

CDMC Info-model (ver.1.1)を探る

(CDMC: Cloud Data Management Capabilities) フレームワーク検討の一環

日付： 2023年9月27日
最終更新日： 2023年9月27日
インフォラボ游悠 (PB)
中岡 実

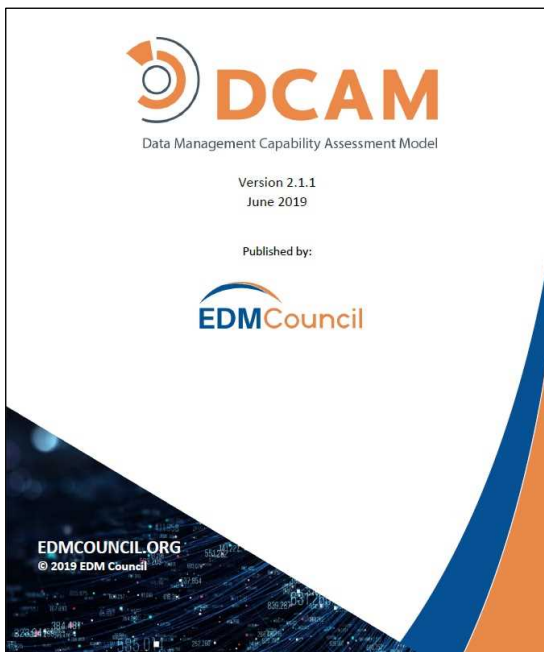
備考： 本資料内容は、EDM Councilにより整理し開示されている以下の内容を元に調査作成したものです。
参照資料： 1. CDMC Information Model Controls Tests Mappings V1.1(Oct., 2022)
2. CDMC Controls Procedures and Test Specifications V1.1.1(Oct.,2021)
3. CDMC Key Controls and Automations V. 1.1.1 (Sept., 2021)



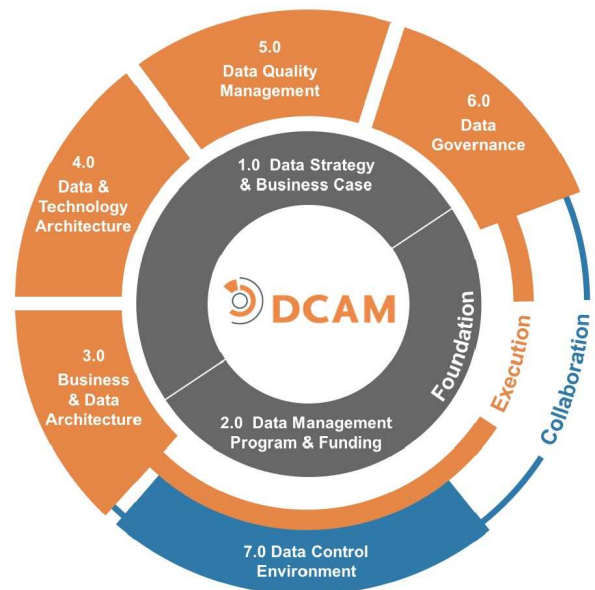
はじめに： EDMC(Enterprise Data Management Council - Data Management Capability Assessment Model (DCAM)

Web紹介ページ： <https://edmcouncil.org/frameworks/dcam/>

・ DCAMの改訂版はV2.2がリリースされている（2020年4月時点）。但しOnlineでの会員提供となり無料入手不可。



DCAM V2.1.1: 2019年6月 (176頁)



DCAM で定義するコンポーネント

8 Components of the Data Management Capability Assessment Model

Component 1.0

Defines the vision and the purpose of the data management program – why is data management important

Component 2

Describes the organizational structure and funding model of the data management program

Component 3.0

Data architecture is the design of information content, driven by business architecture (requirements), aligned to real-world objects and entities

Component 4.0

Technology architecture addresses the physical implementation of data management (platforms, DBs, tool), in collaboration with the business and the data management office



Component 5.0

Deliver data that is trusted and fit-for-purpose, where users have confidence that the data is what they need, without reconciliation

V2.2で追加

Component 8.0
Formalize how the Analytics activities of an organization are structured, executed, and managed and ensure that they are aligned with the DM activities.

Component 6.0

The rules of engagement for data management, focused on the implementation of policies, standards and operational procedures necessary to ensure that stakeholders behave

Component 7.0

A true control environment cannot be achieved until all the data management capabilities are operating collaboratively and in unison across the enterprise

DCAMモデル(V2.2) 機能(ケイパビリティ)項目段階評価項目一覧

●DCAMモデル(V2.2)の評価ディメンション機能項目段階の一覧を以下に示す。また機能項目の下段に位置するサブ機能数を合わせて示す。

Component (階層)	サブ機能数
1.0 Data Management Strategy & Business Case	小計13
1.1 The Data Management Strategy (DMS) is Specified and Shared	5
1.2 The Data Management Business Case is Defined	5
1.3 The Data Management Vision is Defined	3
2.0 Data Management Program & Funding Model	小計28
2.1 The Data Management Program (DMP) is Established	5
2.2 The DM Funding Model has been Established, Approved, and Adopted by the Organization	3
2.3 The Data Management Organizational Structure is Created and Implemented	3
2.4 The Roadmaps for the DMP are Developed, Socialized, and Approved	4
2.5 Data Management Process Excellence Program is Established	3
2.6 Stakeholder Engagement is Established and Confirmed	3
2.7 Communications and Training Programs are Designed and Operational	3
2.8 The DMP is Measured and Evaluated Against Business Objectives	4
3.0 Business & Data Architecture	小計14
3.1 Data Architecture (DA) function is established	3
3.2 Business Architecture (BA) is Integrated with Data Architecture (DA)	4
3.3 Identify the Data	3
3.4 Define the Data	4
4.0 Data & Technology Architecture	小計10
4.1 Technology Architecture (TA) is defined in support of the data management initiative	5
4.2 DM Technology Tool Stack is Identified and Governed	3
4.3 Operational Risk Planning is in Place	2

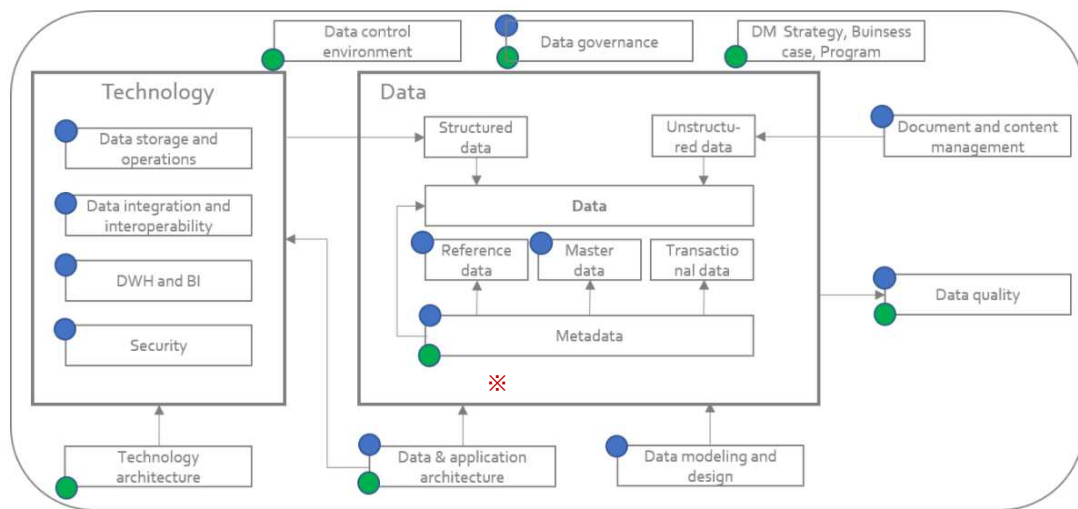
Component (階層)	サブ機能数
5.0 Data Quality Management	小計12
5.1 Data Quality Management (DQM) is Established	4
5.2 Data is Profiled and Measured	3
5.3 DQ Issues are Remediated	2
5.4 DQ is Monitored and Maintained	3
6.0 Data Governance	小計21
6.1 Data Governance (DG) Function is Established	4
6.2 Policy and Standards are Written and Approved	5
6.3 Govern the DM Program	4
6.4 Govern the Data Structure	2
6.5 Govern that the Data is Fit-for-Purpose	3
6.6 Govern the Data Ethics	3
7.0 Data Control Environment	小計9
7.1 Data Control Environment (DCE) is Evidenced	3
7.2 Cross-organization Control Function Collaboration	3
7.3 Data Risk is Managed	3
8.0 The Analytics Culture and Education Needs are Managed	小計30
8.1 The Analytics Function is Established	6
8.2 Analytics is Aligned with Business and Data Management Strategy	4
注1 8.3 Analytics is Aligned with Data Architecture	4
8.4 Analytics is Aligned with Data Quality	2
8.5 The Analytics Platform is Designed and Operational	5
8.6 Model Operationalization is Established	5
8.7 The Analytics Culture and Education Needs are Managed	4

サブ項目数合計: 137

注1 8.0のコンポーネント項目は、Ver.2.2(2020)で追加された

【補足】DMBoK2とDCAMモデルの比較について

– DMBoK2とDCAMについての要素比較を実施（データマネジメントのメタモデル（主要構成要素）を使用）。以下図参照。
 詳細： <https://datacrossroads.nl/2018/12/02/data-management-metamodels-damadmbok2-dcam/>



DAMA-DMBOK 2

● DAMA-DMBOK metamodel ● DCAM metamodel

※DCAM V2.1資料では、「Business & Data Architecture」と表現されている

DAMA-DMBOK 2 defines the metamodel of data management in the form of DAMA Knowledge Areas, represented in the DAMA Wheel form. This model of data management is industry agnostic. In my article ['DAMA-DMBOK in a nutshell'](#), I briefly analyzed the guide and showed a few challenges that you could have used their model.

DCAM (v2.2?)

The DCAM model is organized into eight core capabilities. This model was developed primarily for financial institutions, but, in my opinion, the model is actually rather generic and can be used in other industries as well.

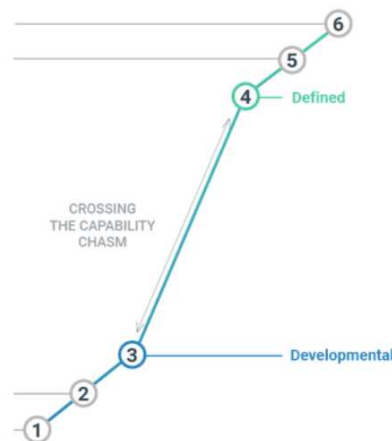
DCAM(Data Management Capability Assessment Model) の拡張要素

Web紹介ページ： <https://edmcouncil.org/frameworks/dcam/>

・クラウドでのデータ環境向けデータマネジメント環境評価としてCDMC(Cloud Data Management Capability)の定義が進められている。主に個人情報や機密データのクラウド環境での管理を確実にしようというための拡張要素としてDCAM評価を補強する位置付けとして捉えられる。機能を提供するベンダー(クラウド環境、各種ツール)、およびそれらを利用する立場にある利用者、双方からの利用を意図して進められている。



CDMCの6つのコンポーネント要素



SCORE	CATEGORY	DESCRIPTION
1	Not Initiated	Ad hoc data management (performed by heroes)
2	Conceptual	Initial planning activities (white board sessions)
3	Developmental	Engagement underway (stakeholders being recruited and initial discussions about roles, responsibilities, standards and processes)
4	Defined	Data management capabilities established and verified by stakeholders (roles and responsibilities structured, policy and standards implemented, glossaries and identifiers established, sustainable funding)
5	Achieved	Data management capabilities adopted and compliance enforced (sanctioned by executive management, activity coordinated, adherence audited, strategic funding)
6	Enhanced	Data management capabilities fully integrated into operations (continuous improvement)

CDMC要素の評価スコアの体系

⇒ こういった環境評価(アセスメント)体系の中で14のキーコントロール(統制)要素が定義され、それらを評価・検討に必要な情報要素が、DCAM情報モデル(オントロジー) V1.1 として整理された(Oct.,2022)

CDMC情報モデル作成の目的

出典： 資料1 p.3

CDMC INFORMATION MODEL OVERVIEW

The CDMC Information Model is an ontology that draws on and combines related open frameworks and standards to describe the information required to support cloud data management. This provides a foundation for the configuration of data catalogs, interoperability of data catalogs and automation of controls across cloud service and technology providers.

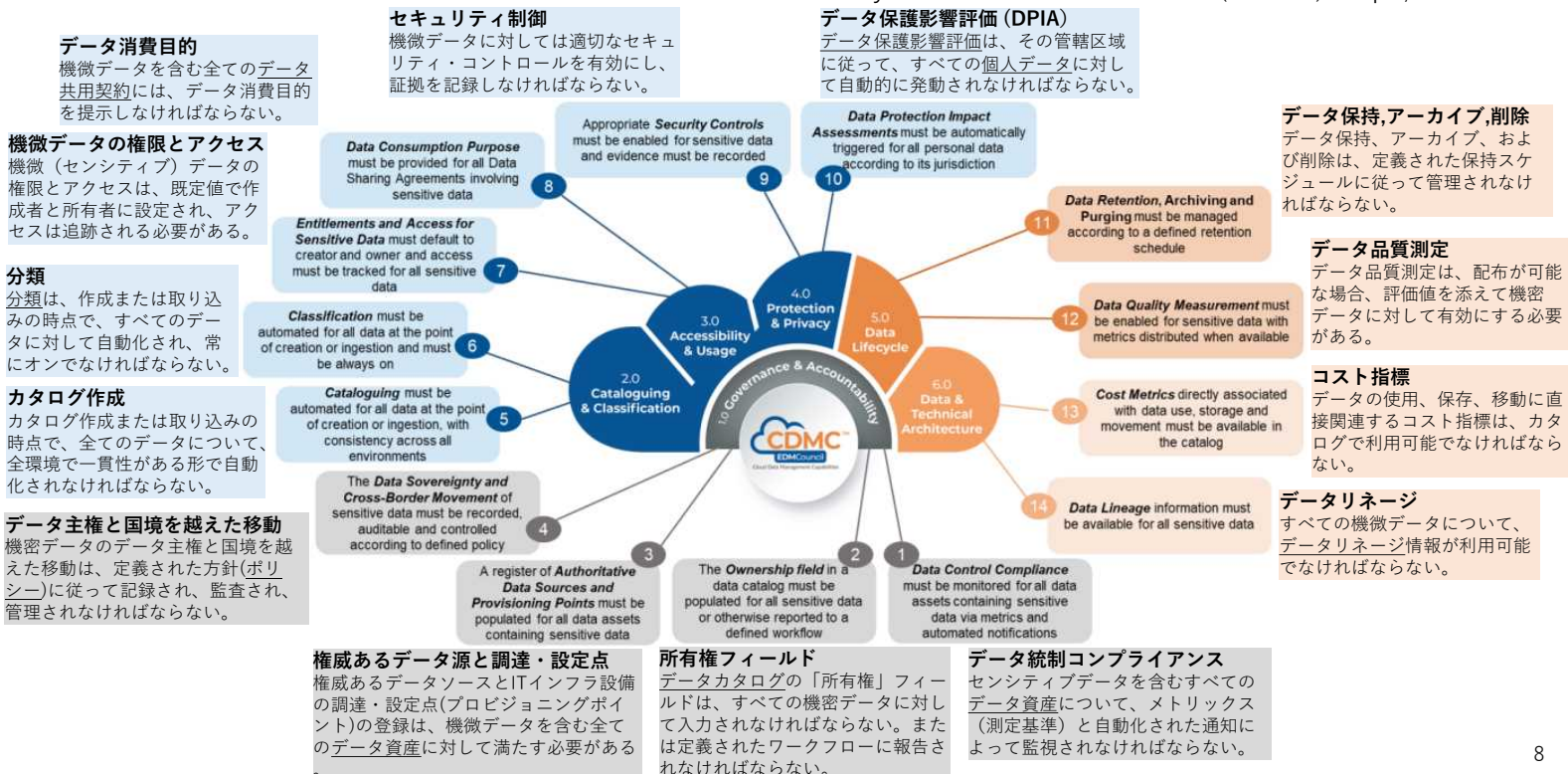
(参考翻訳) ... DEEPL翻訳をベースに一部訂正したもの

CDMC情報モデルは、クラウドデータ管理のサポートに必要な情報を記述するために、関連するオープンフレームワークと標準を利用し、組み合わせたオントロジーである。これは、データ・カタログの構成、データ・カタログの相互運用性、クラウド・サービスおよびテクノロジー・プロバイダ間の制御/統制自動化のための基盤を提供する。

・資料1 CDMC Information Model Controls Tests MappingsV1.1(Oct., 2022)

p. 6 統制要素概要

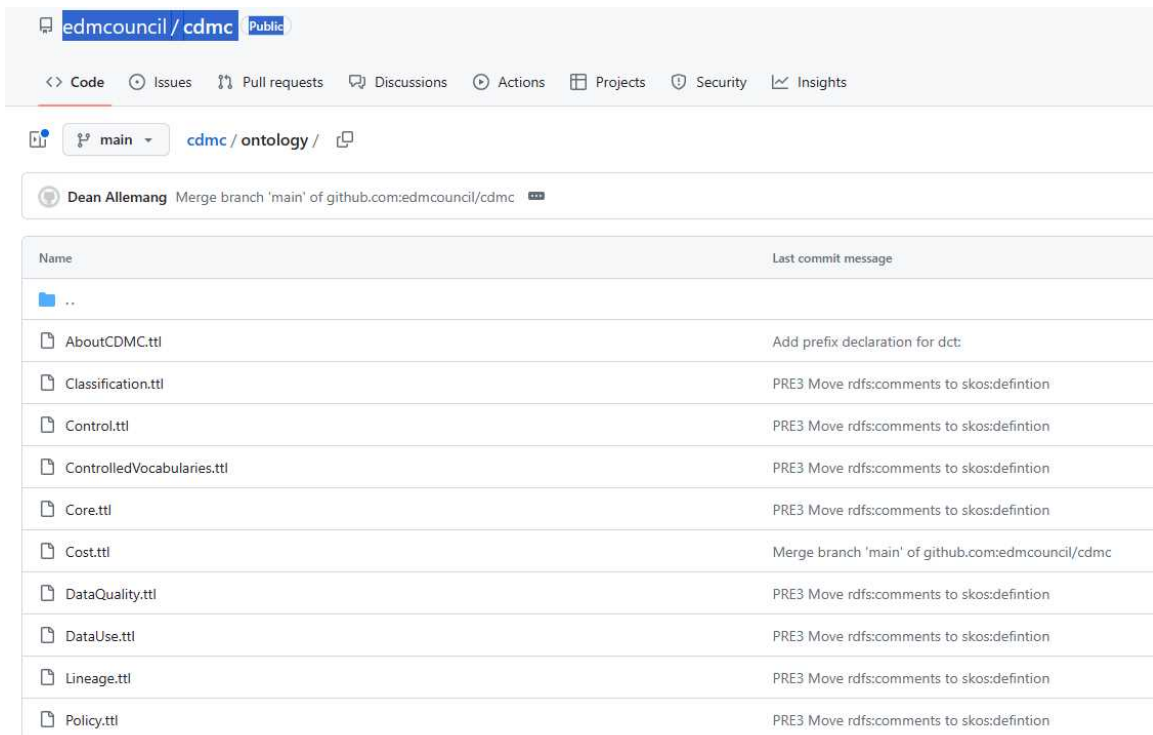
CDMCのキー統制要素は以下の図でまとめられる： 要素項目詳細説明は、以下文書を参照
 ・CDMC Key Controls and Automations (V. 1.1.1) Sept., 2021



CDMC情報モデル(V1.1.1)の入手場所

rdf/rdfs/OWLのオントロジーファイル形式で提供

[cdmc/ontology at main · edmcouncil/cdmc · GitHub](https://github.com/edmcouncil/cdmc)



9

CDMC All info-model V.1.1 (Grafoを利用した視覚化全体図)

【参考】 モジュール(系)内

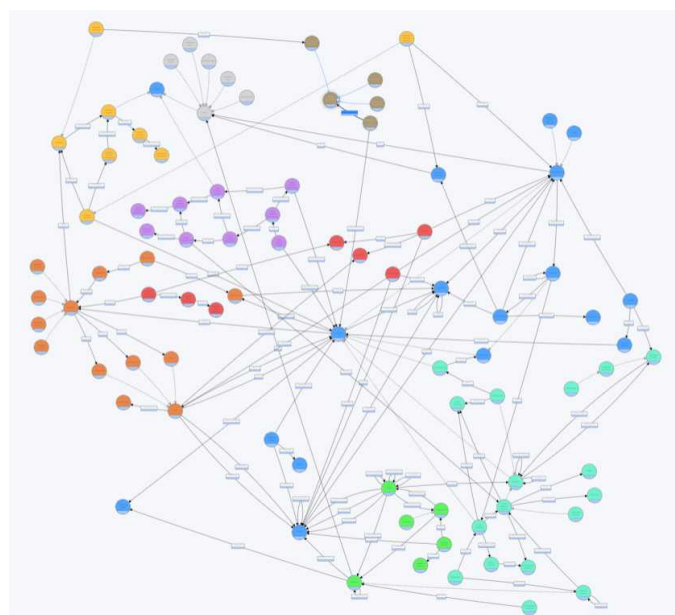
ノード数	構成モジュール(系)名
5	1. Classification系
8	2. Control系
17	3. Controlled Vocaburaries系
17	4. Core系
6	5. Cost系
9	6. Data Quality系
7	7. Data Use系
6	8. Lineage系
13	9. Policy系

計 88 (ノード)

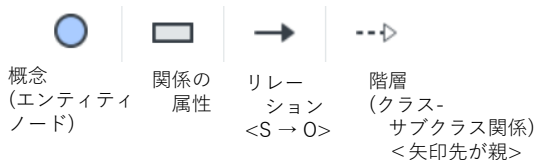
備考：Costに定義エラーらしき内容1個あり

【ポイント】

- ・ 14のKey Control 内容を検証するために必要なデータ領域概念を表す
- ・ データモデル設計レベルに当て嵌めると「概念データモデル」レベル
→ 主要エンティティ(概念)とその関係を表記(データ属性はない)
- ・ 88のクラス(エンティティ)階層とその間のリレーションリンクを表示



【参考】 GRAFOの表示記号



10

【参考】データアーキテクチャ - エンタープライズデータモデル … 考え方の例示

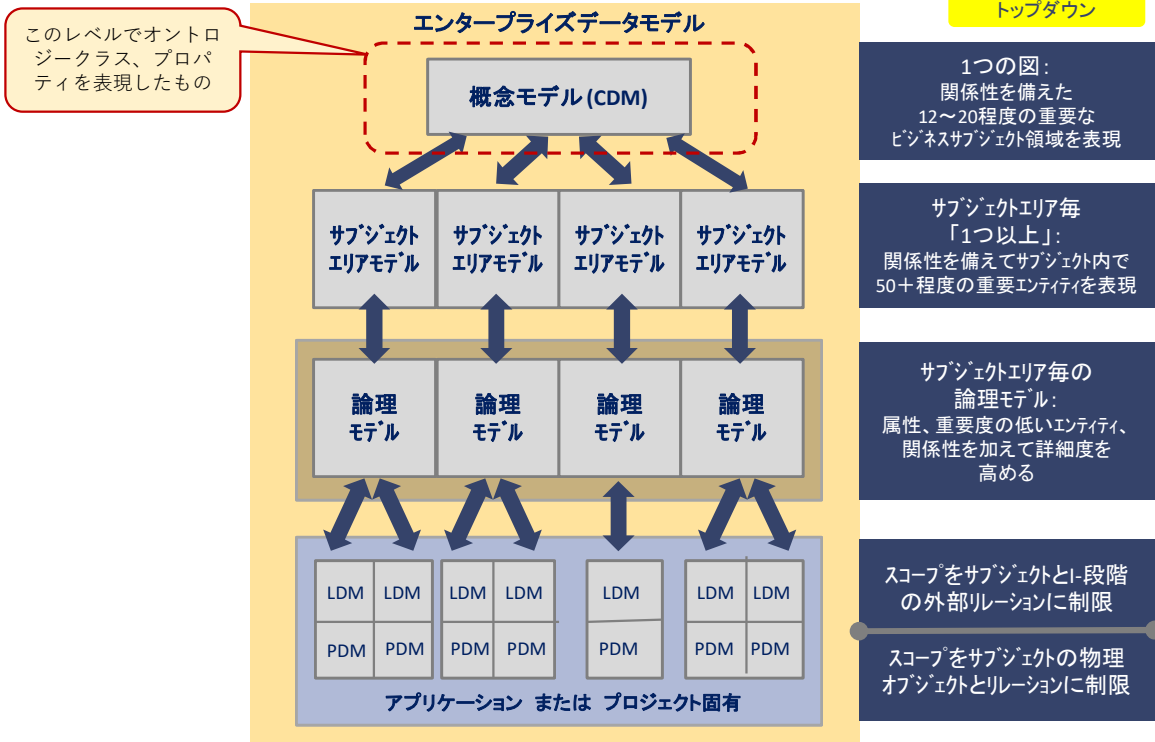


図 エンタープライズデータモデルの全体イメージ例

図出典 DMBOK 2nd p.106図を元に翻訳作成

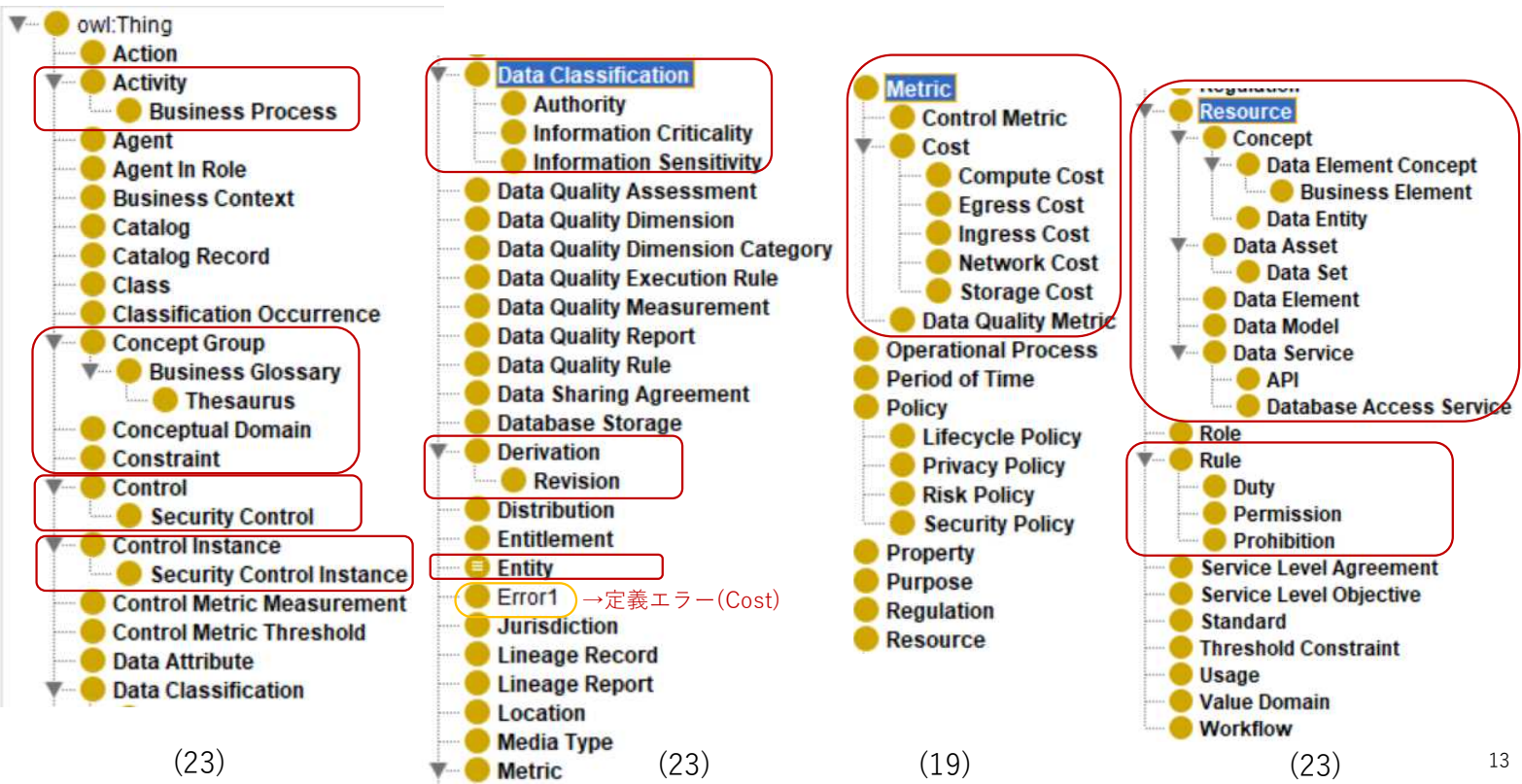
p.5 This document presents the results of the work that has been done to validate the CDMC Information Model by demonstrating its support of the test specifications of the CDMC Key Controls. There is a section for each of the fourteen Key Controls, with a table that presents the following:

(参考翻訳) 同前

本文書は、CDMC情報モデルを検証するために行われた作業の結果を示す。CDMCキーコントロールのテスト仕様をサポートすることを実証する。14のキー・コントロールのセクションがあり、以下のような表がある：

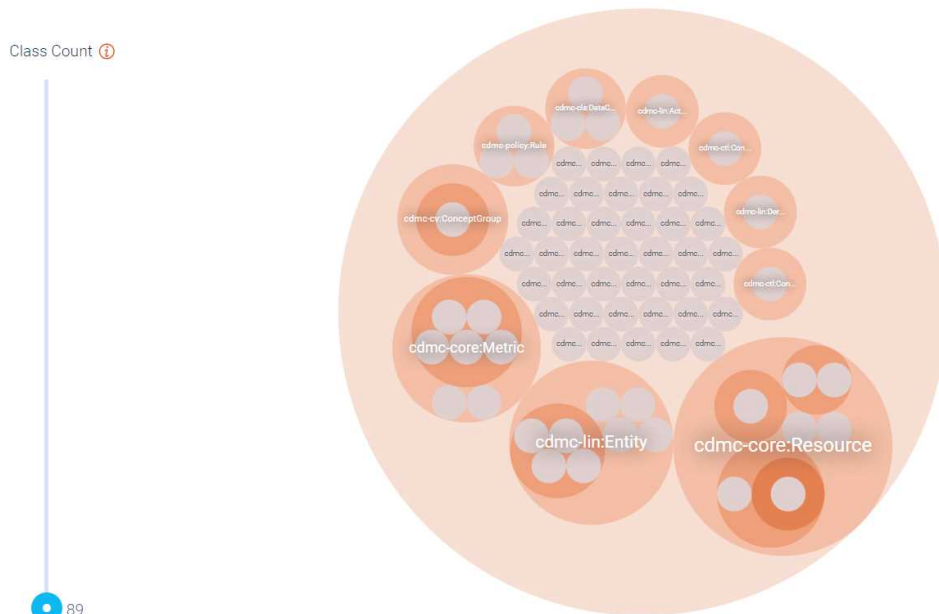
コンポーネント	キー統制(制御)要素を含むCDMCコンポーネント
ケイパビリティ	キー統制要素と整合したCDMCケイパビリティ
統制要素説明	CDMCフレームワーク文書からの、統制要素の説明
統制要素テスト (キー用語を強調)	CDMC制御手続きとテスト仕様文書からの、統制要素のテスト仕様 ・仕様書内の主要用語は、下線とイタリック体で強調表示されている。CDMC情報モデル (次の行) の概念にマッピングされていることを示す。
用語マッピング	テスト仕様の用語とCDMC情報モデルの概念との整合を表形式で表したものを。 ・用語が上位概念の子である概念にマップされる場合、その階層は次のように表示される： [子概念] [親概念] ・マッピングが2つ以上の関連する概念を横断する場合、このパスは次のような形式で示される： [開始概念] / [関連概念] / … / [終着概念]
インスタンスデータ例示	例データのグラフと、そのモデルがテスト仕様書に示されたユースケースをサポートしていることを語る説明文。 - オレンジ色の楕円は概念のインスタンス例を表す - 楕円に関連する青い長方形は概念の名前を示す。ここで 概念が子概念である場合、これらの長方形は子-親階層を示すために積み重ねられる - 矢印のついた線は概念間の関係を表す
インスタンスデータ用語	例で使用されている概念の定義を表形式で表したものを (アルファベット順)。

情報モデルの具体的な利用では、ここに着目

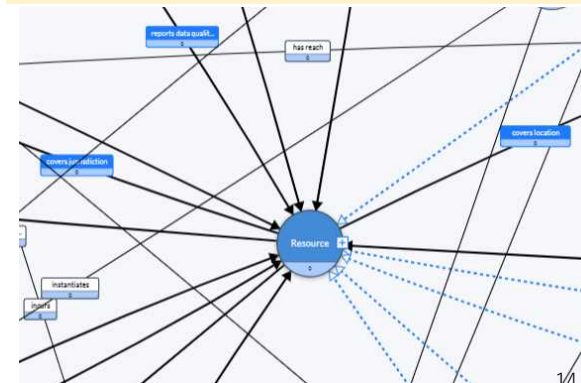


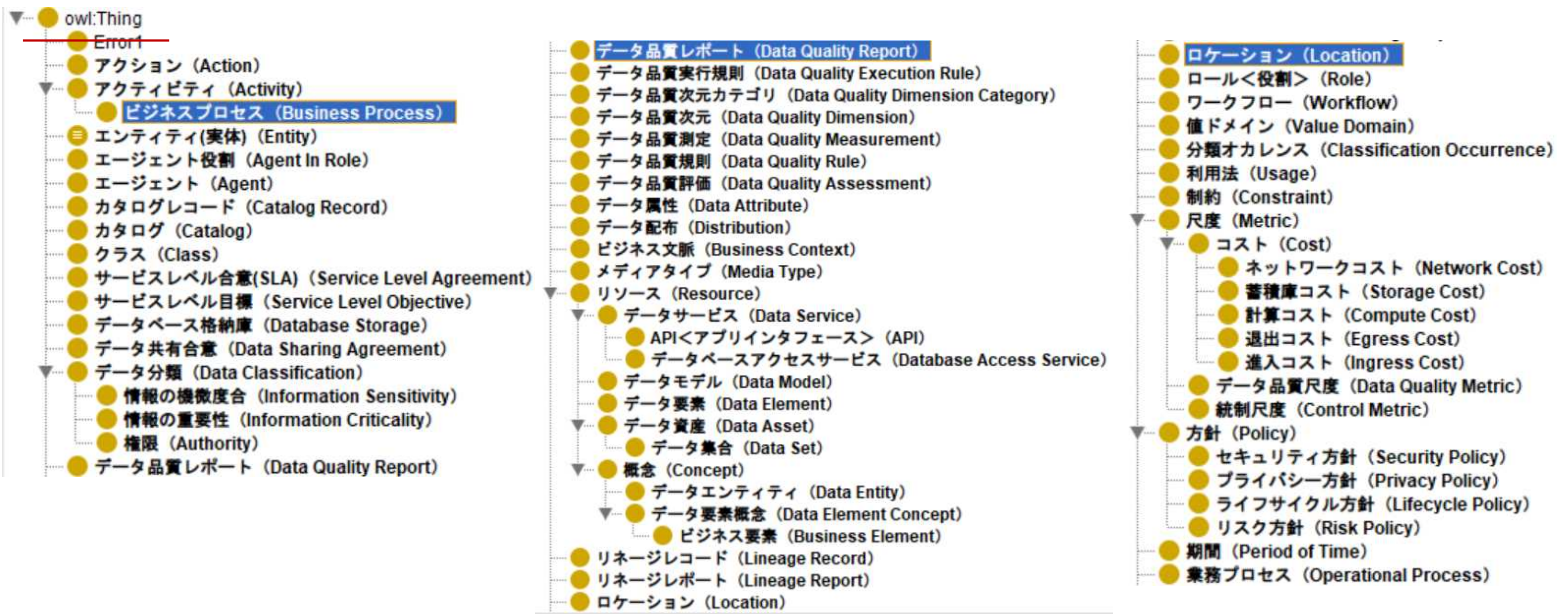
・全オントロジー情報を統合してGraphDBに取り込み、クラス階層図として表示すると以下ようになる。

Class hierarchy



ResourceクラスをGraphで確認(実線が関係線)





【DMBoK2】データモデル・スキーマとモデルレベル考え方の例

ここでの
着目点

●データモデリング・スキーマの利用種類は、実装するDBMSによって適切に使い分けることが期待される。データモデルのレベルと利用スキーマ、DBMSの種類との関係イメージの例を以下に示す。

スキーマ種類	DBMS種類						
	RDBMS	多次元DB (MDBMS)	オブジェクトデータベース	ドキュメントDB	カラムデータベース	グラフデータベース	キーベースデータベース
リレーショナル (Relational)	CDM LDM PDM	CDM LDM	CDM LDM	CDM LDM	CDM LDM	CDM LDM	CDM LDM
ディメンショナル (Dimensional)	CDM LDM PDM	CDM LDM PDM					
オブジェクト指向 (Object-Oriented)	CDM LDM PDM		CDM LDM PDM				
ファクトベース (Fact-Based)	CDM LDM PDM	CDM LDM	CDM LDM	CDM LDM	CDM LDM	CDM LDM	CDM LDM
タイムベース (Time-Based)	PDM						
NoSQL			PDM	PDM	PDM	PDM	PDM

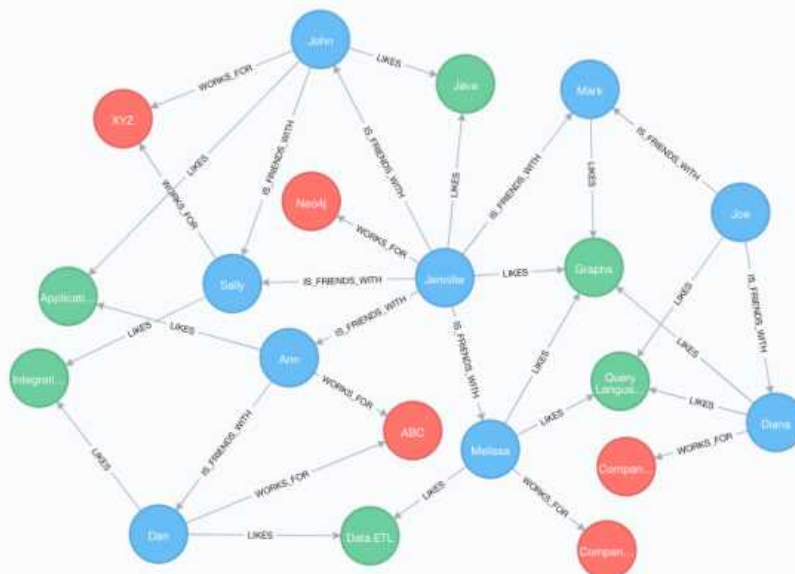
備考: CDM (概念データモデル)、LDM (論理データモデル)、PDM (物理データモデル)

出典: DMBoK 2nd. P.136~137, Table 10を元に作成

©2020.インフォラボ遊悠, All Rights Reserved

グラフデータベースにおける論理データモデルを考える

- グラフDBを用いたモデル表現 (有向、無向) は、インスタンスノードと関係の説明 (エッジ) を用いたものが少ない
- 論理モデル構造を付加する説明例は、余り見つけることができない (物理構造表現が中心)
- こういったネットワークグラフ利用における論理モデル記述を考える
- グラフデータベースにおける論理モデル検討に関しては、以降に説明するように幾つかの重要な考慮ポイントがある



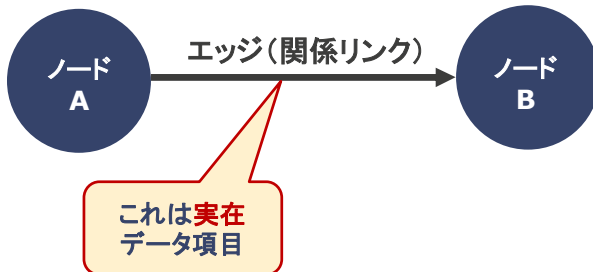
出典: <https://jp.techcrunch.com/2020/02/07/2020-02-04-neo4j-4-0-graph-database-platform-brings-unlimited-scaling/?guccounter=1>

リレーショナル表現(ER図)を用いた論理データモデル表現の試み

【データ構造スコープの相違】 ここでは有向グラフを対象に

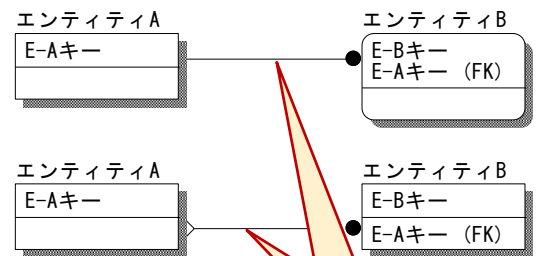
(I) グラフでの基本構造(物理)

- ・インスタンス(ノード)同士の関係性表現
- ・エッジは実在するデータ要素



(II) ER図での基本構造(論理/物理)

- ・エンティティ同士の関係
- ・エンティティキーを用いて関係存在を表記



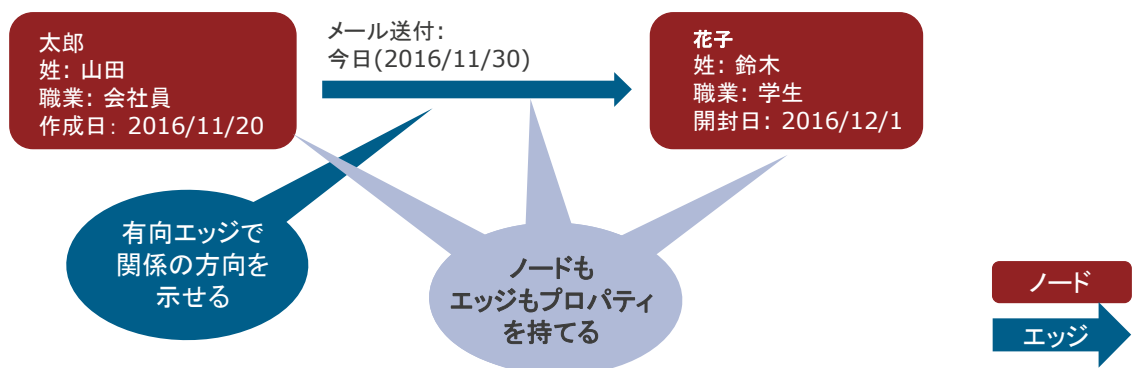
論理モデルでの表現をするためには、

1. ノード群(インスタンス)の抽象化が必要
(ノード群のエンティティ表現を行うということ)
2. エッジのエンティティ表現が必要
3. グラフでのエッジ関係でのグループ化を通じた属性表現が必要

ネットワークグラフの「ノード」と「エッジ」

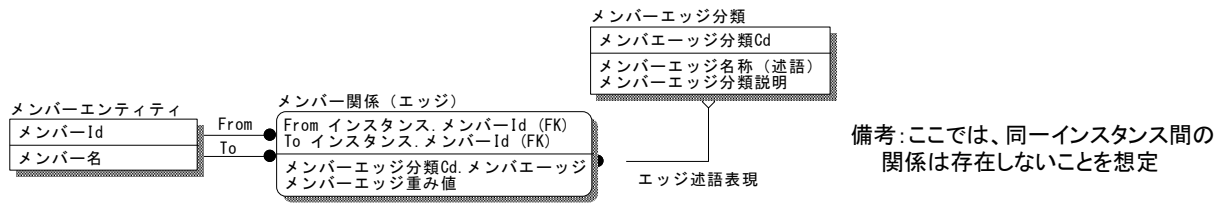
- ネットワークグラフは「ノード」(頂点)と「エッジ」(辺)で関係性の存在を図示表現する
- ノード間の関係を表すエッジには、方向性を持たないもの(無向)と持つもの(有向)がある
- 技術的実装の考え方においては、ネットワークグラフ・データベース仕様は二種類ある
- 「プロパティグラフ・モデル仕様」では、ノードもエッジもプロパティ(属性項目)を持つことができる
- ここでの表現は、インスタンスレベルでのデータ構造の表現図と考えると分かりやすい

例1: プロパティグラフ … モデル管理には、プロパティの属性整理にはある程度正規化的な考え方を持つことが重要
(論理モデル管理連携の必要性)

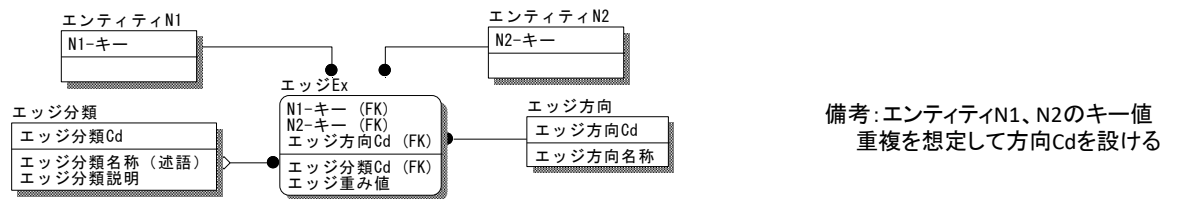


ER図を用いた論理データモデル表現1 (プロパティ系グラフ)

例2: SNS関係グラフ(有向)の単純表現(同一エンティティ内インスタンスの関係)



例3: 異種エンティティ内インスタンス関係グラフ(有向)の単純表現

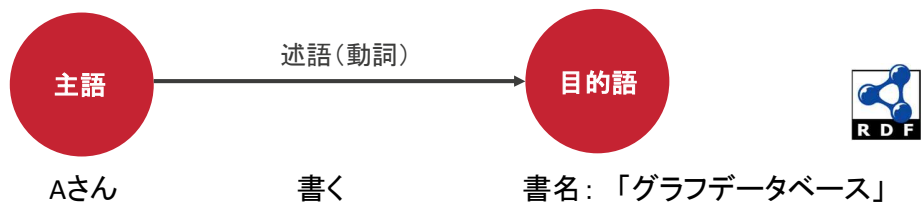


・もうすこし複雑な関係を表現したネットワークグラフの論理モデルについては後で説明する

ネットワーク図を用いた論理データモデル表現 (RDF系グラフ)

- 第二のグラフ表現でのセマンティックWebの領域の話題としてRDF (Resource Description Framework) の考え方がある。
- 「主語 (Subject)」、「述語 (Predicate)」、「目的語 (Object)」の三つ組 (トリプル) でWeb上にあるリソースを記述しようという考え方から始まった。意味を表すメタデータのための利用として当初規定されており、RSS (RDF Site Summary) などでの利用ケースがある。
- 現在は「リンクトデータ」の表現としても提案されており、オープンデータの流れにより、様々な利用が提唱されている。
- この形式のデータ構造枠組みをサポートするグラフデータベースも提供されている

例4: RDFのトリプル(triple)表現



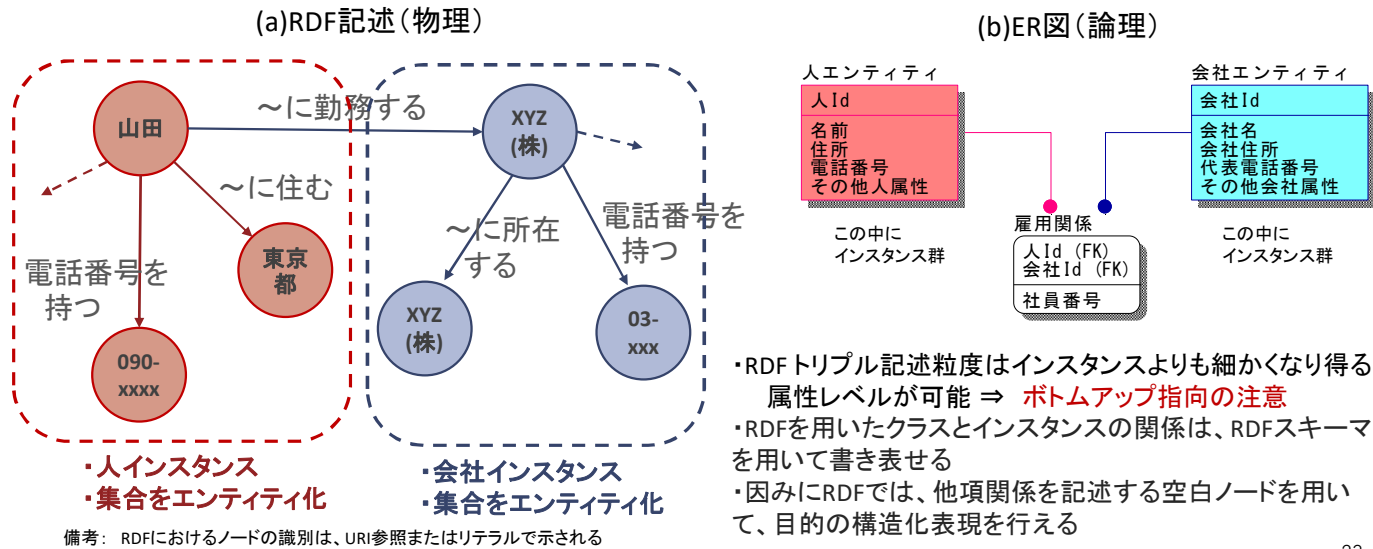
- ✓ このタイプのグラフデータを論理モデル表現するには、特に注意が必要
- ⇒ ノードを単純にインスタンスの表現として捉えられない点があり、モデル表現上の工夫が必要
 - ・オントロジー的発想
 - ・ノードとエッジの「エンティティ・グループ」的対応付けと管理

備考: RDFにおけるノードの識別は、URI参照またはリテラルで示される

ネットワーク図を用いた論理データモデル表現(RDF系グラフ)

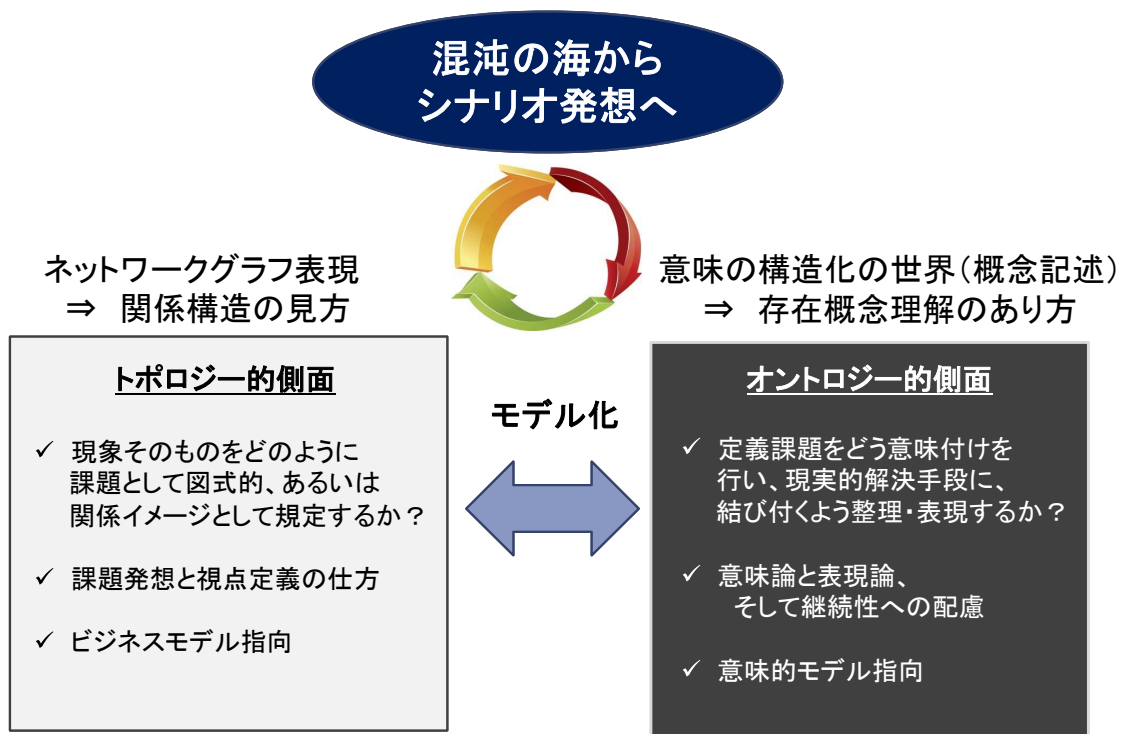
- このタイプのグラフデータをER図発想によるモデル概念表現のための論理データモデル化するには、ノード群およびエッジ群を、意味を捉えた論理エンティティとして扱う対応付けをする発想が必要(論理モデル管理実施を通じて共有)
- 論理モデル(またはオントロジー設計)を構えた上でのRDF/RDFS実装設計が望まれる

例5: RDF利用における「エンティティ」の捉え方



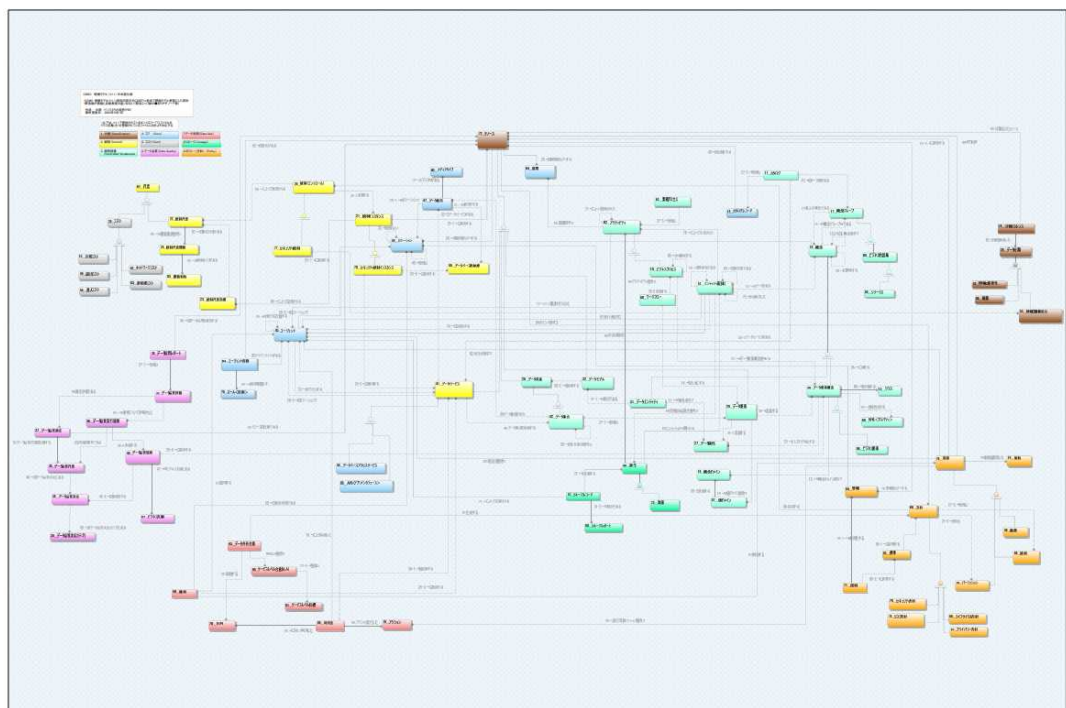
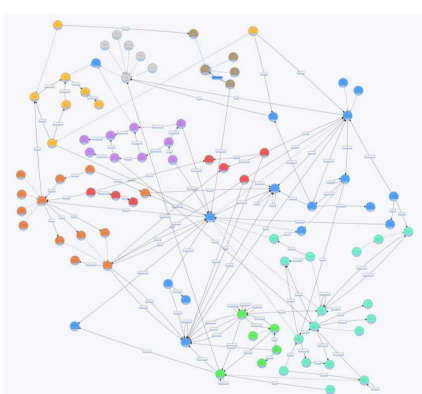
©2020,インフォラボ遊悠, All Rights Reserved

【参考】ネットワークグラフ活用への重要視点 - トポロジーとオントロジー



CDMC V1.1.1 情報モデルをER図形式で表わした例(概念モデルレベル)

CDMC V1.1.1 - CDMC1.0.1.17 (1/1)



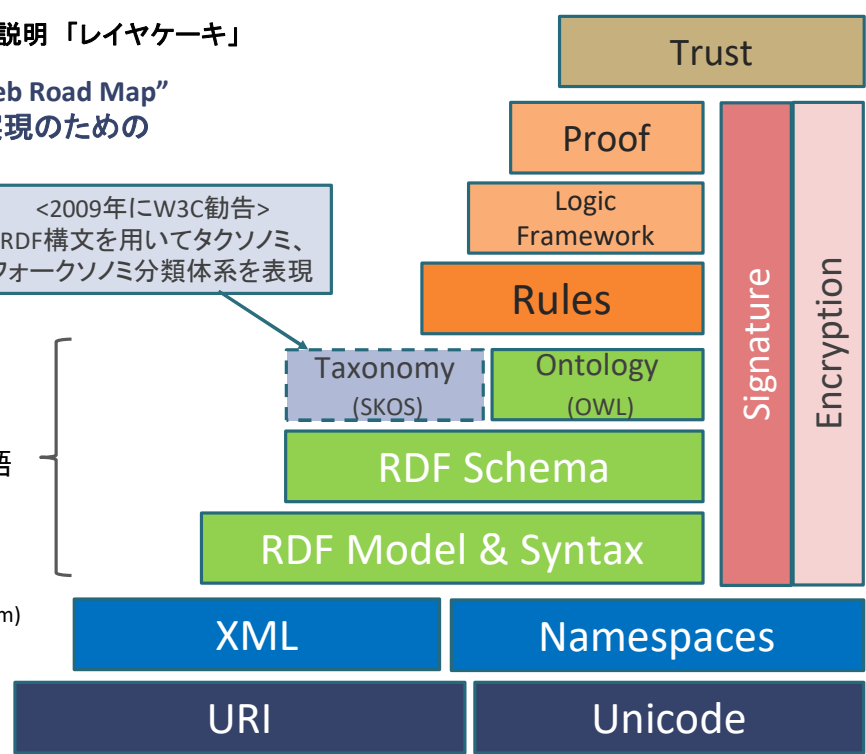
セマンティック・ウェブ実現のためのアーキテクチャ

図 アーキテクチャ説明 「レイヤケーキ」

・バーナーズ・リー“Semantic Web Road Map” (1998年)によるアーキテクチャ実現のための技術の組み合わせ説明図に幾つか説明を付加した。

- ・2004年にW3C 勧告仕様とされた部分
- ・2008年にRDF用検索言語 SPARQLが勧告された

<2009年にW3C勧告>
・RDF構文を用いてタクソノミ、フォークソノミ分類体系を表現



備考:

- SKOS(Simple Knowledge Organization System)
- OWL(Web Ontology Language)

出典:「セマンティック・ウェブのための RDF/OWL 入門」 神崎正英著 (2005年 森北出版) p.6 図1.3 を元に 説明を追記 またSKOSの位置付けを追加した。

(a) RDFの構文要素

- ・ RDF (Resource Description Framework) は、セマンティックWebの基盤となる基本的なフレームワークである。RDFは人があらゆる事柄について表現することを可能とし、それらの表現を1つのモデルに階層化する機能を提供する。RDFは2003年にW3C勧告となった。

語彙	概要
rdf:	・ RDF層の標準的な名前空間としての識別子。RDFの型やプロパティを定義するために使用される。(名前空間「rdf」のグローバルURIは、 http://www.w3.org/1999/02/22-rdf-syntax-ns#)
rdf:type	・ 基本となる型を提供するプロパティ。トリプルにおいて「rdf:type」の主語はどんな識別子でも構わない。目的語は主語の型であると理解される。
rdf:Description	・ RDFでノードを表現する要素。ノードがグローバルな名前を持つときは、rdf:about 属性でURI参照を与える(rdf:about属性がない場合、空白ノードを表す)。
rdf:about	・ ノードのグローバルな識別子を与えるための属性。rdf:Description 要素もしくは型付ノード要素に記述する。
(その他)	・ これら以外にも、幾つかのリソースをひとまとまりのグループとして記述できる「コンテナ」、閉じたグループを表現する「コレクション」、文の直接参照をできるようにする具体化(reification)など、種々の関係を表す語彙が用意されている(ここでは省略)。

27

(b) RDFS(RDFスキーマ言語)の構文要素

- ・ RDFS (RDF Schema) は、オブジェクト言語や他のクラスシステム(つまり、クラス、サブクラス、プロパティ)の世界では良くある共通性と可変性の基本的な概念を記述するための表現能力を提供する。

RDFSは2003年にW3C勧告となった。

【補足】「RDFS-Plus」というOWLのサブセットで、RDFSよりは表現力が高いがOWLほど複雑でない、中間的な拡張要素が定義されている。

- ・ RDFスキーマではクラスによってもものごと(リソース)のグループを表す。RDFSは、このクラスを表現するために幾つかの基本クラスをあらかじめ用意している。

語彙	概要
rdfs:	・ RDFスキーマ言語(RDFS)で使われる識別子を表す。この名前空間の語彙の役割と意味の例を以下に示す。(名前空間「rdfs」のグローバルURIは、 http://www.w3.org/2000/01/rdf-schema#)
rdfs:Resource	・ RDFが表現するあらゆるものは「リソース」クラスに属する。全てのものは、このインスタンスであり、また全てのクラスはrdfs:Resource のサブクラスである。
rdfs:Class	・ RDFのクラス全体を表すクラス。RDFSで表現するクラスは、全てrdfs:Classのインスタンスになる。
rdfs:Literal	・ RDFで記述するリテラル(文字列)が属するクラス。rdfs:Classのインスタンスになる。
rdfs:Datatype	・ データ型を表すクラス。
rdfs:Property	・ RDFのプロパティを表すクラス。プロパティはRDFのトリプルを構成する基本概念であり、rdf: の名前空間に属するものとして表現される。
rdfs:subClassOf	・ 主語となるリソース(クラス)が、目的語となるリソース(クラス)のサブクラスであることを示す。
rdfs:domain	・ プロパティの主語となるリソースのクラス(定義域)を示す。
rdfs:range	・ プロパティの目的語となるリソースのクラス(値域)を示す。
rdfs:subPropertyOf	・ サブプロパティ関係を表現する。推移性を持つプロパティ。
rdfs:label	・ クラス、プロパティの名前を、より人間に分かりやすい形で示すために用いる。
rdfs:comment	・ クラス、プロパティの「意味」を人間が理解できる形で記述するために用いる。
rdfs:seeAlso	・ 主語リソースに関する関連情報のリソースを示すプロパティ。スキーマだけでなく、RDFで記述するリソースの関連情報を示す手段として用いられる。一般的にはWeb上でアクセスできるリソースのURIを記述する。
rdfs:isDefinedBy	・ 主語リソースが定義されているリソースを示すプロパティ。語彙の定義や用例などが解説されているWebページを示すといった使い方が想定される。

28

(c) OWL (Web Ontology Language)の構文要素

・OWLはセマンティックWebにロジック表現の能力を与える。これによりモデル構築者がクラス、オブジェクト、プロパティ間の詳細な制約を表現することが可能になる。OWLは、2003年にW3勧告として承認された。

- owl には、3つのサブ言語が用意されている。
 - ・OWL DL …… OWLの語彙を全て利用しつつ、論理計算の完全性(全ての含意は計算可能)と決定可能性(全ての計算は有限時間内に完了する)を確保するように設計されている。
 - ・OWL Full …… OWLの語彙とRDFの柔軟性を最大限利用可能にするためのもの。OWLの中で最も表現力があるが、計算の完全性、決定可能性は保証されない。
 - ・OWL Lite …… クラス階層の構築と基本的な制約条件の記述が主目的である場合に利用できる。OWL DL/OWL Fullで使える語彙の一部が使えなかったり、使い方が制限されている。

語彙	概要
owl:	・Webオントロジー言語OWLで使われる識別子を表す。この名前空間の語彙の役割と意味の例を以下に示す。(名前空間「owl」のグローバルURIは、 http://www.w3.org/2002/07/owl#)
owl:Class	・ある概念を抽象化して表現する。全ての個体は少なくとも1つの owl:Class に属さなければならない。owl:Classは rdfs:Class のサブクラスである。
owl:Thing	・OWLの組込クラスで、OWLで記述する世界の個体全体に対応する。全ての個体は owl:Thing のインスタンスであり、全てのOWLクラスは owl:Thing のサブクラスになる。
owl:Nothing	・OWLの組込クラスで、インスタンスを一つも持たない「空集合」に対応する。
owl:intersectionOf	・クラスを列挙したクラスの積として表現する。定義域は owl:Class、値域は rdf:List である。
owl:unionOf	・クラスを列挙したクラスの和として表現する。定義域は owl:Class、値域は rdf:List である。
owl:complementOf	・主語クラスを目的語クラスの否定(反転)として表現する。定義域、値域共に owl:Class になる。
owl:Restriction	・プロパティによる制約を記述する。値もしくは出現回数の制約式を必ず一つだけ持つ。
owl:onProperty	・owl:Restriction による制約の対象となるプロパティを示す。定義域は owl:Restriction、値域はrdf:Property
(その他)	・値に関する制約、出現回数に関する制約(カーディナリティ)、同等なクラスと必要十分条件の定義、owlプロパティ、オントロジーヘッダ、オントロジーインポートといった内容を表す様々な語彙がある。

29

(d) 既知の語彙を用いたメタデータ記述

・既に他で定義されている「広く普及している語彙」を参照して、メタデータを作成する手法が取られる。

【定義例】以下表を参照。

名称	概要
ダブリンコアの基本要素 (dc:)	・RDFの記述に利用できる代表的な語彙にダブリンコア(Dublin Core(DC))がある。当初(1995年)に15の基本要素(プロパティ)が定義されている。その後幾つかの追加要素も検討された。Dublin Core Metadata Initiative(DCMI)によって新しい語彙の検討や普及活動が行われている。DC基本要素(DCMES)は、以下の名前空間で示される。一般に「dc:」という接頭辞でマッピングして用いる。 http://purl.org/dc/elements/1.1/ という名前空間URIで示される。
ダブリンコア精密化要素 (dct:)	・基本15要素の意味を限定して、より明確な記述を可能にするために定義された(例えば、日付要素は、単に日付だけでなく、ある時は「公開日」別の場合には「更新日」という異なった意味を与えたいというような場合)。精密化要素は以下で示される。一般に「dct:」という接頭辞にマップして用いる。 http://purl.org/dc/terms/
FOAF(Friend of a friends)	・人に関する情報を記述するための試みとしてRDF拡張を目指す要素。語彙の名前空間は以下で表され、「foaf:」という接頭辞にマッピングして用いる。Agent,Person,Group,Organization といった語彙や人物に関する幅広いリソースの記述も念頭に検討されている。 http://xmlns.com/foaf/0.1/
SKOS(Simple Knowledge Organization System)	・RDFを用いてシソーラス、分類学および統制語などの概念構造体系をセマンティックWeb的手法で記述する。SKOSは、モデル構築者がモジュール化した知識構造を作成できるように設計されている。
その他	・コンテンツ関係の基本的な語彙として、以下のようなものがある。 - RSS(サイトの新着コンテンツなどをメタデータとして記述、配信するために利用) - PRISM(主として雑誌の記事をオンライン配布して再利用するため、記事の内容や権利関係をメタデータとして記述) - CCLP(知的創作物をより広く容易に利用可能とするためのプロジェクト)等。(詳細は各プロジェクトの参照URIで確認)、、、など。

RDF/RDFS/OWLの幾つかの記述形式種類について

- RDFをネットワーク経由でやり取りが可能な文書として記述する構文形式には複数の記法(Notation)がある。

(1) RDF/XML

- RDFのXML構文は、“RDF/XML Syntax Specification”で定義されており、略して「RDF/XML」と呼ばれる。RDFトリプルのグラフをRDF/XMLで表現するには、主語、述語、目的語をそれぞれXMLの要素として記述し、サンドイッチのように入れ子にして組み立てる。
- XML構文では、トリプルの主語、目的語を表すノード要素は `rdf:Description` を用いて記述する。述語は、その名前を修飾名（名前空間接頭辞とローカル名の組合せ）で表したプロパティ要素として記述する。RDFのトリプルは、このノード要素とプロパティ要素を入れ子にしてトリプルを表現する。
- グラフの規模が大きくなると、タグの量が増えて見通しが悪くなる(難点)
<以下で定義されている> RDF 1.1 XML構文 <https://www.w3.org/TR/rdf-syntax-grammar/>



- トリプル「abc123」がWebページ「<https://www.example.com>」を持っているのRDF/XML表現例。
(ここでは例示のための仮想語彙を「ex:」で代替している(本来は、参照語彙のURI表現が必要なところ))
- ```
<rdf:Description rdf:about="urn:pin:abc123">
 <ex:hasHomepage
 <rdf:Description rdf:about="https://www.example.com">
 </rdf:Description>
</ex:hasHomepage>
</rdf:Description>
```

31

## RDF/RDFS/OWLの記述形式種類について(続き)

### (2) Notation 3 RDF (N3) 構文 … Turtle 形式の前身

- RDF の簡潔なシリアライズ表記と修飾名の簡潔かを可能にする。
- 最初に名前空間接頭辞を宣言することで、修飾名を使えるようにしている。修飾名は `<>` で囲まずに直接記述し、URIと区別する。
- 主語、述語、目的語のURIを `<>` で記述し、文の最後にはピリオド(.)を置く。
- N3では、同じ主語を持つプロパティは、主語を省略して';' でまとめて表記できる。
- また、同じプロパティを繰り返して複数の値を記述する場合、プロパティを1回だけ書いて、値を';' でつなぐという簡易表記も可能。
- リソースのタイプ(クラス)を示す `rdf:type` プロパティは、N3では 'a' の1文字だけで省略表記できる。

<以下で定義されている> Notation3 (N3): A readable RDF syntax <https://www.w3.org/TeamSubmission/n3/>

前ページ例を N3 構文記法で書くと以下のようになる。



@prefix ex: <http://example.org/terms/> . → 前ページでは省略したURI指定の名前空間接頭辞宣言  
<urn:pin:abc123> ex:hasHomepage <<https://www.example.com>> .

32

## RDF/RDFS/OWLの記述形式種類について(続き)

### (3) Turtle 構文(TTL) … CDMCの情報モデルオントロジーはこちらを使用

- RDF グラフをコンパクトで自然なテキスト形式で、一般的な使用パターンとデータ型の省略形で完全に記述できるようにしたもの。N3構文を発展させた位置付けとなる。このため記述の基本形式はN3と類似している。
- Turtle は、N-Triples 形式および SPARQL言語 の 3 つのパターン構文との互換性を提供する。
- メディアタイプ: text/turtle
- 一般的なファイル形式: .ttl

<以下で定義されている> RDF 1.1 Turtle: <https://www.w3.org/TR/turtle/> Feb. 2014

### (4) N-Triples 構文

- RDF グラフをエンコードするための行ベースのプレーンテキスト形式。
- メディアタイプ: application/n-triples、text/turtle、または text/plain
- 一般的なファイル拡張子: .nt

<以下で定義されている> RDF 1.1 N-Triples <https://www.w3.org/TR/n-triples/>

### (5) JSON-LD 形式

- RDF を JSON 形式で利用することを目的とした形式
- HTML に埋め込む際は <script type="application/ld+json">...</script> を用いる
- @context はオブジェクトのスキーマを指定する
- @type はオブジェクトの種別(タイプ・クラス)を指定する
- 因みに、Google 推奨形式とされ、利用されているという経緯がある形式

33

## RDF/RDFS/OWLを用いたトポロジー設計: CDMC V1.1.1 情報モデル (例1)

- ファイル Lineage.ttl を参考にオントロジー定義例を検討する(Turtle構文が用いられている) <部分>

```
1 # imports: https://spec.edmcouncil.org/cdmc/ontology/Core
2
3 @prefix cdmc-core: <https://spec.edmcouncil.org/cdmc/ontology/Core/> .
4 @prefix cdmc-lin: <https://spec.edmcouncil.org/cdmc/ontology/Lineage/> .
5 @prefix dc: <http://purl.org/dc/elements/1.1/> .
6 @prefix dct: <http://purl.org/dc/terms/> .
7 @prefix owl: <http://www.w3.org/2002/07/owl#> .
8 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
9 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
10 @prefix skos: <http://www.w3.org/2004/02/skos/core#> .
11 @prefix sm: <http://www.omg.org/techprocess/ab/SpecificationMetadata/> .
12 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
13
14 <https://spec.edmcouncil.org/cdmc/ontology/Lineage>
15 a owl:Ontology ;
16 dct:abstract "This ontology provides concepts and relationships for tracking lineage of data in cloud data management. " ;
17 dct:license "https://opensource.org/licenses/MIT^^xsd:AnyURI ;
18 sm:copyright "Copyright (c) 2022 EDM Council, Inc. " ;
19 owl:imports <https://spec.edmcouncil.org/cdmc/ontology/Core> ;
20 owl:versionInfo "Created with TopBraid Composer" ;
21 .
22
23 cdmc-lin:Activity
24 a owl:Class ;
25 rdfs:subClassOf owl:Thing ;
26 rdfs:label "Activity" ;
27 dc:source <http://www.w3.org/ns/prov#> ;
28 rdfs:isDefinedBy <https://spec.edmcouncil.org/cdmc/ontology/Lineage> ;
29 rdfs:seeAlso <http://www.w3.org/ns/prov#Activity> ;
30 skos:definition "something that occurs over a period of time and acts upon or with entities" ;
31 skos:example "may include consuming, processing, transforming, modifying, relocating, using, or generating entities" ;
32 .
33
```

語彙接頭辞の記述

オントロジーヘッダ定義  
- dct語彙, owlプロパティ等利用  
→ EDMC定義オントロジーCore語彙のインポート

→ N3 rdf:type の省略形を使用 = owl の Class

ThingサブクラスとしてActivityクラス  
定義  
-labelは表示名を定義(UTF-8)

34



## RDF/RDFS/OWLを用いたトポロジー設計: CDMC V1.1.1 情報モデル (例2)

```
33
34 cdmc-lin:Derivation
35 a owl:Class ;
36 rdfs:subClassOf owl:Thing ;
37 rdfs:label "Derivation" ;
38 dc:source <http://www.w3.org/ns/prov#> ;
39 rdfs:isDefinedBy <https://spec.edmcouncil.org/cdmc/ontology/Lineage> ;
40 rdfs:seeAlso <http://www.w3.org/ns/prov#Derivation> ;
41 skos:definition "derivation is a transformation of an entity into another, an update of an entity resulting in a new one, or the creation of a new entity";
42 .
43
44 cdmc-lin:Entity
45 a owl:Class ;
46 rdfs:subClassOf owl:Thing ;
47 rdfs:label "Entity" ;
48 dc:source <http://www.w3.org/ns/prov#> ;
49 rdfs:isDefinedBy <https://spec.edmcouncil.org/cdmc/ontology/Lineage> ;
50 rdfs:seeAlso <http://www.w3.org/ns/prov#Entity> ;
51 owl:equivalentClass _:blank3 ;
52 skos:definition "entity is a physical, digital, conceptual, or other kind of thing with some fixed aspects; entities may be real or imaginary" ;
53 .
54
```

ThingサブクラスとしてDerivation  
クラス定義  
-labelは表示名を定義(UTF-8)

ThingサブクラスとしてEntity  
クラス定義  
-labelは表示名を定義(UTF-8)

35

## RDF/RDFS/OWLを用いたトポロジー設計: CDMC V1.1.1 情報モデル (例3)

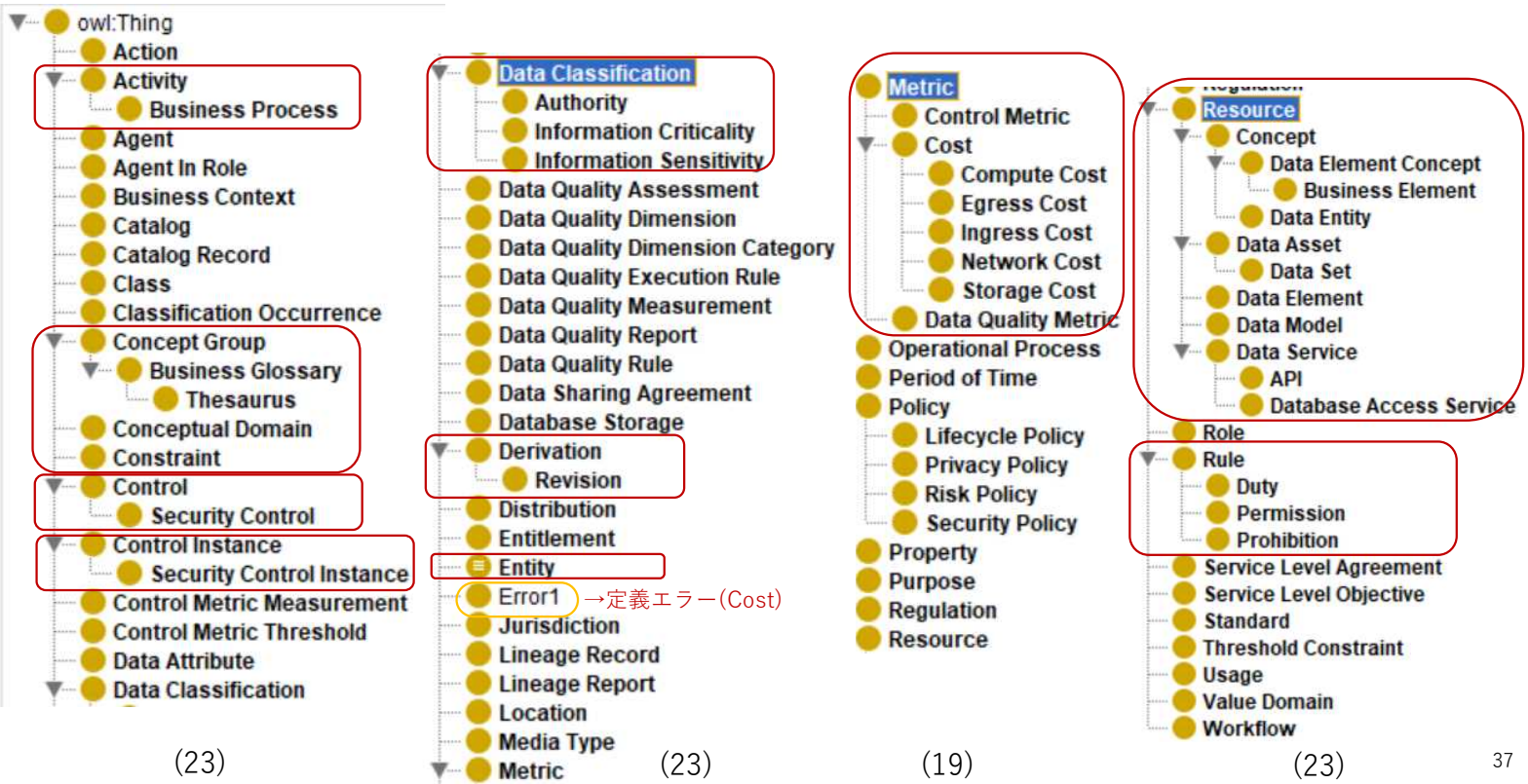
```
83
84 cdmc-lin:derivedFromEntity
85 a owl:ObjectProperty ;
86 rdfs:label "derived from entity" ;
87 dc:source
88 "CDMC" ,
89 "prov:"
90 ;
91 rdfs:domain cdmc-lin:Derivation ;
92 rdfs:isDefinedBy <https://spec.edmcouncil.org/cdmc/ontology/Lineage> ;
93 rdfs:range cdmc-lin:Entity ;
94 rdfs:seeAlso <http://www.w3.org/ns/prov#entity> ;
95 skos:definition "identifies an Entity that is an input to a Derivation" ;
96 .
97
98 cdmc-lin:hadActivity
99 a owl:ObjectProperty ;
100 rdfs:label "had activity" ;
101 dc:source <http://www.w3.org/ns/prov#> ;
102 rdfs:domain cdmc-lin:Derivation ;
103 rdfs:isDefinedBy <https://spec.edmcouncil.org/cdmc/ontology/Lineage> ;
104 rdfs:range cdmc-lin:Activity ;
105 rdfs:seeAlso <http://www.w3.org/ns/prov#hadActivity> ;
106 skos:definition "identifies an Activity that is involved in a Derivation" ;
107 .
108
109 cdmc-lin:hadDuration
110 a owl:ObjectProperty ;
111 rdfs:label "had duration" ;
112 dc:source "CDMC" ;
113 rdfs:domain cdmc-lin:Activity ;
114 rdfs:isDefinedBy <https://spec.edmcouncil.org/cdmc/ontology/Lineage> ;
115 rdfs:range cdmc-core:PeriodOfTime ;
116 skos:definition "specifies the Period of Time over which an Activity took place" ;
117 .
```

プロパティ定義(述語): derivedFromEntity  
-定義域(domain)は、クラスDerivation  
-値域(range)は、クラスEntity  
-labelはプロパティの表示名を定義(UTF-8)

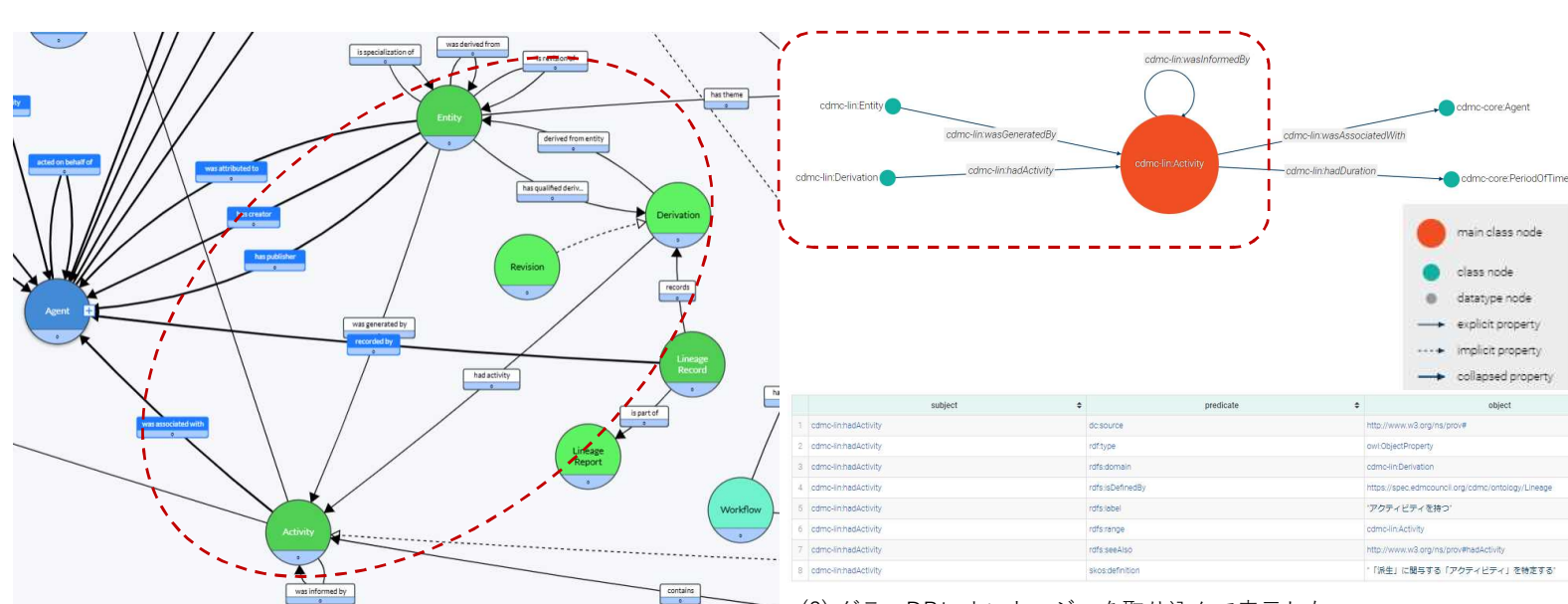
プロパティ定義(述語): hadActivity  
-定義域(domain)は、クラスDerivation  
-値域(range)は、クラスActivity  
-labelはプロパティの表示名を定義(UTF-8)

プロパティ定義(述語): hadDuration  
-定義域(domain)は、クラスActivity  
-値域(range)は、クラスPeriodOfTime  
-labelはプロパティの表示名を定義(UTF-8)

36



RDF/RDFS/OWLを用いたトポロジー設計: CDMC V1.1.1 情報モデル  
 ... Grafo 視覚化ツール、及びグラフDBでの出力例



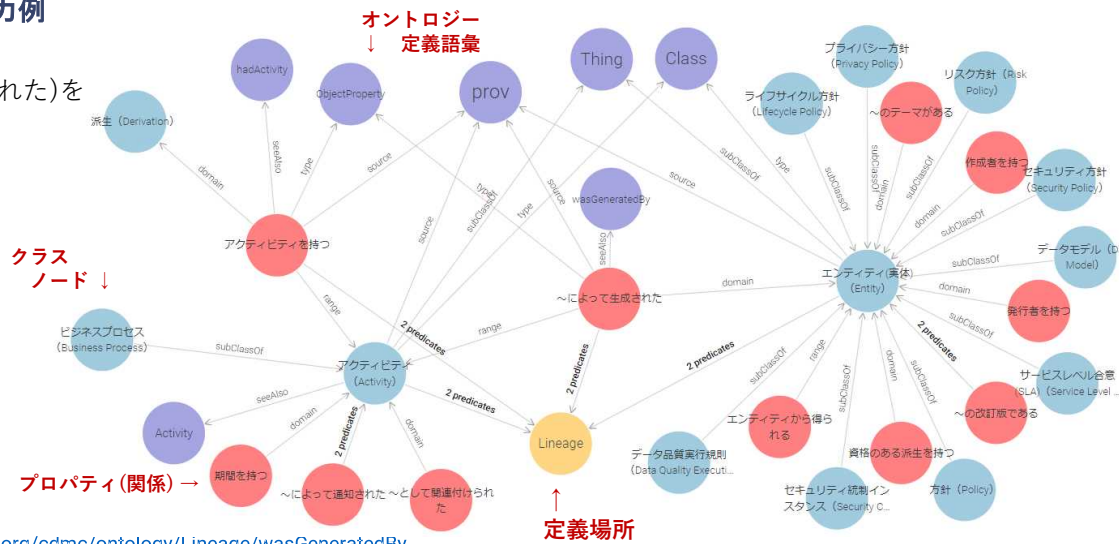
(1) Activity、Derivation、Entityクラス周辺の関係図 (Grafoツール表示)

(2) グラフDBにオントロジーを取り込んで表示した、Activity、周辺のドメイン(定義域)–レンジ(値域)グラフの表示図(上) および、トリプル表示(下:一部日本語化は、中岡による)

# RDF/RDFS/OWLを用いたトポロジー設計: CDMC V1.1.1 情報モデル

## ...GraphDBでのVisual Graph 出力例

・プロパティ  
wasGeneratedBy(~によって生成された)を軸に、グラフDB内に取り込んだトリプル群の視覚グラフ表示

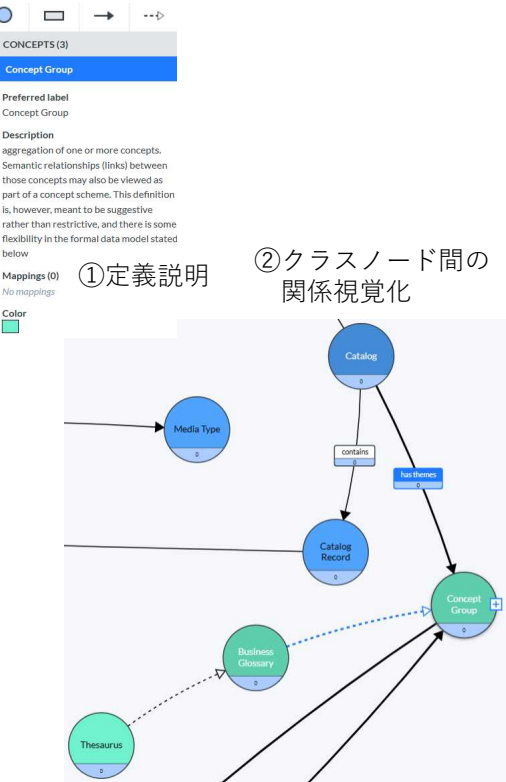


Subject Source: <https://spec.edmcouncil.org/cdmc/ontology/Lineage/wasGeneratedBy>

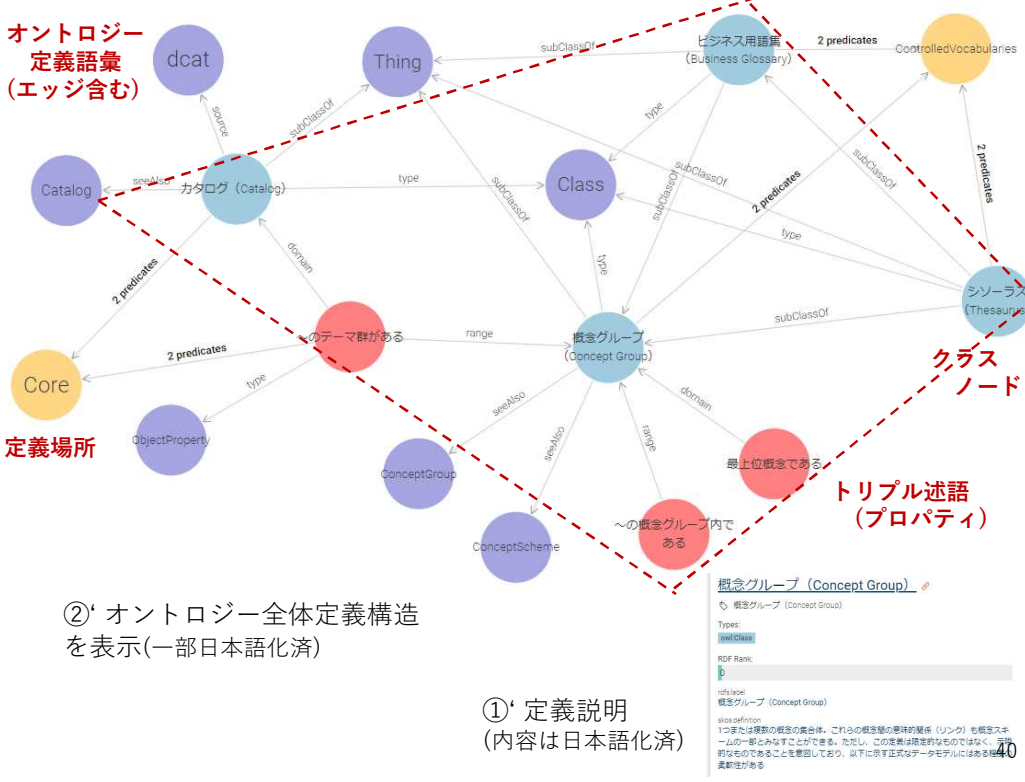
|   | subject                 | predicate        | object                                            | context                          |
|---|-------------------------|------------------|---------------------------------------------------|----------------------------------|
| 1 | cdmc:lin:wasGeneratedBy | dc:source        | http://www.w3.org/ns/prov#                        | http://www.ontotext.com/explicit |
| 2 | cdmc:lin:wasGeneratedBy | rdf:type         | owl:ObjectProperty                                | http://www.ontotext.com/explicit |
| 3 | cdmc:lin:wasGeneratedBy | rdfs:domain      | cdmc:lin:Entity                                   | http://www.ontotext.com/explicit |
| 4 | cdmc:lin:wasGeneratedBy | rdfs:isDefinedBy | https://spec.edmcouncil.org/cdmc/ontology/Lineage | http://www.ontotext.com/explicit |
| 5 | cdmc:lin:wasGeneratedBy | rdfs:label       | "~によって生成された"                                      | http://www.ontotext.com/explicit |
| 6 | cdmc:lin:wasGeneratedBy | rdfs:range       | cdmc:lin:Activity                                 | http://www.ontotext.com/explicit |
| 7 | cdmc:lin:wasGeneratedBy | rdfs:seeAlso     | http://www.w3.org/ns/prov#wasGeneratedBy          | http://www.ontotext.com/explicit |
| 8 | cdmc:lin:wasGeneratedBy | skos:definition  | "「アクティビティ」の構築として「エンティティ」が作成されたことを示す"              | http://www.ontotext.com/explicit |

# Grafo出力とグラフデータベース(オントロジー取込み)間の視覚表示の比較 ... 概念グループに焦点を充てて

(a)Grafo表示



(b)グラフデータベース表示



②' オントロジー全体定義構造を表示(一部日本語化済)

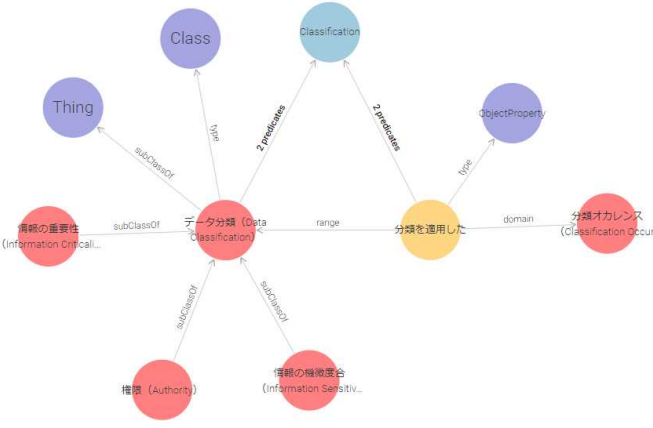
①' 定義説明(内容は日本語化済)



# データ分類 (Data Classification)

Source: <https://spec.edmcouncil.org/cdmc/ontology/Classification/DataClassification>

| subject | predicate                                   | object                           | context                                                                                                                         |
|---------|---------------------------------------------|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| 1       | <a href="#">cdmc-cls:DataClassification</a> | <a href="#">dc:source</a>        | "CDMC"                                                                                                                          |
| 2       | <a href="#">cdmc-cls:DataClassification</a> | <a href="#">dc:source</a>        | "DCAM"                                                                                                                          |
| 3       | <a href="#">cdmc-cls:DataClassification</a> | <a href="#">rdf:type</a>         | <a href="#">owl:Class</a>                                                                                                       |
| 4       | <a href="#">cdmc-cls:DataClassification</a> | <a href="#">rdfs:isDefinedBy</a> | <a href="https://spec.edmcouncil.org/cdmc/ontology/Classification">https://spec.edmcouncil.org/cdmc/ontology/Classification</a> |
| 5       | <a href="#">cdmc-cls:DataClassification</a> | <a href="#">rdfs:label</a>       | "データ分類 (Data Classification) "                                                                                                  |
| 6       | <a href="#">cdmc-cls:DataClassification</a> | <a href="#">rdfs:subClassOf</a>  | <a href="#">owl:Thing</a>                                                                                                       |
| 7       | <a href="#">cdmc-cls:DataClassification</a> | <a href="#">skos:definition</a>  | "データを最も効果的かつ効率的に利用できるように整理するカテゴリー"                                                                                              |



## SPARQL 問合せ言語について

- SPARQLは、RDFで格納されたデータの問合せ言語であり、W3CのRDF Data Access Working Group (DAWG) によって標準化され、2013年3月に SPARQL 1.1 が勧告された。
- SPARQLの問合せの基本的な構成要素はトリプルパターンである。ここで使用するトリプルパターンはトリプル定義と同じように見えるが、リソースの中の 主語(S)/述語(P)/目的語(O)のどの位置にでも変数を置けるという点異なる。多くの点で、RDF問合せエンジンはリレーショナルデータベースの問合せエンジンに似ている。
- 変数は「?」という特別文字を記号の前に記述して指定する。RDF問合せは、元のRDFグラフのサブ集合によって新しいグラフを記述することができる。SPARQLのSELECT形式は、グラフを表に変換したものと考えることができる。グラフの一部がマッチしたグラフパターンは、変数に条件付けられた結果を踏まえて表の値として返される。
- このトリプルパターンの構文は、意図的にN3構文(主語/述語/目的語と続き、ピリオドによって終了)と類似させている。これらの各パターンは、自然な質問文として読み替えることができる。

SPARQL利用例等については、DBpedia サイト等を参照。

<https://ja.dbpedia.org/>



DBpediaはWikipediaから情報を抽出してLOD (Linked Open Data)として公開するコミュニティプロジェクトです。 [本家のDBpedia](#)は主に [Wikipedia英語版](#)を対象としています。DBpedia Japaneseの目的は、 [Wikipedia日本語版](#)を対象としたDBpediaを提供することです。



## SPARQL Query & Update; CDMC RDF データを取り込んだグラフに対して検索を実行した例

```
SELECT * {
 ?s ?p ?o
} LIMIT 100
```

|    |                                                                                                                   |                          |                                    |
|----|-------------------------------------------------------------------------------------------------------------------|--------------------------|------------------------------------|
| 39 | <a href="#">owl:imports</a>                                                                                       | <a href="#">rdf:type</a> | <a href="#">rdf:Property</a>       |
| 40 | <a href="https://spec.edmcouncil.org/cdmc/ontology/Core">https://spec.edmcouncil.org/cdmc/ontology/Core</a>       | <a href="#">rdf:type</a> | <a href="#">owl:Ontology</a>       |
| 41 | <a href="#">owl:versionInfo</a>                                                                                   | <a href="#">rdf:type</a> | <a href="#">rdf:Property</a>       |
| 42 | <a href="#">cdmc-cla:Authority</a>                                                                                | <a href="#">rdf:type</a> | <a href="#">owl:Class</a>          |
| 43 | <a href="#">cdmc-cla:DataClassification</a>                                                                       | <a href="#">rdf:type</a> | <a href="#">owl:Class</a>          |
| 44 | <a href="#">skos:definition</a>                                                                                   | <a href="#">rdf:type</a> | <a href="#">rdf:Property</a>       |
| 45 | <a href="#">cdmc-cla:ClassificationOccurrence</a>                                                                 | <a href="#">rdf:type</a> | <a href="#">owl:Class</a>          |
| 46 | <a href="#">dc:source</a>                                                                                         | <a href="#">rdf:type</a> | <a href="#">rdf:Property</a>       |
| 47 | <a href="#">cdmc-cla:InformationCriticality</a>                                                                   | <a href="#">rdf:type</a> | <a href="#">owl:Class</a>          |
| 48 | <a href="#">cdmc-cla:InformationSensitivity</a>                                                                   | <a href="#">rdf:type</a> | <a href="#">owl:Class</a>          |
| 49 | <a href="#">cdmc-cla:appliedClassification</a>                                                                    | <a href="#">rdf:type</a> | <a href="#">owl:ObjectProperty</a> |
| 50 | <a href="#">cdmc-cla:classifiedResource</a>                                                                       | <a href="#">rdf:type</a> | <a href="#">owl:ObjectProperty</a> |
| 51 | <a href="#">cdmc-core:Resource</a>                                                                                | <a href="#">rdf:type</a> | <a href="#">owl:Class</a>          |
| 52 | <a href="https://spec.edmcouncil.org/cdmc/ontology/Control">https://spec.edmcouncil.org/cdmc/ontology/Control</a> | <a href="#">rdf:type</a> | <a href="#">owl:Ontology</a>       |
| 53 | <a href="#">cdmc-ctl:Control</a>                                                                                  | <a href="#">rdf:type</a> | <a href="#">owl:Class</a>          |
| 54 | <a href="#">cdmc-ctl:ControlInstance</a>                                                                          | <a href="#">rdf:type</a> | <a href="#">owl:Class</a>          |
| 55 | <a href="#">_:genid-0c451ad4047c468b9216b6882a58201734703-blank4</a>                                              | <a href="#">rdf:type</a> | <a href="#">owl:Restriction</a>    |
| 56 | <a href="#">_:genid-0c451ad4047c468b9216b6882a58201734703-blank5</a>                                              | <a href="#">rdf:type</a> | <a href="#">owl:Restriction</a>    |
| 57 | <a href="#">cdmc-ctl:ControlMetric</a>                                                                            | <a href="#">rdf:type</a> | <a href="#">owl:Class</a>          |
| 58 | <a href="#">cdmc-core:Metric</a>                                                                                  | <a href="#">rdf:type</a> | <a href="#">owl:Class</a>          |
| 59 | <a href="#">cdmc-ctl:ControlMetricMeasurement</a>                                                                 | <a href="#">rdf:type</a> | <a href="#">owl:Class</a>          |
| 60 | <a href="#">cdmc-ctl:ControlMetricThreshold</a>                                                                   | <a href="#">rdf:type</a> | <a href="#">owl:Class</a>          |
| 61 | <a href="#">cdmc-ctl:SecurityControl</a>                                                                          | <a href="#">rdf:type</a> | <a href="#">owl:Class</a>          |
| 62 | <a href="#">_:genid-0c451ad4047c468b9216b6882a58201734703-blank1</a>                                              | <a href="#">rdf:type</a> | <a href="#">owl:Restriction</a>    |
| 63 | <a href="#">cdmc-ctl:SecurityControlInstance</a>                                                                  | <a href="#">rdf:type</a> | <a href="#">owl:Class</a>          |
| 64 | <a href="#">_:genid-0c451ad4047c468b9216b6882a58201734703-blank2</a>                                              | <a href="#">rdf:type</a> | <a href="#">owl:Restriction</a>    |
| 65 | <a href="#">_:genid-0c451ad4047c468b9216b6882a58201734703-blank3</a>                                              | <a href="#">rdf:type</a> | <a href="#">owl:Class</a>          |
| 66 | <a href="#">_:genid-0c451ad4047c468b9216b6882a58201734703-blank3</a>                                              | <a href="#">rdf:type</a> | <a href="#">owl:Restriction</a>    |
| 67 | <a href="#">cdmc-ctl:ThresholdConstraint</a>                                                                      | <a href="#">rdf:type</a> | <a href="#">owl:Class</a>          |

43

### ①RDFストア内(エンドポイント)のトリプルの件数確認

```
SELECT (COUNT(*) AS ?count)
{
 ?s ?p ?o .
}
```

| Filter query results |                    | count |
|----------------------|--------------------|-------|
| 1                    | *1719**xsd:integer |       |

### ②RDFストア内プロパティの一覧を取得

```
SELECT DISTINCT ?p
{
 ?s ?p ?o .
} LIMIT 100
```

|    | p                                                                                                                           |
|----|-----------------------------------------------------------------------------------------------------------------------------|
| 1  | <a href="#">rdf:type</a>                                                                                                    |
| 2  | <a href="#">rdfs:subPropertyOf</a>                                                                                          |
| 3  | <a href="#">rdfs:subClassOf</a>                                                                                             |
| 4  | <a href="#">rdfs:domain</a>                                                                                                 |
| 5  | <a href="#">rdfs:range</a>                                                                                                  |
| 6  | <a href="#">owl:equivalentClass</a>                                                                                         |
| 7  | <a href="http://proton.semanticweb.org/protonyas#transitiveDist">http://proton.semanticweb.org/protonyas#transitiveDist</a> |
| 8  | <a href="#">owl:inverseOf</a>                                                                                               |
| 9  | <a href="#">rdf:has</a>                                                                                                     |
| 10 | <a href="#">rdf:has</a>                                                                                                     |
| 11 | <a href="#">rdfs:label</a>                                                                                                  |
| 12 | <a href="#">rdfs:definedBy</a>                                                                                              |
| 13 | <a href="#">rdfs:seeAlso</a>                                                                                                |
| 14 | <a href="#">dcterms:abstract</a>                                                                                            |
| 15 | <a href="#">dcterms:license</a>                                                                                             |
| 16 | <a href="#">ami:copyright</a>                                                                                               |
| 17 | <a href="#">owl:imports</a>                                                                                                 |
| 18 | <a href="#">owl:versionInfo</a>                                                                                             |
| 19 | <a href="#">skos:definition</a>                                                                                             |
| 20 | <a href="#">dc:source</a>                                                                                                   |
| 21 | <a href="#">owl:isProperty</a>                                                                                              |
| 22 | <a href="#">owl:isValueFrom</a>                                                                                             |
| 23 | <a href="#">cdmc-core:Relationship</a>                                                                                      |
| 24 | <a href="https://spec.edmcouncil.org/11170/0/Property">https://spec.edmcouncil.org/11170/0/Property</a>                     |
| 25 | <a href="#">owl:unionOf</a>                                                                                                 |
| 26 | <a href="#">owl:isValueFrom</a>                                                                                             |
| 27 | <a href="#">skos:example</a>                                                                                                |
| 28 | <a href="#">skos:scopeNote</a>                                                                                              |

44

③RDFストア内プロパティの一覧を頻度順に取得

```
SELECT ?p (count(?p) AS ?count)
{
?s ?p ?o .
}
GROUP BY ?p
ORDER BY DESC(?count)
LIMIT 100
```

|    | p                                                      |                    |
|----|--------------------------------------------------------|--------------------|
| 1  | rdfs:seeAlso                                           | "253"^^xsd:integer |
| 2  | rdf:type                                               | "252"^^xsd:integer |
| 3  | rdfs:subClassOf                                        | "198"^^xsd:integer |
| 4  | rdfs:label                                             | "183"^^xsd:integer |
| 5  | rdfs:isDefinedBy                                       | "177"^^xsd:integer |
| 6  | skos:definition                                        | "177"^^xsd:integer |
| 7  | dc:source                                              | "134"^^xsd:integer |
| 8  | rdfs:domain                                            | "87"^^xsd:integer  |
| 9  | rdfs:range                                             | "87"^^xsd:integer  |
| 10 | rdfs:subPropertyOf                                     | "31"^^xsd:integer  |
| 11 | owl:someValuesFrom                                     | "29"^^xsd:integer  |
| 12 | owl:onProperty                                         | "24"^^xsd:integer  |
| 13 | owl:imports                                            | "14"^^xsd:integer  |
| 14 | dcterms:abstract                                       | "9"^^xsd:integer   |
| 15 | dcterms:license                                        | "9"^^xsd:integer   |
| 16 | sm:copyright                                           | "9"^^xsd:integer   |
| 17 | owl:versionInfo                                        | "9"^^xsd:integer   |
| 18 | rdf:rest                                               | "7"^^xsd:integer   |
| 19 | rdf:first                                              | "6"^^xsd:integer   |
| 20 | http://proton.semanticweb.org/protonsys#transitiveOver | "5"^^xsd:integer   |
| 21 | owl:equivalentClass                                    | "4"^^xsd:integer   |
| 22 | owl:inverseOf                                          | "4"^^xsd:integer   |
| 23 | cdmc-core:field_reference                              | "4"^^xsd:integer   |
| 24 | owl:unionOf                                            | "3"^^xsd:integer   |
| 25 | https://iso.org/iso-11179/v0/Property                  | "1"^^xsd:integer   |
| 26 | owl:allValuesFrom                                      | "1"^^xsd:integer   |
| 27 | skos:example                                           | "1"^^xsd:integer   |
| 28 | skos:scopeNote                                         | "1"^^xsd:integer   |

④RDFストア内、主語当たりの平均プロパティ数を調べる

```
SELECT (AVG(?count) AS ?average)
WHERE {
SELECT ?s (COUNT(?p) as ?count)
WHERE {
?s ?p ?o .
}
}
GROUP BY ?s
}
```

|   | average                                   |
|---|-------------------------------------------|
| 1 | "7.045081967213114754098361"^^xsd:decimal |

⑤RDFストア内、リソースが属するクラスの一覧を使用頻度順に取得する

```
SELECT ?type (COUNT(?s) AS ?count)
WHERE {
?s a ?type .
}
GROUP BY ?type
ORDER BY DESC(?count)
```

|    | type                             | count             |
|----|----------------------------------|-------------------|
| 1  | owl:Class                        | "91"^^xsd:integer |
| 2  | owl:ObjectProperty               | "89"^^xsd:integer |
| 3  | rdf:Property                     | "33"^^xsd:integer |
| 4  | owl:Ontology                     | "9"^^xsd:integer  |
| 5  | owl:Restriction                  | "8"^^xsd:integer  |
| 6  | owl:AnnotationProperty           | "6"^^xsd:integer  |
| 7  | owl:TransitiveProperty           | "4"^^xsd:integer  |
| 8  | owl:SymmetricProperty            | "4"^^xsd:integer  |
| 9  | rdfs:Class                       | "3"^^xsd:integer  |
| 10 | rdfs:Datatype                    | "3"^^xsd:integer  |
| 11 | rdf:List                         | "1"^^xsd:integer  |
| 12 | rdfs:ContainerMembershipProperty | "1"^^xsd:integer  |

## 【参考】 オントロジー、セマンティックWeb、グラフDB関連参考書

- 知の科学 オントロジー工学  
溝口理一郎 著 人工知能学会 編 2005年 オーム社
- オントロジー構築入門  
古崎晃司 他3名著 溝口理一郎 編 2006年 オーム社
- セマンティック・ウェブのためのRDF/OWL入門  
神崎正英 著 2005年 森北出版
- オントロジー工学の理論と実践  
溝口理一郎 著 人工知能学会 編 2012年 オーム社
- 実践セマンティックWeb (RDF/RDFS/OWLによるオントロジー設計ガイド)  
Dean Allemang, James Hendler 著  
セマンティックWeb委員会 訳 (2010年3月) 発行: (株)ジャストシステム
- オープンデータ時代の標準Web API SPARQL  
加藤他4名 著 (2015年11月) 発行: インプレスR&D
- Graph Databases (2<sup>nd</sup>.Edition) 2015, O'Reilly books,  
Ian Robinson, Jim Webber, Emil Eifrem 著
- The Knowledge Graph Cookbook - Recipes that work (1<sup>st</sup> Edition) 2020,  
edition mono/monochrom (Austria) Andreas Blumauer, Helmut Nagy 著

47



インフォラボ游悠 (PB)

中岡 実

[minoru.nakaoka@infolabyouyou.com](mailto:minoru.nakaoka@infolabyouyou.com)

<https://info.infolabyouyou.com>

48