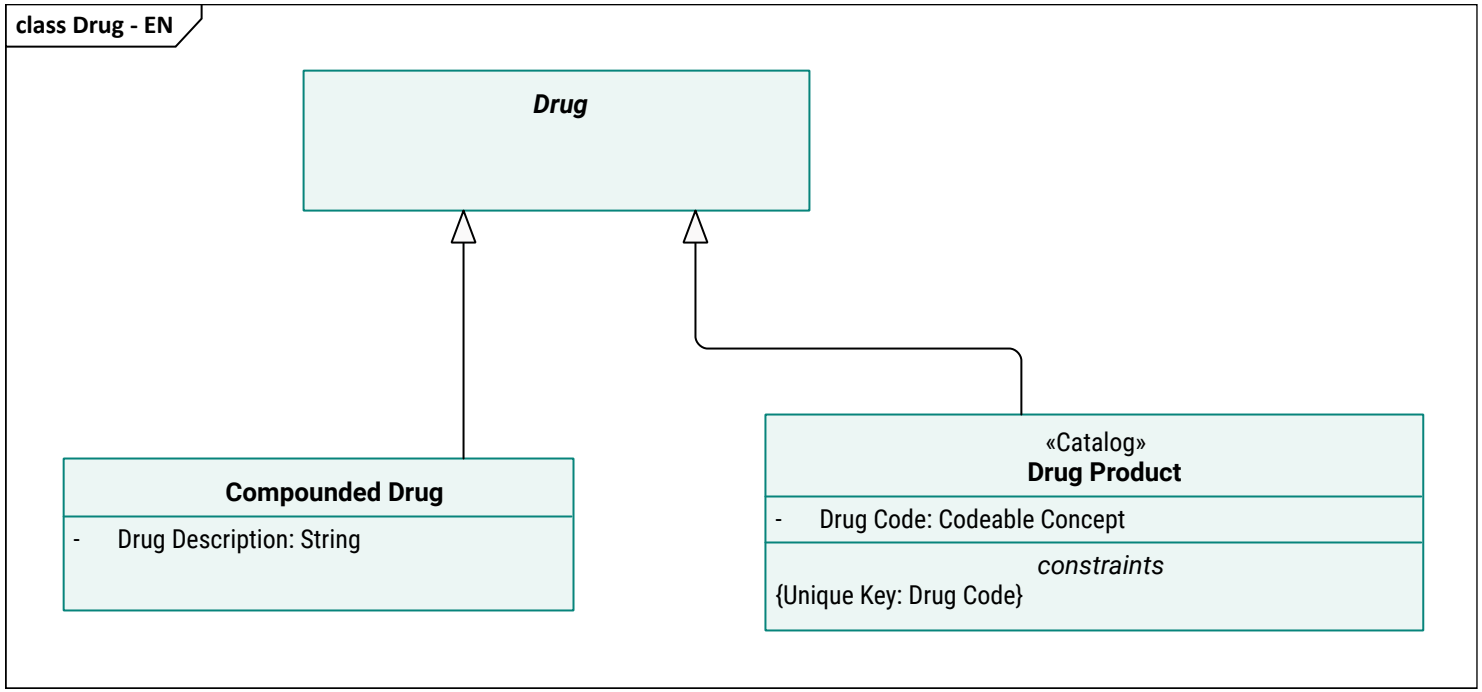
Remarks:

- The concept of SCDM abstract class is implementation-independent: in a particular implementation, tables or other physical objects may be created (or not) for classes representing abstract classes.
- The concept of SCDM abstract class is time-varying: a concrete class in this release of the SCDM may become abstract in a future release, once more is known about possible subclasses. Eventually, in a mature model, all superclasses should be abstract.

Catalog classes

The Catalog stereotype in EA identifies classes which represent externally managed stable or slowly-changing sets of reference information: e.g., Drug Product, Health Care Service (as part of a jurisdictional service registry).

Figure 3 Sample of Catalog classes within the Drug subject area



The set of drug products is represented and managed in a standard knowledge bases such as CCDD or Health Canada NPN: it is a Catalog class. On the other hand, a Compounded Drug is a custom medication (e.g., prepared at dispense time or in a hospital) which is not part of an externally managed catalog.

Constraints

Constraints in the SCDM define the rules and conditions that data must satisfy to uphold standardized data governance policies. These constraints support key data quality dimensions, ensure semantic consistency and enforce metadata integrity across the model.

By embedding these rules within the logical design, the model helps maintain meaningful relationships between data elements, aligns with business and regulatory requirements, and provides a robust foundation for trustworthy and interoperable data use across Canada's health care systems.

Constraints represented in the SCDM

1. **Unique key:** A unique business key, shown in the SCDM at the bottom of a class, consisting of one or more non-technical attributes or relationship ends that uniquely identify a class instance. In place of attributes, a unique key may also contain expressions such as "XOR(attribute, attribute, ...)" or "attribute if specified." The SCDM does not include technical unique keys; therefore, not all classes have a unique key constraint. At implementation, those business keys may correspond to natural keys, as distinct from system-generated primary keys, which are out of SCDM scope.
2. **Relationships:** Define how classes relate. Governed by relationship-end multiplicity (e.g., 1-to-many, shown as 1..*). There may also be explicit constraints (validity rules) shown for the relationships.
3. **XOR (exclusive OR):** Only 1 of a set of optional attributes or relationships must be present. Used to enforce mutual exclusivity in the data model.
4. **Inclusive OR:** 1 or more of a set of optional attributes or relationships must be present.
5. **Must-populate attribute property:** Shown as [1..1P] or [1..*P]. This semantic constraint indicates that an attribute must be populated with either a valid value or an absent value.
    - **Mandatory property (attribute multiplicities [1], [1..*]) is indicated in the SCDM only where it applies in all possible or known use cases ("mandatory-by-definition"). If an attribute or a relationship end is mandatory for some use cases and optional for others, it has optional multiplicity in the SCDM.**

6. **Syntactic constraints:** Only data types are specified. No lengths, formats or regex patterns are enforced in the SCDM. These are typically added during physical model implementation.
7. **Semantic constraints:** Ensure contextual meaning of data elements. They help align the model with business meaning and real-world concepts using a coded concept approach.

Constraints not represented in the SCDM

These constraints are typically left for the physical model or application level to enforce.

- **Check:** Value range checks or conditional expressions are not specified. These are deferred to physical or application layers.
- **Default:** Default values are not defined in the SCDM. They are determined at the physical model level, influenced by semantic rules.
- **Not null:** Not explicitly enforced. However, it can be inferred from mandatory attribute or relationship-end multiplicity ([1], [1..*]).
- **Primary key**: The SCDM doesn't define primary keys. Decisions between natural and surrogate keys are made during physical modelling.
- **Foreign key:** Not defined in the SCDM. Relationships exist conceptually but foreign key constraints are implemented in the physical model.
- **Triggers:** Not part of the SCDM. May be implemented in the database to enforce rules (e.g., XOR) during physical design.
- **Stored procedures/functions:** Not included in the SCDM. Recommended to apply complex business rules at the application or workflow level.

Recommendations for implementers

- When working with the SCDM, do not expect traditional DBMS constraints to be defined upfront.
- Use the SCDM to understand data relationships, uniqueness requirements and business semantics.
- Primary key, foreign key, check and default constraints are expected to be defined by physical data modellers.
- Compliance with XOR rules and support for data-absent reasons are mandatory, as both have significant implications for validation logic.
- Compliance with explicit business rules, as defined in the document with that name, is mandatory, as it has significant implications for validation logic.

Use case implication

This framework is useful not only for health care interoperability use cases, but for decision-support systems, including descriptive, diagnostic, prescriptive and predictive analytics where:

- **Data completeness and semantics** are critical (e.g., clinical decisions, regulatory reporting);
- **Business and regulatory rules** must be documented and preserved from logical to physical models; and
- **Physical implementation decisions** are deferred, allowing flexibility and technology independence.

SCDM data types

Technology-agnostic complex data types

In a CIHI semantic model (designed as Unified Modelling Language [UML] model), attributes are technology-agnostic, attribute types are specified independently of any implementation technology, including programming languages (e.g., Java, C#, or Python) and database management systems (e.g., SQL).

This approach promotes platform independence and ensures that the model captures the conceptual semantics of the application domain rather than implementation-specific details.

All attributes in the SCDM have a complex data type, i.e. one that consists of multiple parts. Each part may be either another complex data type or a primitive technology-agnostic type such as string or number. Complex data types improve model abstraction, promote reusability and support semantic consistency across the model.

Primitive data types are written in lower case (e.g., integer), whereas SCDM complex data types are in initial caps (first letter only in upper case; e.g., Integer, which has primitive integer data type as one of its parts).

Primitive data types are generic, technology-agnostic and based on the ISO 11404 standard, except for file and polygon. The latter is based on Statistics Canada's GIS representation standard.

Many complex data types such as Codeable Concept, Quantity, Range, Ratio, Observation Period and Sampled Data, as well as Identifier, are generally based on HL7 FHIR R4, although simplifications or variations are possible.

Complex data type parts are minimally constrained to support the maximum number of use cases (as is the case in FHIR). Additional attribute-specific constraints on complex data type parts, i.e., constraints applicable only to a specific attribute of that type, shall be specified in the data dictionary (DD) description of the attribute (i.e., in the attribute's EA Notes field); e.g.,

- Value set bindings for codeable concept parts;
- Constraints on values of non-codeable concept parts (e.g., decimal part = 0 for a Quantity part, > 0 or > 0.0 for any numeric part, Date only for Date Time part);
- Making an optional part mandatory for the attribute only (e.g., Start Date Time is mandatory for a specific Period attribute).

If the attribute is mandatory, i.e., has multiplicity [1] (not shown in EA diagrams) or [1..*], then all the data type parts that are mandatory for the attribute must be populated with a valid value, i.e. a value consistent with the data type of the part and that is not a data-absent reason (DAR). Optional parts of a mandatory attribute need not be populated.

If the attribute has must-populate property, represented as multiplicity [1..1P] or [1..*P], then all the data type parts that are mandatory for the attribute must be populated with either a valid value or a DAR. Optional parts of a must-populate attribute need not be populated.

Mandatory parts of an optional attribute must be populated with a valid value whenever the attribute is populated. Likewise, must-populated parts of an optional attribute must be populated with a valid or absent value whenever the attribute is populated.

Common data type parts

All or nearly all data types have the following optional attribute parts:

- **Free text description:** string: May be entered **in addition to the other attribute data type part(s)**, including the common Absent Value Code part, to provide additional unstructured information. **May not be entered as the only data type part of an attribute, nor instead of the other attribute data type part(s)**. Must be semantically consistent with — i.e., value may not contradict — the value(s) of the other data type part(s) for the given attribute.
- **Absent Value Code:** member of the generic DAR value set. May be entered as **the only data type part of an attribute or instead of the other specific attribute data type part(s)**. **May not be entered in addition to any data type part other than free text description**. URL of generic DAR value set: https://terminology.hl7.org/CodeSystem-data-absent-reason.html.

This table describes, for each SCDM data type, the parts that are specific to it. Those are in addition to the above optional common parts, which every data type has (i.e., every attribute may have an Absent Value Code instead of its specific data type part value(s), and/or a free text description in addition to either the Absent Value Code or one or more specific data type part value(s).

Complex data type	Meaning	Parts	Comments
Integer	A non-decimal number (e.g., 3)	• Value: integer (primitive type)	May add constraints or subtypes in the next release to represent positive (>0) and non-negative (>=) integers
String	A sequence of characters (may include numbers or other symbols; e.g., “H4V2W3”)	• Value: string (primitive type)	N/A
Text	Long text, possibly mapped to a CLOB relational physical data type	• Value: long string / text (primitive type)	N/A
Indicator	A Boolean: true or false	• Value: Boolean (primitive type)	N/A

Complex data type	Meaning	Parts	Comments
Codeable Concept	A member of a value set, represented by a code unique within a code system, as well as the corresponding bilingual name and (optionally) description.	Data Type Class Parts: not populated in an attribute of this data type but rather specified in the attribute's data dictionary (DD) description (aka bound to the attribute) – in Sparx EA, specified in the Notes field of the attribute. Valid Value Set Reference(s), 1 or more; each may be a URL or "To be developed." See Value Set(s) section above for further details. These value set(s) contain valid, i.e., allowed values of the attribute; on the other hand, the absent values are specified in the generic DAR value set bound to the common Absent Value Code part. Data Type Instance Parts: populated, if applicable, as part of the value of an attribute having this data type. • Value Code: string (primitive type). Represents a valid attribute value, i.e., a non-DAR member of a bound valid value set (not a member of the generic DAR value set). • Value Code System URL: string, URL format: only needs to be populated if >= 1 members of the bound valid value set(s) match the specified code but have different code system references. Otherwise (almost always) not specified.	N/A
Date	1 or more of the following: day, month, year.	• Value: date (primitive type). If day is specified, month and/or year must also be specified. If month is specified, year must also be specified.	This data type shall be assigned only to attributes for which time is known to be inapplicable in all possible use cases.
Time	1 or more of the following: hours, minutes, seconds, fractions of seconds	• Value: time (primitive type) • Time Zone Code: Codeable Concept. Value Set: To be developed.	This data type shall be assigned only to attributes for which date is known to be inapplicable in all possible use cases. This data type should be used for timepoint attributes. Duration attributes are generally integers or decimals with a temporal unit of measure.
Date Time	Date followed by time; date is mandatory, time is optional	• Value: Date part followed by Time part (see above); Date part is mandatory. The individual parts (Date, Time) may not be Free Text Description, nor Absent Value Code.	This data type shall be assigned only to attributes which are known, in all possible use cases, to have at least a date part (1 or more of day, month, year: see Date data type above), and, always or sometimes, a time part as well.

Complex data type	Meaning	Parts	Comments
Period	Date–time interval	• Start Date Time: Date Time • End Date Time: Date Time Mandatory: XOR((Start Date Time OR End Date Time), Absent Value Code) Start Date Time <= End Date Time if both are specified. The above specific data type parts, may not themselves be Free Text Description nor Absent Value Code.	A Period attribute's DD description (EA note) may specify the following bindings which, only for that attribute, override the generic properties of the data type (i.e. the properties to which every attribute of this type defaults unless otherwise specified): • Start and/or End is a Date (no time part): overrides the generic Date Time data type. • Start and/or End is mandatory: overrides the generic optional property of these parts.
Identifier	A generally alphanumeric value and related information which, taken together, uniquely identifies something an instance of some concept (e.g., a person's health care number or driver's licence number, an organization's business identifier, an encounter identifier)	• Value: string; mandatory (primitive data type). Generally alphanumeric; uniquely identifies an instance of a concept within an identifier system. and which is not necessarily meaningful although the concept is. • System: string. Specifies the namespace within which Identifier.Value is unique - generally a URL or URI. Should be populated if known. • Type Code: Codeable Concept. Examples: MRN, health card, passport, driver's licence, resource identifier (or a subtype thereof, such as encounter identifier). • Type Description: string. To be used only when Type Code is known but not in the value set(s) bound to an identifier attribute's type code Description: string. • Use Code: Codeable Concept. Examples: official, usual, temporary, secondary, old. Value set: https://hl7.org/fhir/R4B/valueset-identifier-use.html • Effective Period: Period Mandatory: XOR((Value AND (XOR Type Code, Type Description), Absent Value Code) The above specific complex data type parts (Type Code, Use Code, Effective Period) may not themselves be Free Text Description nor Absent Value Code.	The identifier data element (e.g., social insurance number) shall have clinical/business relevance, although its values generally do not (e.g., may have a fully or partially random or anonymized value).

Complex data type	Meaning	Parts	Comments
Sampled Data	Multi-part observation-sample data element	• Origin: Quantity; mandatory (zero/baseline value and unit of measure) • Period milliseconds: decimal (>0.0); mandatory • Factor: decimal (>0); mandatory • Lower limit: decimal • Upper limit: decimal • Dimensions: integer (>0); mandatory • Data: string; mandatory Mandatory parts, unless attribute of this type = Absent Value Code: Origin, Period, Factor, Dimension, Data Lower limit <= Upper limit if both are specified	Used only in Observation context.
Decimal	A decimal number with 1 or more significant digits (e.g., 3.45)	• Value: decimal (primitive type)	May add constraints or subtypes in the next release to represent positive and non-negative decimals.
Quantity	A decimal value together with its unit of measure	• Value: decimal (primitive type) • Unit of Measure Code: Codeable Concept (may not be Free Text Description, nor Absent Value Code)	May add constraints or subtypes in the next release to represent positive and non-negative quantities. May also add subtypes to represent duration, volume, length, distance, etc., each with a specific Unit of Measure Code value set.
Range	An interval of quantities, with at least one end (lower and/or upper) specified	• Lower Limit: Quantity • Higher Limit: Quantity Mandatory: XOR((Lower Limit OR Higher Limit), Absent Value Code) Lower Limit <= Higher Limit if both are specified. The above specific Quantity data type parts may not themselves be Free Text Description nor Absent Value Code.	N/A

Complex data type	Meaning	Parts	Comments
Ratio	Consists of 2 quantities: a numerator and a non-zero denominator The value = numerator ÷ denominator	• Numerator: Quantity: mandatory • Denominator: Quantity (non-zero): mandatory Mandatory: XOR((Numerator AND Denominator), Absent Value Code) Numerator <= Denominator if both are specified. The above specific Quantity data type parts may not themselves be Free Text Description nor Absent Value Code.	N/A
Life Date Time	Represents either a precise date–time or fuzzy reported timepoint of a generally-clinical life event, such as the onset of an allergy or intolerance.	• Value: Date Time • Age: Quantity (value and unit of measure code), >= 0 (0 at birth). • Age Range: Range (with units, e.g. Years), 0 <= min <= max. • Time Period: Period • Description: string Mandatory unless Absent Value Code: XOR(Value, Age, Age Range, Time Period) OR Description The above specific complex data type parts may not themselves be Free Text Description nor Absent Value Code.	N/A
Polygon	A sequence of coordinates to denote the boundary of a geographic shape	• Value: polygon (primitive type)	Polygons in the current model are encoded based on Statistics Canada's standard.
File	A document, generally binary (e.g., an image), possibly with metadata, such as an X-ray or MRI scan	• Value: base64 (primitive type) • File type: Codeable Concept (may not be Free Text Description nor Absent Value Code)	Reference standard for base64 binary: RFC 4648
Binary	A binary data element (e.g., QR code) without metadata.	• Value: binary (primitive type)	Depending on the implementation framework, the variety of binary data type most appropriate to a specific attribute can be chosen (e.g., fixed or variable length binary in SQL; read-only (bytes) or read-write (byte array) in Python).

Point-in-time architecture

A point-in-time architecture ensures that a system can reconstruct the state of data at any past date and time, rather than only storing the current state. In other words, a point-in-time architecture implementation approach ensures that record change history is preserved in systems. The model assumes that any legislative requirements for point-in-time architecture across different jurisdictions are handled by the source software systems used to capture and manage clinical information. Therefore, these requirements do not need to be explicitly represented within the model's attributes.

Effective and expiry date/date time (what they mean)

Effective and expiry dates and date times (as well as start and end dates/times) are the 2 parts of Period complex data type; Period attributes are business attributes in the model. Those attributes indicate when an instance of the class to which they belong is valid or active (e.g., Provider Practicing Affiliation Period attribute).

How to understand Period values:

- No dates entered: The class is considered active or valid forever.
- Only effective or start date populated: The class instance becomes active or valid starting on that date and is valid until further notice.
- Only expiry or end date populated: The class instance is valid until that date and not after.
- Both dates are populated: The class instance is valid between those dates.

Architecture patterns

Party Role

A cornerstone of the SCDM design is the use of a person-centric approach to ensure precise representation of data about patients and other actors as required to provide high-quality health care services.

The person-centric approach was expanded into the Party-In-A-Role industry-standard modelling pattern, depicted in the Person Information Diagram. This diagram allows for the representation of any actor such as person, organization, group or animal to play one or more roles such as Patient, Health Care Provider, Health Care Organization, etc. This

representation also supports privacy and security by adding attributes based on who the actor is or based on what role the actor is playing. This approach also adds flexibility, scalability and stronger analytics support to the SCDM In particular, parties and their roles will also be central to future clinical resource management information modelling, such as for bed management, device management, inventories, hospital orders, etc.

Data Absent Reason

Missing value reasons are used to document additional information when clinical or administrative data is intentionally not provided. Missing value reasons can be represented either by a context-specific concept (e.g., “no known procedures”) or with a generic absent reason (e.g., “asked but unknown”). Context-specific concepts will be associated with a data element (e.g., Procedures) and, where available are included in the terminology foundation for that data element. Generic absent reasons are not associated with any specific data element and may be used throughout the Pan-Canadian Health Data Content Framework as needed by implementers and in accordance with requirements set in the SCDM. The SCDM identify data elements where absent reasons are not allowed, represented as mandatory attributes ([1] or [1..*]). Implementers may choose the full set or a subset of absent values to be used with the clinical or administrative value sets. In some cases, relevant missing value reasons are incorporated directly into the value set where those missing value reasons are recommended for all implementations.

The following table contains the proposed concepts that can be used to indicate the reason data values are missing from data elements (e.g., clinical or administrative records). This proposed value set includes concepts, definitions, contextual descriptions and clinical-facing displays for optional implementation.

The PCHDCF recommends that implementers use the absent value codes from this table, which is a subset of the HL7 DAR value set specified for the SCDM attributes: https://terminology.hl7.org/CodeSystem-data-absent-reason.html

HL7 FHIR DataAbsentReason					HL7 V3 NullFlavor	
Code	Label	Definition	Contextual description for when to use	Client-facing display (English)	Code	Label
unknown	Unknown	The value is expected to exist but is not known.	A response is unavailable and the reason is unknown (e.g., unconfirmed whether the question was asked or not).	Unknown	UNK	unknown
asked-unknown	Asked But Unknown	The source was asked but does not know the value.	The question was asked, but the person does not know or is unsure.	Asked but unknown	ASKU	asked but unknown
asked-declined	Asked But Declined	The source was asked but declined to answer.	The question was asked but the person prefers not to answer.	Prefer not to answer		
temp-unknown	Temporarily Unknown	There is reason to expect (from the workflow) that the value may become known.	A response is expected but not currently known.	Temporarily unknown	NAV	temporarily unavailable
not-asked	Not Asked	The workflow didn't lead to this value being known.	The question was not asked or not discussed at this time.	Not discussed	NASK	not asked
not-performed	Not Performed	The value is not available because the observation procedure (test, etc.) was not performed.	A response is unknown because the procedure or test was not performed.	Not performed		
error	Error	Some system or workflow process error means that the information is not available.	A value is not available due to a process or technical error.	Error		
not-applicable	Not Applicable	There is no proper value for this element (e.g., last menstrual period for a male).	There is no applicable response (e.g., last menstrual period for a male).	Not applicable	NA	not applicable

A **Data Absent Reason (DAR) class** has been created for each subject area that is core in a patient's clinical record: Allergy and Intolerance, Patient Health Concern, Procedure, Immunization, Medication Statement, Family History. This class serves to specify data-absent reasons (DARs), which are members — coded concepts — of the generic external DAR value set.

Thus, the SCDM contains a Patient Allergy or Intolerance Data Absent Reason class, a Medication Statement Data Absent Reason class, etc., each one shown in the corresponding subject area diagram.

A DAR class has 2 sets of attributes:

- >= 0 context attributes, all having optional multiplicities.
- >= 1 (usually = 1) DAR
- attributes: at least 1 is mandatory.

The combination of context attributes specifies the optional applicability context of a DAR. For instance, Allergy or Intolerance Code is a context attribute in the Allergy or Intolerance DAR class. If populated, the value of the DAR attribute, Allergy or Intolerance Data Absent Reason Code, applies to a specific allergy/intolerance code (e.g., patient declined to answer whether they are allergic to Penicillin-G). If none of the context attributes are specified, the DAR applies to the entire data category (e.g., patient declined to answer whether they have any allergies or intolerances).

A diagram with a DAR class has 2 entry points: the main class (e.g., Patient Allergy or Intolerance) and the DAR class (e.g., Patient Allergy or Intolerance Data Absent Reason). For the same context (e.g., a specific allergy/intolerance code, or all possible allergies or intolerances of a patient), either the DAR class or the main class is populated, not both. Thus, valid information and queries against it can be separated from DARs.

- Remark: a DAR is distinct from a no-known concept. For instance, "unknown" as Allergy or Intolerance DAR (e.g., it is unknown whether a patient has allergies) is different from the no-known concept that it is known that the patient has no (reported or recorded) allergies. No-known concepts are generally part of a value set containing valid values; they are not members of the generic DAR value set.

Provenance

Provenance refers to information about the origin and history of an asset (also known as entity in World-Wide Web Consortium (W3C) terminology) — whether physical, digital, conceptual or procedural. It includes details about the activities and agents involved in the creation, transformation or usage of the asset. Provenance supports assessments of assets in terms of: Reliability; and Trustworthiness.

SCDM implementation

In scope

The Provenance subject area focuses specifically on data provenance, which is a subset of general provenance.

- It documents the origin and history of data, capturing how it was created and managed across systems.
- It provides a standardized record describing how activities and agents have influenced a specific asset.

Out of scope

The following are outside the scope of the Provenance subject area:

- Use of standardized specifications and frameworks to collect and query provenance data, including mechanisms for detecting inconsistencies or update failures.
- Storage, immutability and traceability solutions (e.g., blockchain versus centralized implementations), including performance and scalability concerns.

Approach

The UML model adopts the 5W1H pattern (what, who, where, why, when, how) to guide modelling efforts and ensure consistency in metadata and semantics across related systems.

- What: Refers to the asset — an independent, identifiable unit of information.
- Who: Identifies the agent, such as a person, organization, device or service involved in the activity.
- Where: Specifies the location where the activity occurred.

- Why: Provides the provenance reason or purpose behind the activity. Provenance improves data trustworthiness and reliability, while data quality aspects are managed within the broader provenance framework.
- When: Captures time-related aspects, including when an asset was created or used, when an activity occurred and the duration of the agent's involvement.
- How: Describes the activity or process through which the asset was produced or altered.

When to implement

Some other subject areas (e.g., "asserted by" and "recorded by" from Patient Health Concern or [Clinical Notes] Note Author) already include data elements that track the origin and modifications of data. In such cases, these existing elements should be preferred over the data captured in Provenance subject area.

Use the Provenance subject area when

- Additional provenance details are needed beyond what's available in other subject areas; and
- An explicit, standardized record of provenance is required for interoperability, traceability or audit purposes.

How to implement it

1. **Define provenance requirements**

   Implementers shall decide what must be tracked: author, organization, source system, timestamp, reason, transformation process, device, consent context and/or target record.

2. **Attach provenance to key clinical and administrative program packages.**

   e.g.: Party/Patient, Observation, Service Request, Medication Request, Diagnostic Report, Patient Encounter, Consent, Document Reference.

3. **Use standard models**

   Recommended to standardized using:

   - FHIR Provenance for data lineage: who created/updated/transmitted a resource.

4. **Capture provenance at every Healthcare Event**

   Record provenance when information is:

   - Created by a clinician
   - Imported from another hospital
   - Transformed between protocols from (e.g., HL7 v2/CDA/X12 into FHIR)
   - Information is corrected or reconciled
   - Event is generated by a device, lab system or AI tool

5. **Use uniqueness and versioning**

   Although not specified in semantic model, provenance requirements may introduce needs for versioning and unique identification. Every asset should have stable identifiers, timestamps and version history.

6. **Create a provenance store**

   Store provenance asset instances either:

   - Inside the internal repository as Provenance resources
   - In a separate immutable provenance repository
   - Or in a hybrid model linked by enterprise/pan - Canadian IDs

7. **Secure and govern access**

   Provenance data may reveal sensitive information, such as clinician identity, patient movement, organization relationships or care context, so apply access control, encryption, retention rules and audit monitoring.

Main challenges

1. Interoperability: Different EHRs, labs, HIEs, devices and legacy systems may represent provenance differently, FHIR is recommended to implement interoperability standard.
2. Completeness: Provenance is only useful if captured consistently. Missing timestamps, authors or source system IDs weaken trust.
3. Legacy systems: Older HL7 v2, CDA, paper, fax and manual workflows often lack rich provenance metadata.
4. Privacy and consent: Provenance itself can contain protected health information or reveal sensitive care patterns.
5. Trust: Provenance records must be tamper-resistant; otherwise, they cannot support legal, clinical or compliance use.

6. Data transformation: When data is mapped, normalized, deduplicated or reconciled, it is hard to preserve both original and transformed lineage.
7. Governance: Organizations must agree on who owns provenance, how long it is retained, and how disputes or corrections are handled.

Implementation and alignment methods

PCHDCF may be used in various ways:

- As a reference model
- As an accelerator to design any data model
- As a standard for data exchanged across system boundaries.
- To align existing data assets where opportunities for standardization exist.
- As a learning resource for understanding the Canadian healthcare system and for training AI tools, software developers, data analysts and related professionals

The model provides a common, shared framework of concepts, definitions and relationships to guide understanding, communication and design that can be used, reused and built upon today and in the future.

The following is a series of suggestions for how systems could start aligning to the PCHDCF:

- Map local data models to corresponding SCDM classes and their attributes, definitions and relationships, in order to establish semantic alignment.
  - Explore and understand the SCDM, starting with the Canadian Core Data for Interoperability (CACDI) model, followed by models of the subject area in which you're interested.
  - Identify and document gaps and overlaps.
- Use the SCDM as a reference when designing new systems or features, or if there are opportunities to redesign for alignment.
  - Base the new design of new systems on the SCDM structures and semantics.
  - Follow the SCDM definitions, relationships and constraints/rules as the default baseline while extending for specific needs as necessary.
  - Note that the SCDM is technology-agnostic, and while it should be used to guide technical implementations, implementers are expected to add technical aspects according to their context (e.g., reference tables, primary keys, foreign keys).

- Start planning to adjust local implementations (as makes sense in specific circumstances) to conform to the SCDM.
  - Use the SCDM to guide and plan system updates and enhancements.
- Implement standard terminologies and value sets.
  - Start using the value sets from the standard.
  - Ensure proper coding of the data according to the standard or provide clear mapping between locally used value sets and the ones in the model.
- Leverage reusable interoperability components and patterns.
  - The number of reusable interoperability components is expected to continuously grow in future releases.
- Integrate with the Business Glossary.
  - Ensure business terms used in the implementation are aligned with the standard Business Glossary linked to the SCDM.
  - Use these standardize terms to drive shared understanding among technical and business teams.
- Increase awareness of the SCDM among your teams so that they are familiar with the concepts and components.
- Provide feedback to the Connected Care team at CIHI in terms of priorities and tools we can provide to support adoption.

cihi.ca

98983-0826