

メタデータスキーマの再利用を指向したスキーマ設計支援システム

庄山 和男[†], 永森 光晴[‡], 杉本 重雄[‡]

[†]筑波大学図書館情報専門学群, [‡]筑波大学図書館情報メディア研究科

E-mail:s0312168@ipe.tsukuba.ac.jp, {nagamori, sugimoto}@slis.tsukuba.ac.jp

概要

本研究では、既存のメタデータスキーマを基礎にして、メタデータスキーマの作成を支援するシステムの開発に取り組んだ。本研究で開発したシステムは、Dublin Coreのアプリケーションプロファイルの概念を基礎として、応用目的にあった新しいスキーマを、既存スキーマから作るプロセスを支援する。本システムは、目的のスキーマに再利用可能な記述項目(エレメント)を既存スキーマの中で検索し、見つけたエレメントを基礎として新規メタデータスキーマの定義作業を支援する機能を持つ。

キーワード

メタデータスキーマ, アプリケーションプロファイル, メタデータスキーマレジストリ, メタデータの相互利用性, メタデータスキーマの再利用

A Software Tool to Support Metadata Schema Development by Schema Search-and-Reuse

Kazuo SHOYAMA[†], Mitsuharu NAGAMORI[‡], Shigeo SUGIMOTO[‡]

[†]School of Library and Information Science, University of Tsukuba,

[‡]Graduate School of Library, Information and Media Studies, University of Tsukuba

E-mail:s0312168@ipe.tsukuba.ac.jp, {nagamori, sugimoto}@slis.tsukuba.ac.jp

Abstract

Metadata has been recognized as an important issue for the Internet and digital libraries. In general, we need to define a metadata schema(s) which is tailored to our applications and services. However, in general, it is expensive to develop a metadata schema from scratch. This paper describes a system to support metadata schema development based on Dublin Core Application Profile and the XML/RDF technology. The system help a user find metadata schemas and metadata elements from a metadata schema database and define a metadata schema as an application profile in accordance with the user's requirements. In other words, this system help users reuse existing schemas, which will not only decrease the cost of schema development but also enhance interoperability of metadata.

Keywords

Metadata Schema, Application Profile, Metadata Schema Registry, Metadata Interoperability, Metadata Schema Reuse

1 はじめに

現在インターネットの発達とともに、ネットワーク上で提供されるリソースは増加の一途をたどっている。その膨大な資源の中から、より目的に適合した情報を発見し、利用するためにメタデータの重要性が広く認められている。

現在までに様々なコミュニティにおいて、それぞれの目的に合ったメタデータスキーマの開発が行われてきた。メタデータスキーマ開発者はメタデータスキーマの開発において、メタデータの相互利用性を重視するか、コミュニティの利用目的に特化したメタデータ記述の要望を満たすことを重視するか、いずれかを選択しなくてはならないというジレンマに陥りがちである。前者では自身の利用目的に特化したメタデータ記述を排除することになり、後者では新規にメタデータスキーマを開発しなければならない。また、ネットワーク上で利用されるメタデータスキーマを記述するためにはXMLやXML/RDF等の知識が必要であることや、多大な時間をかけて定義したメタデータスキーマの要素を再利用するための具体的方法も普及していない。

これら新規メタデータスキーマ作成に関わるジレンマや要素の再利用といった課題を解決するには、既存メタデータスキーマと要素を発見し再利用するためのモデルと、応用目的にあった新しいスキーマを既存スキーマから作るプロセスを支援するツールが必要であると考えられる。そこで本研究では、既存メタデータスキーマから利用目的に沿った要素を見つけ出し、それらを組み合わせて新たなメタデータスキーマを作成するためのモデルの提案と、本モデルに基づいたメタデータスキーマ設計支援システムについて述べる。

2 メタデータスキーマとメタデータスキーマ開発

2.1 メタデータスキーマ

メタデータスキーマとは、メタデータの記述形式を決める規則である。メタデータスキーマ[14]を用いることで、記述対象とするリソースの分野や言語の違い関わらずメタデータを統一的に記述することができ、ウェブ上のリソース発見の効率化を図ることができる。代表的なメタデータスキーマには、国際標準となっているDublin Core Metadata Element Set (DCMES)[5]がある。DCMESはウェブ上のリソースのメタデータを記述するためのもので、Dublin Core Metadata Initiatives(DCMI)[4]によって定義されている。DCMESでは、語彙を定めるだけでその記述法や構文は定義しておらず、RDFやHTMLなどで記述するための推奨方法は別途提案されている。

本研究では、メタデータスキーマは下記の要素からなると定義する。

(1) 属性集合(属性語彙)

メタデータとして記述するリソースの属性の定義。例えば、タイトル、著者、出版者など、属性を表す語とその集合の定義。

(2) 属性値型集合(属性値型語彙)

メタデータとして記述する属性値の記述形式の定義。例えば、日付や名前の記述形式、主題を表す統制語彙など、属性値の型あるいは形式を表す語とその集合の定義。

(3) 構造的制約(抽象構文)

属性値記述の省略可能性や繰返し条件などメタデータの構造的な制約。構造的制約は、具体的なシステム上でのメタデータの実現形式に依存しないメタデータの構文規則を決める。

(4) 実現形式(具象構文)

システム上でのメタデータの具体的表現形式。

一般にメタデータ記述とは、リソースが持つ属性と属性値の組の記述の集まりと捕らえることができる。本論文では上に示した属性集合(属性語彙)をエレメントセットとし、その要素を記述項目(エレメント)と呼ぶ。

2.2 メタデータスキーマレジストリ

メタデータスキーマレジストリは、メタデータスキーマの正確な情報の提供や、メタデータスキーマの流通性を高めるといった役割を担う。メタデータスキーマレジストリは、メタデータスキーマ閲覧や検索をする機能を持つ。メタデータ作成者にとってはメタデータ作成のためのリファレンスとなる。メタデータスキーマ開発者にとっては、リファレンスとしてだけでなく、既存メタデータスキーマを発見するためのデータベースとしてメタデータスキーマ開発に利用することができる。

1999年頃からDCMIレジストリワーキンググループ[11]では、メタデータスキーマレジストリに関する議論と実装が行なわれている。DCMIレジストリワーキンググループでは、メタデータスキーマ開発者のメタデータスキーマの発見、メタデータスキーマ開発の支援、そして各種のソフトウェアから利用するためのアプリケーションインタフェースの構築を目指している。

DCMIメタデータスキーマレジストリ[12]では、DCMIによって定義されたメタデータスキーマに関する情報を提供している。また、多言語への対応を指向しており、2007年1月現在で26ヶ国語によるDCMESが提供されている。メタデータスキーマの記述形式にはRDFを採用し、RDF形式のメタデータスキーマであればDCMIによって定義されたもの以外も登録可能である。レジストリの機能には、スキーマの検索と閲覧、スキーマの修正、登録の管理、ウェブサービスを利用したアプリケーションインタフェースの提供等がある。

われわれの研究室では、筑波大学レジストリとして、DCMIの定義するメタデータスキーマ以外のメタデータスキーマをレジストリに登録する試みを行っている[13]。筑波大学レジストリでは、60種類以上のメタデータスキーマを収集・蓄積し、提供している。

2.3 関連研究

メタデータスキーマを収集し検索、閲覧可能にする試みが、幾つかのプロジェクトにより行われている。ここではSwoogle[6]とSchemaWeb[7]について述べる。

Swoogleは、ウェブ上で公開されているセマンティックWEB文書を対象とした専門検索エンジンで、RDF文書やHTMLに埋め込まれたRDF、N3文書、メタデータスキーマ等を収集している。収集された文書の関係が計算されインデックスがつけられている、検索結果はオントロジーの統計等に基づいてランキング表示される。Swoogleでは、ウェブサービスとしてソフトウェアエージェントへのサービスも提供している。

SchemaWebでは、登録されたメタデータスキーマに対してディレクトリ型の閲覧、検索が可能である。登録されたスキーマの閲覧、スキーマに含まれるエレメント情報の閲覧、スキーマの全文検索、蓄積されたスキーマに関する情報を呼び出すことができるウェブサービスが提供されている。Swoogleではメタデータスキーマへのリンクのみを提供しているが、SchemaWebではメタデータスキーマの詳細(プロパティ、クラス等の一覧)が閲覧可能になっている。

SwoogleとSchemaWebではメタデータスキーマの検索と閲覧機能の提供は行っているが、メタデータスキーマ作成の機能は提供していない。

3 メタデータスキーマ作成支援モデル

本研究におけるメタデータスキーマの設計はDublin Coreのアプリケーションプロファイル[8]の概念を基礎として、既存メタデータスキーマから利用目的に沿ったエレメントを見つけ出し、それらを組み合わせることで利用目的に沿った新規メタデータスキーマの設計を行う。

本モデルでは、メタデータスキーマレジストリに登録されたメタデータスキーマからエレメントを応用し新たなスキーマの作成に利用することで、従来のように新規にエレメントを作成する手間を省く。また、アプリケーションプロファイルとして記述することにより既存メタデータスキーマの再利用性を向上させることができる。作成したメタデータスキーマをレジストリに登録することで、アプリケーションプロファイルとして記述したメタデータスキーマについても調査の対象とし、それを再利用可能とすることで再利用されたスキーマと再利用したスキーマ間での相互利用性を高めメタデータスキーマの設計に好循環を生み出すことができると考える。本研究における新規メタデータスキーマ設計に必要な手順は、A) 既存エレメントの発見、B) 発見したエレメントが要求に適しているか検討、C) メタデータスキーマの記述とする。

A) 既存エレメントの発見

既存エレメントの発見には、既存メタデータスキーマを検索し利用目的に沿ったエレメントの発見を支援するシステムが必要とされる。

全文検索によるメタデータスキーマ検索では、ユーザが入力したキーワードとメタデータスキーマの定義文章内に含まれるテキストが一致している必要があった。しかし、利用目的に沿ったエレメントを検索するには文字列の一致だけではなくエレメントの利用目的に合わせて探せることが重要である。例えば、文字列の一致だけでは検索キーワードとして「title」のようにエレメント名に相当する項目名を入力する場合においても、本の「title」と人の肩書きの「title」といった意味的に異なるものも区別無く検索されてしまう。また、利用目的を「Personal」のように入力する場合、文字列の一致のみでは、検索もれが大きくなる恐れがある。その原因の一つとしてメタデータスキーマのRDF記述の中には、利用目的を示す単語が記述されていないものや、利用目的を明確に説明していないものが多いことがあげられる。そこで、メタデータスキーマ及びそれに含まれるエレメントを意味的な要素も含めて検索できるシステムが要求される。

本研究ではメタデータスキーマ及びそのエレメントに対して利用目的等のコンテキストを与えるカテゴリとの対応付けを与える。全文検索とカテゴリ検索を組み合わせることで、ユーザの要求に一致したメタデータスキーマの発見を適切にナビゲートするシステムを実現する。エレメントのカテゴリはDublin Coreを参考に作成した。Dublin Coreは含まれるエレメント数は少ないが、広い範囲を示すエレメントが定義されているため、カテゴリ作成の参考に適していると考えたからだ。Dublin Coreのエレメントでも意味的に重複するものは統合し、Dublin Coreの概念に含まれないものに関しては随時追加していく。スキーマのカテゴリはスキーマを登録する際に随時スキーマの利用目的を示すワードを付与することとした。

表 1 本研究でDublin Coreを参考に作成したエレメントのカテゴリ一覧

カテゴリ名	カテゴリの定義
name	それに与えられた名称
subject	その内容の中心となる事柄
description	その内容の説明、記述
contributor	その生成に関わった実体
date	そのライフサイクルにおけるなんらかの事象の日付
type	その内容の性質又はジャンル
format	その物理的形態又はデジタル形態での表現形式
identifier	それを一意に特定するための識別子
source	それが作り出される源になった資源、物事への参照
language	その知的内容を表すのに用いられる言語
relation	関連資源、物事への参照
coverage	その内容が表す範囲又は領域
rights	それに含まれる、又はかかわる権利に関する情報
audience	それが意図する、または役に立つ対象である実体
rightsHolder	それに対して権利を所有している、または管理している人または組織
provenanve	その確実性、安全性、解釈のために重要である、それが作られてからの所有と保管における変遷
pedagogy	教育学に含まれる実体・説明・記述

B) 発見したエレメントが要求に適しているか検討

エレメントが要求に適しているか検討するためには、エレメントが属しているスキーマ情報及び関連するエレメントに対する理解も必要である。メタデータスキーマはテキストで記述されることが多く、視覚化というのはあまり行われてこなかった。しかし、近年メタデータの世界でもXML/RDFの普及とともにRDFグラフが様々な場面で使用されはじめ、メタデータを視覚化する動きがみられる。前述した関連研究でのメタデータスキーマ閲覧画面において、メタデータスキーマ情報はテキストで提供されている。本研究ではそれらを改善し、閲覧画面や検索結果画面にグラフ図での描画を用いることでメタデータスキーマ及びエレメント情報の直感的な理解を促す。

C) メタデータスキーマの記述

スキーマの記述には、XMLやXML/RDF等の知識が必要である。ユーザであるメタデータスキーマ開発者、学習者等はその知識を会得している必要があり、初めてメタデータスキーマを開発しようとしている者など、XML/RDF等の知識を持たない利用者にとって、メタデータスキーマの作成はハードルの高いものとなっていた。前述した関連研究では、既存スキーマの検索や閲覧機能は提供されているが、メタデータスキーマの新規作成機能は提供されていない。本研究では、Dublin Coreのアプリケーションプロファイルの概念を基礎として、応用目的にあった新しいスキーマを作成する。新規メタデータスキーマの情報(名前や利用目的等)や既存のスキーマから再利用したエレメントの制約条件を対話的に加えていくことで新規メタデータスキーマの作成を可能にする。

4 メタデータスキーマ設計支援システム

本研究では、3章で述べたモデルに基づいたメタデータスキーマ設計支援システムを構築した。

本システムは、既存メタデータスキーマから利用目的に沿ったエレメントの発見を支援し、それらエレメントを組み合わせた新規メタデータスキーマの作成を行う。

既存メタデータスキーマを検索し利用目的に沿ったエレメントを発見する検索機能として、キーワード入力の際にリアルタイムに候補を表示させるインクリメンタルサーチによる動的な検索を可能とした。さらに、意味的な要素も含めて検索できるように検索の結果表示された候補からエレメントのカテゴリ及びスキーマのカテゴリ(利用目的を端的に示した語彙)による絞込みを行うことができる。図1は、キーワード「name」+カテゴリ「name」でエレメント情報を検索した結果である。

スキーマ情報及び関連するエレメントに対する理解を促す機能としてスキーマ及びエレメント情報の表示機能に通常テキスト情報に加え、カテゴリとの対応付けを元にエレメントの情報及びスキーマの情報をビジュアル化して表示できる。グラフ表示には、我々の研究室で作成したシステム Hybrid And Network-Assisted Vocabulary Interface(HANAVI)[9]のグラフ表示ライブラリと TouchGraph LinkBrowser[10]を用いた。図2、3、4ではキーワード「name」+カテゴリ「name」でエレメント情報を検索した結果をグラフ表示している。図2はHANAVIインタフェースでのグラフ表示、図3は TouchGraphインタフェースでのエレメントとカテゴリのグラフ表示、図4はTouchGraph インタフェースでのスキーマとカテゴリのグラフ表示である。

本システムでは、Dublin Coreのアプリケーションプロファイルの概念に基づいてメタデータスキーマを記述する。検索されたエレメントから要求に一致したエレメントを選択し、それに何回出現可能等の制約条件やメタデータスキーマの利用目的など、必要な項目を対話的に入力することでメタデータスキーマの作成を半自動的に行う。図5に既存エレメントに制約条件を入力するフォーム画面を示す。

システムを利用したメタデータスキーマ設計は、まずエレメントを検索し、情報の理解にはグラフ表示を参考にする。要求に一致したエレメントの選択は一時保存領域へのドラッグアンドドロップで行い、必要なエレメントがそろった所で必要な項目の入力画面に移る。図6にエレメントの一時保存画面を示す。必要な項目を入力し、入力を完了すると新規メタデータスキーマとしてシステムに保存される。保存されたメタデータスキーマは作成者ごとに管理され、エレメントの追加・編集・削除といった編集作業が可能である。また、作成したメタデータスキーマのXML/RDF形式での出力をサポートしている。図7は、本システムを用いて作成したメタデータスキーマのXML/RDF記述である。

5 おわりに

本研究ではメタデータスキーマ開発者のための対話的なメタデータスキーマ開発支援システムを構築した。検索機能では、単純な文字列の検索だけではなく、エレメントに対して付与したカテゴリ情報やスキーマに対して付与したタグ情報を利用した。また、エレメントに含まれるドメイン(エレメントの記述対象がどのような部類に属するか)、レンジ(エレメントの目的語がどのような部類に属するか)といった詳細情報を利用したメタデータスキーマ及びエレメントの発見を試みた。表示機能では通常テキスト情報での表示に加えカテゴリやタグとの対応付けを元にエレメント間の関係及びスキーマ間の関係を視覚化した。スキーマ及びエレメント情報の視覚化は、本システムの利用者であるメタデータスキーマ開発者やメタデータスキーマ学習者にとってメタデータスキーマの発見と利用を支援するために有用である。

今後の課題としてはオフラインでのメタデータスキーマ設計に対応すること、メタデータスキーマレジストリとの相互運用、データベースにおけるXMLのネイティブなサポートなどがある。現在エレメントの定義にドメインとレンジを含むメタデータスキーマは多くないが、今後ドメインとレンジに関する情報を持つメタデータスキーマの増加が予想される。本研究では、カテゴリに加えドメインとレンジの情報を考慮した検索機能を拡張する予定である。また、RDFで記述されていないメタデータスキーマに関しても登録し参照できるようにする仕組みも提案したい。

参考文献

- [1] 杉本重雄「メタデータの基礎概念とモデル」『図書館目録とメタデータ』No.4,東京都, 2004.
- [2] 神崎正英「セマンティック・ウェブのためのRDF/OWL入門」,東京都, 2004.
- [3] 吉野真世「アプリケーションプロファイルを利用したメタデータスキーマ開発支援手法」,つくば, 筑波大学, 2006. 修士論文.
- [4] DCMI. (online), available from <<http://dublincore.org/>>
- [5] DCMI. “Dublin Core Metadata Element Set, Version 1.1” (online), available from <<http://dublincore.org/documents/dces/>>
- [6] Swoogle. (online), available from <<http://swoogle.umbc.edu/>>
- [7] SchemaWeb. (online), available from <<http://www.schemaweb.info/>>
- [8] CEN ISSS MMI-DC Workshop. “Guidelines for machine-processable representation of Dublin Core Application Profiles”. (online), available from <<http://www.ukoln.ac.uk/metadata/cen/ws-mmi-dc/final-132/>>
- [9] HANA VI: Hybrid And Network-Assisted Vocabulary Interface. (online), available from <<http://raus.slis.tsukuba.ac.jp/subjects/graph>>
- [10] SourceForge. “TouchGraph”. (online), available from <<http://sourceforge.net/projects/touchgraph/>>
- [11] DCMI. “DCMI Registry Community” (online), available from <<http://dublincore.org/groups/registry/>>
- [12] DCMI. “The Open Metadata Registry” (online), available from <<http://dublincore.org/dcregistry/>>
- [13] 筑波大学. “The Open Metadata Registry”. (online), available from <<http://juror.slis.tsukuba.ac.jp/dcregistry/index.html>>
- [14] DCMI. “DCMI Abstract Model” (online), available from <<http://dublincore.org/documents/abstract-model/>>

The best way to search, design and share Application Profiles.

[Home](#) | [Browse](#) | [Element Cluster](#) | [Schema Tag](#) | [Edit](#)

e: c: s: [\[help \]](#)

e: エレメントの名前/説明:
c: クラスター名:

キーワードを入力

カテゴリ名を入力

[\[MATRIX \]](#) | [\[RDF \]](#) | [\[HANAUI Graph \]](#) | [\[Touch Graph \]](#)

Element List

[+] **title** s: Dublin Core Element Set v1.1 c: name

label: Title [RDF]

comment: A name given to the resource.

description: Typically, a Title will be a name by which the resource is formally known.

[+] **nick** s: FOAF c: name

label: nickname [RDF]

comment: A short informal nickname characterising an agent includes login identifiers, IRC and other chat

nicknames.

[+] **name** s: FOAF c: name

label: name [RDF]

comment: A name for some thing.

domain: <http://www.w3.org/2000/01/rdf-schema#Thing>

range: <http://www.w3.org/2000/01/rdf-schema#Literal>

subPropertyOf: <http://www.w3.org/2000/01/rdf-schema#label>

[+] **firstName** s: FOAF c: name

label: firstName [RDF]

comment: The first name of a person.

domain: <http://xmlns.com/foaf/0.1/Person>

range: <http://www.w3.org/2000/01/rdf-schema#literal>

ここに利用するエレメントの情報を一時的に保存します。

エレメントの横に表示されている[+]をクリック、またはエレメントをドラッグアンドドロップすることでここに追加することができます。

エレメントが揃ったら [Edit](#) に進んでください。

検索結果

エレメント名 s:スキーマ名 c:カテゴリ名

詳細情報 (label, comment, description,

domain, range, super_property 等)

図 1 エレメントの検索とテキスト表示

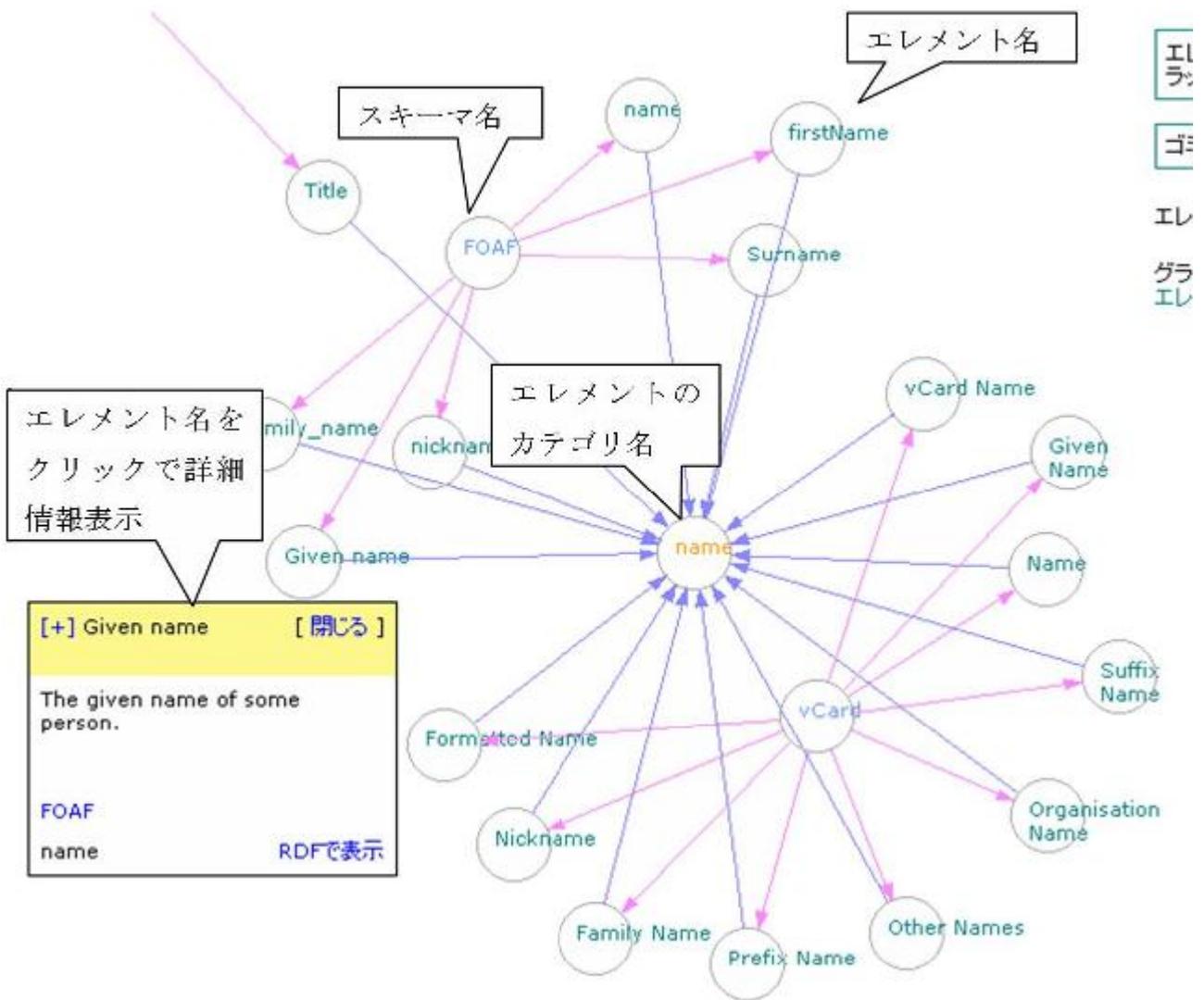


図 2 エlementとカテゴリを繋ぐグラフ表示 (HANAVIインターフェース)

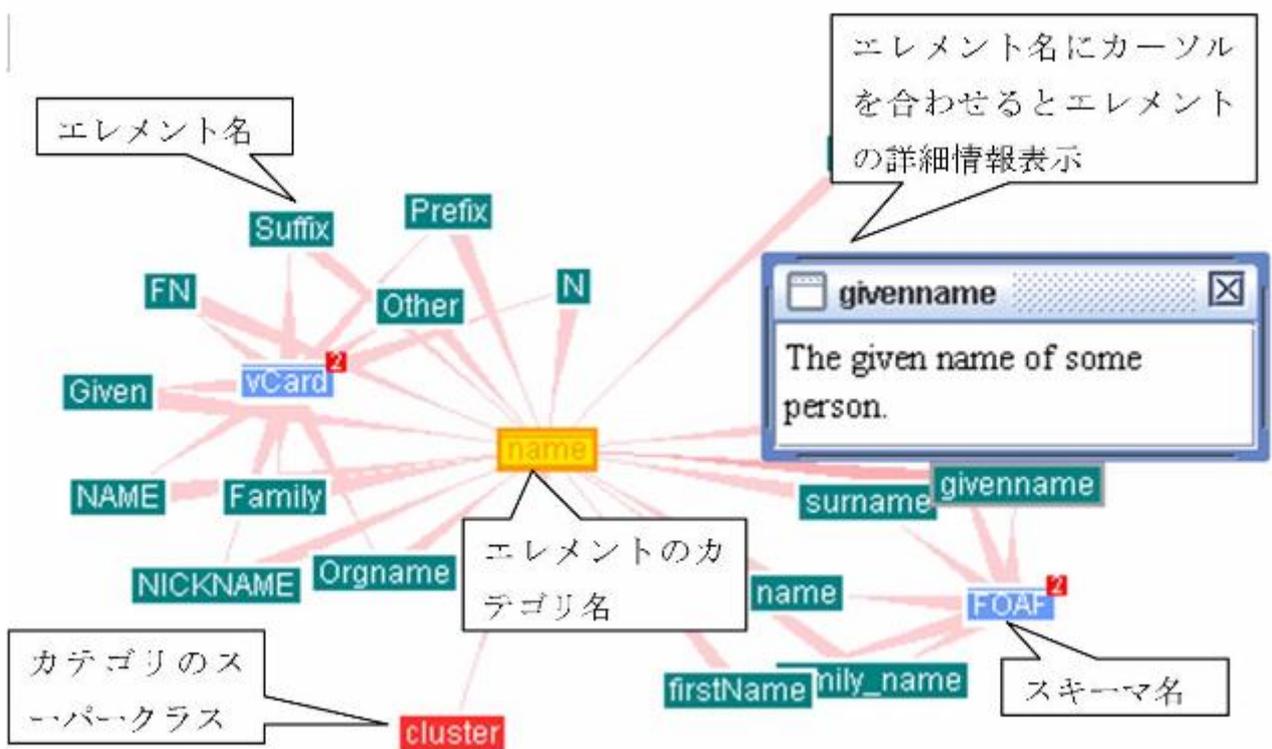


図 3 エlementとカテゴリを繋ぐグラフ表示 (TouchGraphインタフェース)

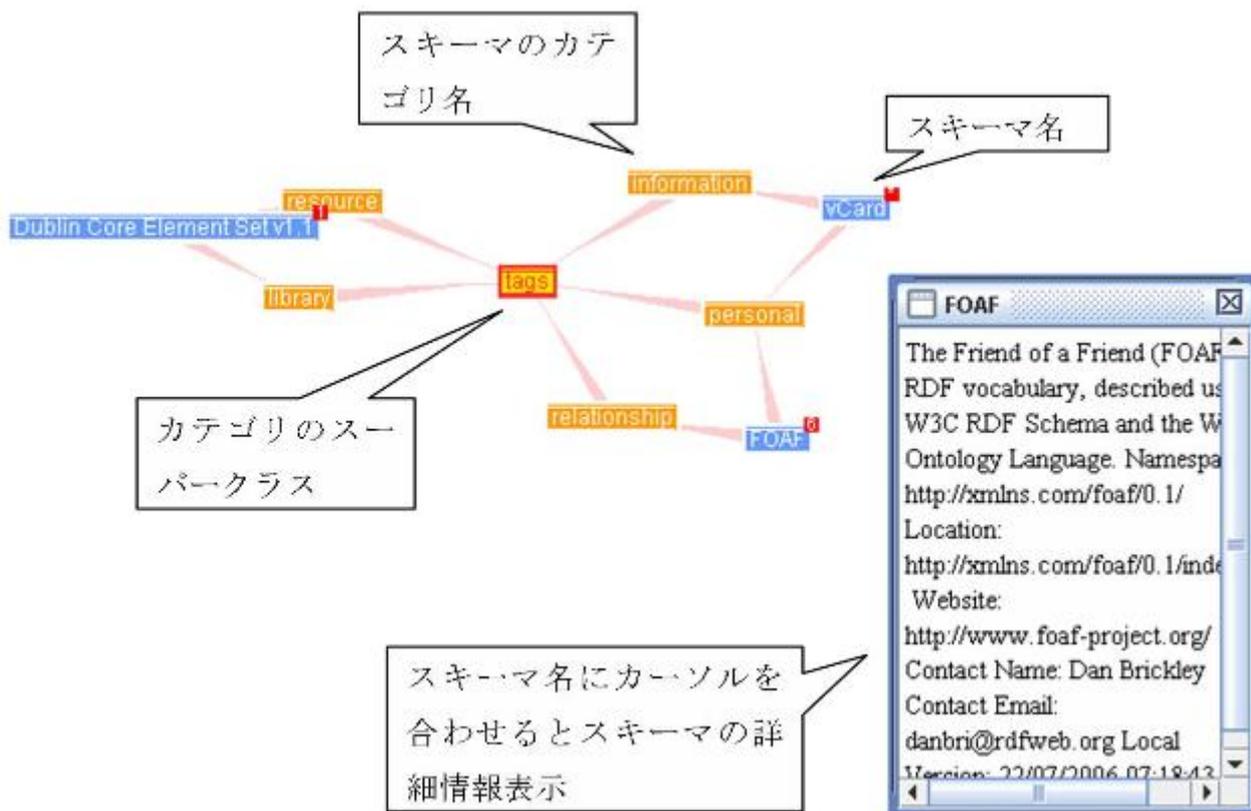


図 4 スキーマとカテゴリを繋ぐグラフ表示 (TouchGraphインタフェース)

The best way to search, design and share Application Profiles.

Home | Browse | Element Cluster | Schema Tag | Edit

PropertyUsage: 利用するエレメント http://xmlns.com/foaf/0.1/firstName について記述する項目です。

namespace:

name:

利用するエレメントのuri:

ラベル:

comment:

description:

obligation:

condition:

maxOccurs:

PropertyUsage: 利用するエレメント http://xmlns.com/foaf/0.1/givenname について記述する項目です。

namespace:

name:

利用するエレメントのuri:

ラベル:

comment:

description:

obligation:

condition:

maxOccurs:

PropertyUsage: 利用するエレメントについて記述する項目です。

namespace:

name:

利用するエレメントのuri:

図 5 エレメント制約条件の対話的付与

eエレメントの名前/説明:name
クラスター:n

[MATRIX | RDF | HANAUI Graph | Touch Graph]

Element List

[+] **title** s: Dublin Core Element Set v1.1 c: name
label: Title [RDF]
comment: A name given to the resource.
description: Typically, a Title will be a name by which the resource is formally known.

[+] **hasVersion** s: Dublin Core Element Refinements and Encoding Schema c: relation
label: Has Version [RDF]
comment: The described resource has a version, edition, or adaptation, namely, the referenced resource.
subPropertyOf: http://purl.org/dc/elements/1.1/relation

エレメントをドラッグアンドドロップ

ここに利用するエレメントの情報を一時的に保存します。

[+] nick s: FOAF create
label: エレメントの横に表示されている[+]をクリック、またはエレメントをドラッグアンドドロップすることでここに追加することができます。IRC and other chat nicknames.

ゴミ箱

エレメントが揃ったら Editに進んでください。

図 6 利用したいエレメント情報の一時保存

```

<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dcap="http://purl.org/ws/nnmi/dcterms/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/">
  <dcap:SchemaDocument rdf:about="http://test.com/sample_addressbook">
    <dc:title>サンプルアドレス帳</dc:title>
    <dc:description>住所、氏名、情報の更新日を項目として持つアドレス帳のサンプルスキーマ</dc:description>
    <dc:terms:modified>
      <dc:terms:W3CDTF>
        <rdf:value>2007-01-28</rdf:value>
      </dc:terms:W3CDTF>
    </dc:terms:modified>
    <dc:publisher rdf:resource="http://test.com"/>
  </dcap:SchemaDocument>

  <dcap:Agency rdf:about="http://test.com">
    <dc:title>tests</dc:title>
    <dc:description>テスト用ユーザ</dc:description>
  </dcap:Agency>

  <dcap:AppProfile rdf:about="http://test.com/sample_addressbook">
    <dc:title>サンプルアドレス帳</dc:title>
    <dc:description>住所、氏名、情報の更新日を項目として持つアドレス帳のサンプルスキーマ</dc:description>
    <dc:publisher rdf:resource="http://test.com"/>
    <dc:rdfs:isDefinedBy rdf:resource="http://test.com/sample_addressbook"/>
  </dcap:AppProfile>

  <dcap:PropertyUsage rdf:about="http://test.com/sample_addressbook/address">
    <dcap:uses rdf:resource="http://www.w3.org/2001/vcard/rdf3.0#ADR"/>
    <dc:rdfs:label>Address</dc:rdfs:label>
    <dc:description>住所を記述する。</dc:description>
    <dcap:obligation rdf:resource="http://purl.org/ws/nnmi/dcterms/Obligation/optional"/>
    <dcap:maxOccurs>unbounded</dcap:maxOccurs>
    <dcap:isMemberOf rdf:resource="http://test.com/sample_addressbook"/>
  </dcap:PropertyUsage>

  <dcap:PropertyUsage rdf:about="http://test.com/sample_addressbook/family">
    <dcap:uses rdf:resource="http://www.w3.org/2001/vcard/rdf3.0#Family"/>
    <dc:rdfs:label>Family Name</dc:rdfs:label>
    <dc:description>苗字を記述する</dc:description>
    <dcap:obligation rdf:resource="http://purl.org/ws/nnmi/dcterms/Obligation/optional"/>
    <dcap:maxOccurs>unbounded</dcap:maxOccurs>
    <dcap:isMemberOf rdf:resource="http://test.com/sample_addressbook"/>
  </dcap:PropertyUsage>

  <dcap:PropertyUsage rdf:about="http://test.com/sample_addressbook/given">
    <dcap:uses rdf:resource="http://www.w3.org/2001/vcard/rdf3.0#Given"/>
    <dc:rdfs:label>Given Name</dc:rdfs:label>
    <dc:description>名を記述する</dc:description>
  </dcap:PropertyUsage>

```

図 7 作成したメタデータスキーマ(アプリケーションプロファイル)の XML/RDF形式出力