

ドメイン知識の記述支援と活用のためのデジタル資産管理システム基盤「KASHIWADE」の開発 – 情報システム開発における設計書・設計知識への適用

Development of the platform system for Digital Asset Management with Domain Knowledge – A Case Study on Design Documents and Knowledge in Information System Development

中村 覚^{*1} 大和 裕幸^{*1} 稗方 和夫^{*1} 岡田 伊策^{*1*2} 齋藤 稔^{*2} 笈田 佳彰^{*2}
Satoru NAKAMURA Hiroyuki YAMATO Kazuo HIEKATA Isaac OKADA Minoru SAITO Yoshiaki OIDA

^{*1} 東京大学
THE UNIVERSITY OF TOKYO

^{*2} 富士通株式会社
FUJITSU LIMITED

In this study, the platform system for Digital Asset Management which manages digital assets such as documents or images and domain-specific knowledge integrally with Semantic Web was developed. This system provides the functions to support the description of domain knowledge with nodes and links, and to relate it to digital assets with metadata extracted automatically from them. This system was applied to the management of design documents and knowledge in Information System Development, and the effectiveness of developed system was verified.

1. はじめに

情報技術の発展に伴い、Microsoft Office 文書やテキストファイル、画像などの多様な形式のデジタル資産(以下、アセットと呼ぶ)が蓄積されている。これらには有益な情報が含まれることが多く、その再利用が望まれている。しかし、その量が膨大となった場合などには、人間が必要な情報を正確かつ短時間で検索することは困難となる。この課題に対する解決方法の1つとしてセマンティックウェブ技術が挙げられる。セマンティックウェブは情報リソースに意味を付与し、計算機による自律的な処理を可能とする技術である。情報リソースに記述された情報をメタデータとしてインデックス化し、さらにその概念階層やデータ構造の規約をオントロジーとして与えることによって、意味構造に基づいた知的処理を実現する[古崎 07]。

この技術を適用した研究事例として、川村らはウェブサービス検索エンジンを開発し、適切な情報へアクセスするためのインデックスとしてメタデータを利用している[川村 05]。また吉岡らはデータスキーマの変換、知識集約型工学のためのフレームワーク(KIEF)を開発し、人やシステム間における知識の相互運用性を高める知識共有の媒体としてオントロジーを利用している[関谷 99]。さらにオントロジーを知識モデルの規約として利用した研究として、古崎らはオントロジエディタ「法造」を用い、実規模の石油精製プラントオントロジーを構築している[古崎 02]。

このようにセマンティックウェブ技術による知識管理の有用性は数多く示されている。またオントロジーの記述手法に関する研究やオントロジエディタ等の記述支援ツールの開発も行われている。しかし、アセット管理へのセマンティックウェブ技術の実適用を考えた場合、次のような課題点が存在する。まず対象とするドメインのアセット管理の目的によって、インデックスとして必要なメタデータが異なり、またその抽出方法が多様である点である。例えば管理対象とするアセットがフォーマットに基づいて作成されている場合、そのフォーマット構造を活用したメタデータ抽出が有用である。つまり目的に応じたメタデータの抽出方

法を柔軟に変更可能な環境が必要である。次に、対象とするドメインにおける暗黙知または形式知化されていない共通語彙などの知識(以下、ドメイン知識と呼ぶ)をアセットに関連付けるための方法である。知識の記述はオントロジエディタ等のツールを用いて構築する方法が一般的であるが、この知識をアセットのメタデータに紐づけて管理するための環境は整えられていない。したがってメタデータを用いたアセットの管理を行う環境と、外部ツールを用いて構築された知識体系を連携するための環境が必要となる。

この課題の解決に向けて、本研究ではセマンティックウェブ技術を実適用する際のシステム基盤の開発を目的とする。また情報システム企業におけるパッケージシステム開発における設計書およびドメイン知識の管理に本システムを適用し、その有用性の検証を行う。

2. セマンティックウェブ技術を用いたアセット管理手法

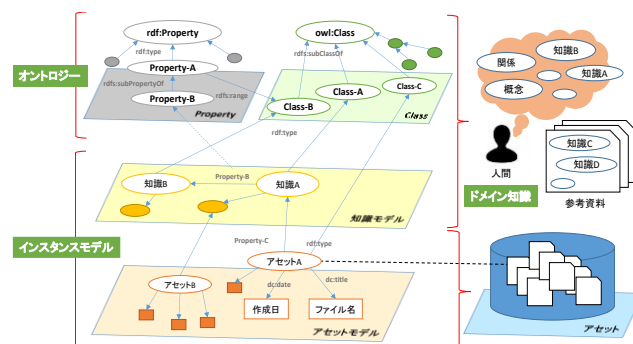


図1 アセット管理のための知識体系

セマンティックウェブ技術を用いたアセット管理を実現するための知識体系を図1に示す。大きく「オントロジー」と「インスタンスモデル」の2階層に分けられる。また、「インスタンスモデル」層については構築方法の違いから、オントロジーが提供する概念定義に基づいて構築する「知識に関するインスタンスモデル(以下、知識モデルと呼ぶ)」と、アセットの内容や情報を記述する「アセットに関するインスタンスモデル(以下、アセットモデルと

呼ぶ)」に分けている。以下では、セマンティックウェブ技術に関する知識を有する技術者が、対象ドメインの有識者と管理対象のアセットを用い、この知識体系を構築するための流れを示す。

2.1 オントロジーの構築

まず対象とするドメインの有識者へのヒアリングやインタビューを通じ、ドメインで用いられる「概念」の集合と、それら概念間の関係を記述する。概念間の関係の記述においては、is-a 関係に基づく概念の階層関係や、part-of (全体-部分) 関係、attribute-of (属性) 関係等について、RDF スキーマが提供している rdfs:subClassOf や rdfs:Property を用いる。この構築には、Protégé や法造などのオントロジエディタの利用を基本とする。

2.2 知識モデルの構築

構築したオントロジーをメタモデルとして利用し、定義した概念におけるオブジェクトをインスタンスとして表現する。さらにそれらの関係を記述したインスタンスモデルをドメインにおける知識モデルとして構築する。この知識モデルの構築には、有識者へのヒアリング結果に基づいて人間が記述する他、これらの関係が記述された参考資料から機械的に抽出する。

2.3 アセットモデルの構築

ファイルとして存在するアセットをインスタンスモデルとして記述し、アセットに記述されている内容をメタデータとして付与する。アセットの「ファイル名」や「作成日」といった基本的なメタデータに加え、知識モデルと関連付けるための情報を抽出する。膨大なアセットに対して、人手によるメタデータの抽出には多大なコストを要するため、計算機を用いた機械的なメタデータ抽出を基本とする。

2.4 知識モデルとアセットモデルの関連付け

ドメインにおける知識モデルとアセットモデルを関連付けることによって、アセットに対する計算機の知的処理を実現する。具体的には、オントロジーを参照した推論を行うことによって、直接記述されていない関係を抽出し、知識モデルで記述された関係に基づくアセットの検索や管理が可能となる。

3. 開発したデジタル資産管理システム基盤

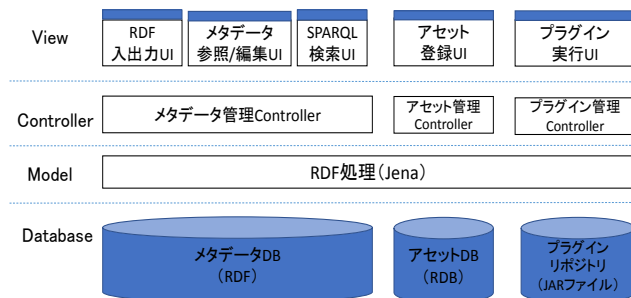


図2 デジタル資産管理システム基盤の概要

開発したデジタル資産管理システム基盤「KASHIWADE」の概要を図2に示す。システムはJavaを用いて構築されたウェブアプリケーションであり、セマンティックウェブアプリケーションを構築するためのライブラリである「Apache Jena」を利用している。2.1のオントロジー、2.2の知識モデルを管理する「メタデータ管理機能」、2.3のアセットモデルを管理する「アセット管理機能」、および2.4の知識モデルとアセットモデルの関連付けを行う「プラグイン管理機能」の3つに大別し、各機能の説明を行う。なお、本システムは筆者らが開発・公開している、メタデータに基づく

アセット管理を支援するアプリケーションフレームワーク [KASHIWADE]をベースとし、ドメイン知識を活用したアセット管理の知的処理を実現するための機能拡張を行ったシステムである。

3.1 メタデータ管理機能

(1) RDF ファイルのインポート機能

オントロジエディタで構築されたオントロジーをはじめとして、外部ツールを用いて作成された RDF または OWL ファイルに記述された RDF トリプルのセットをメタデータ DB に登録する。この機能を用いることで、2.1で構築したオントロジーをシステムから利用することが可能となる。

(2) メタデータ参照/編集インタフェース

システムに登録されたオントロジー、およびインスタンスモデルに関する RDF トリプルを参照および編集するための機能を提供する。システムに登録された RDF トリプルは、図3に示すメタデータ参照インタフェースを用いて参照できる。図3上部は、ある Property に関する本インタフェースの表示例を示す。対象とする URI に付与されたメタデータが表形式で表示され、RDF トリプルにおける述語、目的語、およびそのタイプ (Resource または Literal) が表示される。タイプが Resource であるメタデータについては、付与されたリンクによって、選択した Resource に関するメタデータ参照インタフェースに遷移することができる。

またタブを切り替えることによって、図3下部に示す推論結果を反映したメタデータの一覧を参照することができる。推論によって、上位の Property において定義された Domain および Range に関する RDF トリプルが追加されている例を示す。

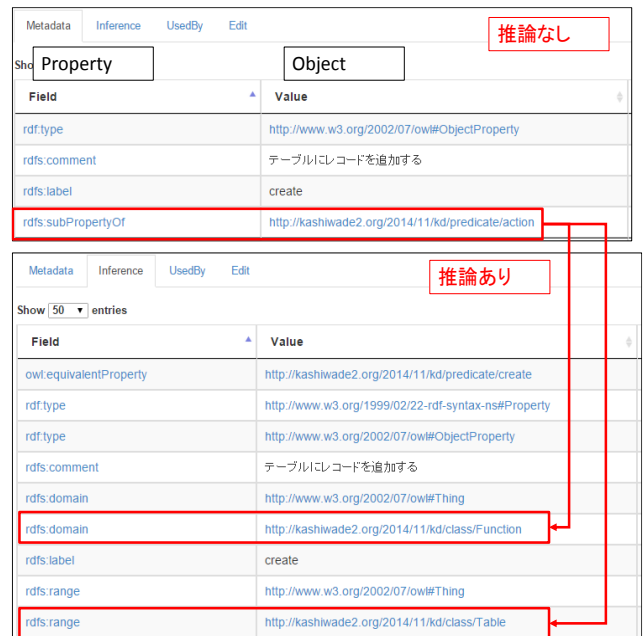


図3 メタデータ参照インタフェース

(3) SPARQL 検索機能

SPARQL による検索を行うことで、システムに登録された RDF トリプルを抽出する。また、本システムは検索結果の可視化機能を提供する。具体的には、検索結果に含まれる RDF トリプルの出現割合や出現数をグラフ化して表示する。この可視化機能を用いることで、システム内に登録された RDF トリプルの全体像を把握することができる。

3.2 アセット管理機能

2.3のアセットモデルを取り扱うための機能を提供する。具体的には、アセット登録 UI を用いてアセットをシステムにアップロードすることで、自動的に URI が付与され、その URI を主語とした RDF トリプル(アセットモデル)がメタデータ DB に登録される。またアセット名や登録年月日などの基本的なメタデータが自動的に付与される。さらにアセットのバイナリデータはアセット DB に登録され、付与された URI を参照することによって、そのアセットをダウンロードすることができる。

3.3 プラグイン管理機能

アセット DB に登録されたバイナリデータ、およびメタデータ DB に登録された RDF トリプルのセットを引数として、新たな RDF トリプルをメタデータ DB に登録するプラグインの実行環境を提供する。プラグインはシステムとは独立した JAR ファイルとして開発され、システムはこれらプラグインを呼び出して実行する。このプラグイン機能を用いることによって、利用者は目的に応じたメタデータを機械的に抽出することができ、またそのアルゴリズムの実装に注力することができる。

プラグインの利用目的は、大きく 2 つに分けられる。1 点目は 2.3 で述べたアセットからのメタデータ抽出であり、アセットのバイナリデータを引数として、記述された情報を抽出する。例えばテキスト情報からのキーワード抽出などが挙げられる。2 点目の目的は、2.4 で述べた知識モデルとアセットモデルの関連付けである。プラグインによって抽出したアセットに関するメタデータを用いて、知識モデルとの関連付けを行うためのアルゴリズム例を図 4 に示す。例えば「知識 C」という Literal 形式のメタデータを持つ「アセット A」に対し、それと同じ文字列をラベル名として持つインスタンス「知識 C」が関連付けられる。システムに登録されたアセットとドメイン知識を関連づけることにより、知識モデルとして記述した関係に基づくアセットの検索が可能となる。

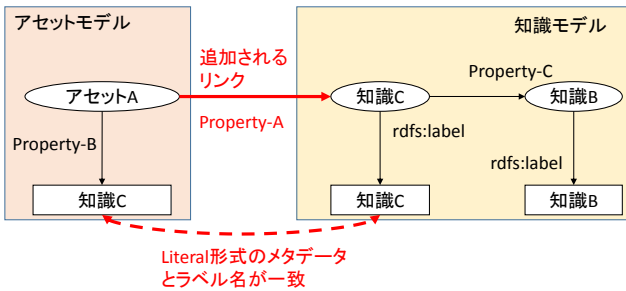


図 4 知識モデルとアセットモデル関連付けの例

4. KASHIWADE の適用事例

稗方らはパッケージシステム開発で作成される設計書の再利用性の向上を目的とした「項目 CRUD 検索」を提案した[稗方 14]。この検索手法は、ER 図に記述された「マスタ」とそのマスタが持つ「項目」の関係、CRUD 図に記述された「機能」とそれが作用する「マスタ」の関係、および各設計書に記述された「機能」に関するメタデータを相互に関連付け、ある「項目」に仕様変更等が発生した際に、この変更に影響する設計書を機械的に抽出する手法である。本章では、この「項目 CRUD 検索」の実現に必要な知識体系を KASHIWADE 上で構築することを目的とする。なお、RDF や RDF スキーマで提供される語彙の他に、「Class」、「Property」、「Instance」を区別するための接頭名詞空間として、それぞれ「c:」、「p:」、「i:」を用いる。

(1) オントロジーの構築

まず対象とするドメイン知識をオントロジーとして記述する。本ケーススタディでは、対象とするパッケージシステム開発における語彙をライトウェイトオントロジーとして構築した。その例を図 5 に示す。なお、オントロジエディタとして Protégé を利用した。

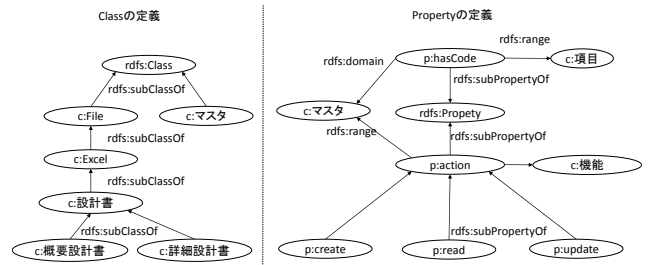


図 5 構築したオントロジー例

Class に関しては、対象とするパッケージシステム開発で作成される概要設計書や詳細設計書等の設計書の種別、およびパッケージシステムが提供する「機能」や「マスタ」、「項目」等の語彙を定義した。さらに Class 間の階層関係について、RDF スキーマの語彙の一つ「rdfs:subClassOf」を用いて記述した。

また、ER 図に記述されている「マスタ」と「項目」の関係を記述するための Property を定義し、合わせて Class「マスタ」を Domain、Class「項目」を Range として定義した。さらに CRUD 図に記述されている「機能」とそれが作用 (Create, Read, Update, Delete) する「マスタ」の関係を記述するための Property の定義を行った。これらの Property については、「機能」と「マスタ」の関係を記述するための Property として定義した「p:action」のサブプロパティとして定義した。このように定義したオントロジーを OWL ファイルとして出力し、KASHIWADE にインポートする。

(2) 知識モデルの構築

前項で定義した Class のインスタンスの関係を記述する。本ケーススタディでは Excel ファイルに記述された「ER 図」と「CRUD 図」からこれらの関係を抽出した。この記述例を図 6 に示す。

まず ER 図から Class「マスタ」「項目」のインスタンスの関係を抽出した。例えば Instance「組織マスタ」と Instance「組織コード」の関係を Property「p:hasCode」によって紐づけた。また CRUD 図からは Class「機能」「マスタ」のインスタンスの関係を抽出した。例えば Instance「店舗マスタ登録」と Instance「組織マスタ」の関係を Property「p:read」を用いて記述した。この情報抽出については、Microsoft Office 形式のファイルを取り扱うためのライブラリ「Apache POI」を利用したバッチプログラムを作成し、抽出結果として得られた RDF ファイルを KASHIWADE にインポートした。

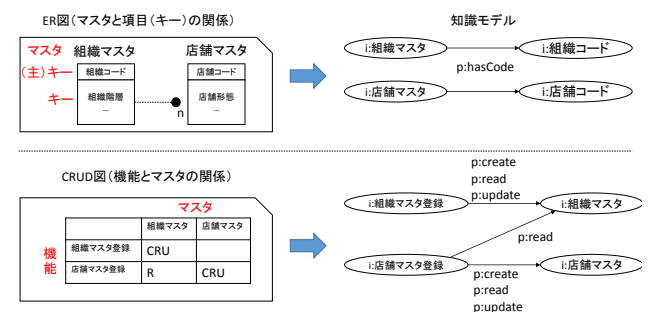


図 6 ER 図、CRUD 図からの知識モデル記述例

(3) アセットモデルの構築

管理対象とする設計書からのメタデータ抽出を行う。対象とするパッケージシステムが提供する「機能」に関する設計書について、KASHIWAVE のアセット登録機能を用いてアセット DB に登録する。次にプラグイン機能を用いて、その設計書に記述された機能名を自動的に抽出する。対象とした設計書はドメイン固有のフォーマットに基づいて作成されており、設計書が対象とする機能名がヘッダーに記述されていた。したがってヘッダーの表構造に基づき、そこに記述された機能名を抽出するプラグインを開発し、その機能名を設計書のメタデータとして登録した。

(4) 知識モデルとアセットモデルの関連付け

これまでに構築した知識モデルとアセットモデルに関する RDF トリプルを引数とし、図 4 に示したアルゴリズムに基づいた関連付けを行うプラグインを開発、実行する。この結果、構築される知識体系を図 7 に示す。また推論機能を適用することによって、赤字で示したノードおよびリンクが追加される。例えば Property「p:read」は Domain として Class「機能」、Range として Class「マスタ」が定義されている Property「p:action」のサブプロパティであるため、それに関連づけられた「店舗マスタ登録」は Class「機能」のインスタンスであり、「店舗マスタ」は Class「マスタ」のインスタンスであるという RDF トリプルが追加されている。

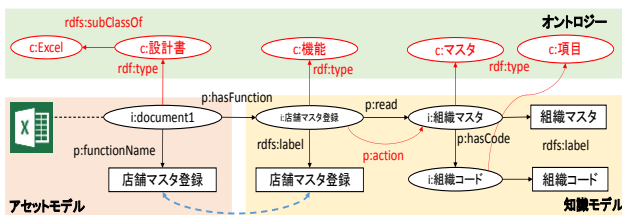


図 7 構築される知識体系

さらにこの知識体系を構築したことによって、先述した「項目 CRUD 検索」を実現できる。例えば、「店舗コード」という項目に作用する機能の設計書の抽出を目的とし、検索対象とする項目を持つマスタ、そのマスタに作用する機能、その機能に関連する設計書、という RDF トリプルを条件として与えた場合における検索結果の可視化例を図 8 に示す。この例では、「店舗コード」に関する仕様変更が生じた際、「店舗マスタ登録」という機能に関する 511 個の設計書に影響が生じることを示している。また稗方らは、この「項目 CRUD 検索」を用いた被験者実験を行い、検索結果における再現率の向上などの有用性を示している。

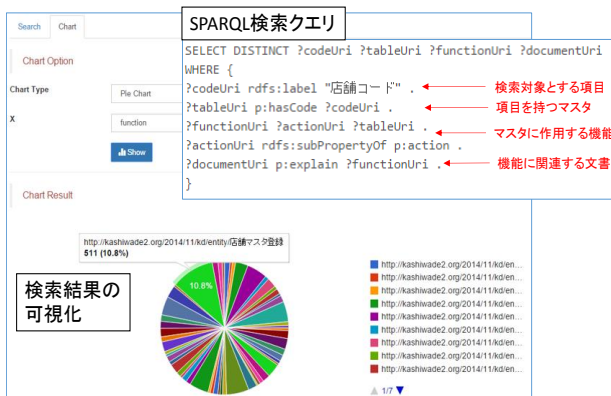


図 8 検索結果の可視化例

5. 考察

本研究ではアセット管理へのセマンティックウェブ技術の実適用を支援するシステム基盤を開発した。4章の適用事例では、管理対象とするアセットに対して、提案手法で示した「オントロジー」「知識モデル」「アセットモデル」、およびそれらの関連付けの各手順を KASHIWAVE 上でを行い、セマンティックウェブ技術による知識体系を構築した。また「項目 CRUD 検索」のような知識処理を実現でき、適用事例は開発したシステムの有用性を示す一例であると考ええる。この手順については、ドメインによらず共通である。またドメイン固有の部分については、プラグインによるメタデータ抽出を行う環境を構築することで、ドメインの目的に応じた柔軟なメタデータ抽出が可能となった。さらにオントロジェディタ等の外部ツールやバッチプログラムによって作成した RDF ファイルに記述された RDF トリプルのセットをインポートする機能を構築することで、アセットに付与されたメタデータに基づくアセットとドメイン知識の関連付けが可能となった。

対象とするドメインへのアセット管理の実適用を考えた際には、データ構造だけでなく、ドメインのニーズに応じた UI 開発も重要である。本論文ではデータの記述方法に主眼を置いて説明したが、KASHIWAVE は他のウェブアプリケーションからの検索要求に応答する SPARQL Endpoint としての機能を有する。したがって検索結果として得られる JSON 形式のデータに基づいて、柔軟な UI 開発が可能となり、様々な分野におけるアセット管理において、本システム基盤が適用可能であると考えられる。

6. 結論

本研究ではセマンティックウェブ技術を用い、ドメイン知識の記述支援と活用のためのデジタル資産管理システム基盤「KASHIWAVE」を開発した。対象ドメインに依存しないアセット管理機能やメタデータ管理機能については本システム基盤が提供し、ドメイン依存の部分については、オントロジェディタ等の外部ツールやプラグインの利用を可能とする環境を構築した。今後は他分野のデジタル資産管理に本システム基盤を適用し、システムの改良や評価を行う。

参考文献

[古崎 07] 古崎晃司, 北村徳信, 溝口理一郎: オントロジー工学的立場から見たオントロジー利用研究の分類と考察, 21 回人工知能学会全国大会論文集(JSIAI07), 1D3-3, 2007

[川村 05] 川村隆浩, 長谷川哲夫, 大須賀昭彦, Massimo Paolucci, Katia Sycara: セマンティック Web サービスマッチメーカーの公開実験に基づく評価, 人工知能学会論文誌, Vol. 20, No. 6, pp.426-436, 2005

[関谷 99] 関谷貴之, 吉岡真治, 富山哲男: オントロジーを用いた統合的設計支援環境の実現, 人工知能学会論文誌, Vol. 14, No. 6, pp. 1051-1060, 1999

[古崎 02] 古崎晃司, 来村徳信, 佐野年伸, 本松慎一郎, 石川誠一, 溝口理一郎: オントロジー構築・利用環境「法造」の開発と利用—実規模プラントのオントロジーを例として, 人工知能学会論文誌, Vol.17, No. 4, pp. 407-419, 2002

[KASHIWAVE] metadata based application framework 「KASHIWAVE」, <http://sourceforge.net/projects/kashiwade/>

[稗方 14] 稗方和夫, 大和裕幸, 深田直人, 中村覚, 岡田伊策, 齋藤稔, 笈田佳彰, 渡辺郁雄, 松本滋: システムの仕様変更調査における設計情報を用いた影響分析システムの開発, 第 24 回設計工学・システム部門講演会, 日本機械学会, 2014