

WWW10 調査報告書 (別紙)

2001年5月21日

セマンティック WEB タスクフォース

清水委員長(NEC)、袴田委員(富士通)

目次

1	TF4 Metadata.....	1
2	Machine learning for information extraction from XML-marked up text on the Semantic Web.....	10
3	An RDF Framework for Resource Discovery	10
4	The RDFSuite: Managing Voluminous RDF Description Bases.....	12
5	Semantic Web Languages: RDF vs. SOAP Serialization	13
6	Jena: Implementing the RDF Model and Syntax Specification	14
7	Conceptual Open Hypermedia = The Semantic Web?.....	16
8	A comprehensive framework for learning ontologies for the Semantic Web	18
9	Using RDF(S) to provide multiple views into a single ontology.....	21
1 0	Accessing Information and Services on the DAML-Enabled Web.....	22
1 1	RDF Models for Dynamic Syndication and Wireless Application.....	24
1 2	Mobilizing the Semantic Web with DAML-enabled Web Services.	26
1 3	The Design and Implementation of the Redland RDF Application Framework 27	
1 4	Combining RDF and XML Schemas to Enhance Interoperability Between Metadata Application Profiles.....	30
1 5	Enabling knowledge representation on the Web by Extending RDF Schema. 33	
1 6	Annotea: Shared Web Annotations for Building an Open RDF Infrastructure34	
1 7	RDFdb and Algae.....	36
1 8	Building a Semantic Web Site	36
1 9	RDF Expert.....	38
2 0	WebKB.....	39
2 1	Sesame	40

1 TF4 Metadata

発表者 : Stuart Weibel (Director of the Dublin Core Metadata Initiative)

(ア) 要点

➤ ウェブの状況

- 検索システムは、機能よりビジネスモデルが、先行している。
- インデックスのカバレッジは、予測できず、限られている。
- 同じものが多く、精度が低い。
- インデックススパムに犯されている。
- そのリソースは、変転が、激しい。
- バージョン、エディション、バック問題は、どうなるのか？
- アーカイピング（記録保管庫）にできない。
- 信用できるサービスやサービス品質の高いものは一部である。
- IPR の管理が、難しい。

➤ Metadata とは

データに対する構造化データである。

- カオスを解消する手助けになる。
- 自動発見操作を可能にする。

色々な局面を変えうる。

- スペシャリゼーション(特化)
- ディセントラリゼーション(非集中)
- デモクラティゼーション(平等化)

➤ インターネットは、色々なコミュニティを包含している。

➤ インターオペラビリティのためには次の規則が必要である。

意味(セマンティクス)

- 構成要素の意味

構造(ストラクチャー)

- ヒューマンリーダブル
- マシンパーサブル

構文(シンタックス)

- 意味と構造とを運搬するための文法

➤ Metadata を用いた成果はないのか？

MARC STANDARDS がある。

一連の MARC 標準は、世界で唯一の最も成功しているリソース記述標準である。

➤ MARC カタロギングとは

MARC-AACR2 は、本当にカタロギングできるのか？

- MARC は、通信形式(ストラクチャーとシンタックス)である。
- AACR2(Anglo-American Cataloging Rules)は、カタロギング規則を規定している。

- このモデルをウェブに適用するのは、何故まずいのか？
高価である。
 - 複雑すぎる
 - カタログ作業のプロが、必要である。
 図書目録に限定されている。
 - 変化のないリソース
 - リソースの増加とリソース間の関連付けの操作が、未完成である。
 Anglo-centric である。
 - MARC21 は、MARC レコードの 3/4 を考慮しているが、沢山あるその他の物について考慮していない。
- Metadata は色々なことに適用できる。
 - リソースディスカバリー
 - 文書管理
 - 権利管理
 - 内容格付け
 - セキュリティと監査
 - 記録状況
 - プロダクトとサービス
 - データベーススキーマ
 - プロセス制御もしくは記述
- Metadata において挑戦すべき事項
 - 色々なメタデータとの共存
 - テンション：機能と単純性
 - テンション：拡張性と相互互換性
 - 人間と機械での生成と利用
 - コミュニティ特有の機能、生成、管理、アクセス手段
- Warwick Framework: モジュール Metadata
 - Warwick Metadata Workshop(DC-2)で導入された Metadata の概念アーキテクチャ。
 - モジュール Metadata の仕様、収集、エンコーディング及び交換を可能にする為の概念アーキテクチャ
 - Metadata efforts(Dublin Core を含む)の為に裏方作業を行う。
 - 包括的要素集合のブラックホールを回避する。
 - パッケージレベルでの相互互換問題に焦点を当てる。
- モジュール化により、分散管理を実現する。
 - 熟練者(ソフトウェアベンダの事ではない)は、次に責任をもつこと、

- セマンティクスの定義
- 管理
- アクセス管理
- データの権威付け
- 共有と分配

(イ) Dublin Core に付いて

➤ Dublin Core の歴史

1994 年：ウェブ記述の為の簡単なタグを開発

1995 年：Dublin Core は、必要とされた多くの語彙（"Warwick Framework"）の一つ。

1996 年：テキストとイメージの記述の為、13 要素を 15 要素に拡張

1997 年：WF は、リソースディスクリプションフレームワーク(RDF)の為、形式記述を必要とする。

2000 年：Dublin Core Metadata Initiative は、修飾子の推奨を行い、その組織の視野を core 以外にも拡張

➤ Dublin Core Workshop Series

- US DC1,3.6
- UK DC2
- Australia DC4
- Finland DC5
- Germany DC7
- Canada DC8
- Tokyo(2001) 10月22日～26日

➤ デジタルツーリストのピジン言語

Metadata は言語である。

Dublin Core は、諸々の境界を越えてリソースを見つける為の単語の数が少いで簡単な言語 - ピジン言語 -

異なる言葉を話す人は、自然に、コミュニケーションにピジン化された言葉を使う。

我々は、全員、インターネット上のツーリストである。

➤ Dublin Core の文法

<http://www.dlib.org/dlib/october00/baker/10baker.html>

各人の母国語のように難解なものではなく、覚えるのが簡単で、実用的になるよう設計されている。

ピジン言語：少ない語彙（Dublin Core：15 の特別な名詞とたくさんの形容詞）

単純な文法：文(ステートメント)は、単純かつ固定の形式

切なものであるべきであり、そうならば、リソース検索に有用となる。

➤ 修飾詞の”サルでも分かる原則(Dumb-Down Principle)”

15 の要素は、修飾詞の有無に拘わらず、理解でき、且つ、有用である。

修飾詞は、意味を明確にする。(しかし、理解が難しいものであってはならない。)

名詞は修飾語句なしで、意味をなさなくてはならない。

仮に処理ソフトが、不明な修飾詞に出会ったら、その意味を捜すか、又は、単純に無視すべし。

➤ 新たな用語のプロポーザルの平等なレビュー

DCMI は、新たな DCMI 用語(修飾詞および要素)のプロポーザルのレビューをボードで行う。

文法原則を考慮してプロポーザルを評価する。(修飾詞は無視可能か否か?)

(仮)承認状態の結合モデル：提案、確認、推奨、廃棄

最初の修飾詞は、2000 年の 7 月に推奨された。

<http://pur.org/DC/documents/rec/dcmes-qualifiers-20000711.htm>

➤ Dublin Core に対する公開質問

- 15 属性の適切な値とは何ですか？それらは、どのように境界を超越した検索に使えるのですか？

- DCMI は、どのように Dublin Core 実用化をするのですか？

- より複雑な Metadata でリソース記述をするとき、AP は、ビジン言語である DC をどのように使う事ができるのですか？

➤ 検索桶対記述

曖昧検索桶として DC 要素を想定した場合、

- 異なる桶の為には、それにそれぞれに適切な異なるデータの型が必要：URL、日付文字列、用語文字列、名前

- 三島由紀夫について書いた本と、三島由紀夫が書いた本の分離

検索桶：リソース検索のため

しかし、一般に、曖昧分類は、リソース記述に充分でない。

- 検索後、画面上に更に詳細な説明の表示

➤ DCMI は、そのミッションを拡大(2000 年 10 月)

DCMI のミッションは、次の活動を通じて、インターネットを用いたリソース検索を簡単にする。

- 境界を超越した検索の為の Metadata 標準を開発する。(例：Dublin Core)

- Metadata 集合の互換性のため枠組みを定義する。

- コミュニティの発展を促す、または、規範となる特別な Metadata 集合の開発を促す。

➤ DCMI の組織の見直し

ミッションの拡大

- エージェント(もしくは、事象)の為の中核 Metadata 要素
- 多くの領域固有の Metadata を統合するための枠組み

組織見直しモデル

- W3C または Unicode コンソーシアムのような組織の会員
- 合意ドリブンモデルの維持
- 国際的視野
- より良い教育、文書化、普及活動

➤ DCMI 活動

- 標準の開発と保守
- Metadata 保管と基盤
- 技術作業グループと定期的なワークショップ
- 教育資料と利用者の手引き
- 教育と訓練
- オープンソースソフトウェア
- 他の標準または、利用者コミュニティとの連携

➤ Dublin Core の公的認知

CEN ワークショップ合意

- Dublin Core の要素を CWA13874 として保証
- 欧州業界に利用ガイドラインとして提供

NISO Z39.85

- National Information Standards Organization

ANSI 会員

- 現在投票中

➤ DC-2001

東京で DC-2001 を開催

- 10月22日～26日

3本のトラックがある。

- 技術作業グループ会議
- 実装報告と研究発表
- 一般の人に対する紹介と講習

➤ Dublin Core の背景

- もしも、Dublin Core が、DCMI の中核ならば、それ以外のものとは何か？
- もしも、Dublin Core が単純なピジン言語ならば、Metadata 言語の広い展望とは何か？
- ピジン言語がどのように、より複雑なモデルや”アプリケーションプロファイ

ル”に関連するのか？

➤ 他の語彙と一緒に DC を利用する場合

特定アプリケーションプロファイル(政府情報、教育、数学)は、以下のものを必要とするかも知れない。

- Dublin Core 要素の General-purpose 利用
- より領域固有の標準で作られた要素の利用
- 特定の局所的利用の為に DC 要素の狭い標準定義
- 既存標準の範囲外にあるローカルな要素の新設

➤ 名前空間とアプリケーションプロファイル

名前空間は、用語と定義とを宣言する。

- Dublin Core 名前空間=Dublin Core 標準

アプリケーションプロファイルは名前空間から用語を再利用する。

- たくさんの名前空間から用語を取り込むかも知れない。
- 局所的な目的のため定義を付加するかもしれない。
- 総べての用語は、名前空間の中に定義されねばならない。
- 局所的に定義された名前空間があるかもしれない。

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/22-rdf-syntax-ns#"
  xmlns:dc="http://pur.org/dc/elements/1.1/"
  xmlns:sy="http://pur.org/rss/1.0/modules/syndication/"
  xmlns:co="http://pur.org/rss/1.0/modules/company">
```

```
<dc:publisher> The O'Reilly Network </dc:publisher>
<dc:creator> Rael Dormfest(mailto:rael@oreilly.com) </dc:creator>
<dc:rights> Copyright &#169. 2000 O'Reilly & amp,associates. Inc </dc:rights>
<dc:date> 2000-01-01T12:00+00:00 </dc:date>
<dc:subject> XML </dc:subject>
```

```
<sy:updatePeriod> hourly </sy:updatePeriod>
<sy:updateFrequency> 2 </sy:updateFrequency>
```

```
<co:name> XML.com </co:name>
<co:market> NASDAQ </co:market>
<co:symbol> XML </co:symbol>
```

```
</rdf:RDF>
```

図 2 多重名前空間の例

➤ 局所的利用の為に標準定義の付加

Dublin Core 名前空間の場合

- DC:Title – ある要素のマシンリーダブル名
- "Title : リソースに与えられた名前" – ヒューマンリーダブルな名前と定義

コレクション記述プロファイル(UKOLN)の場合

- DC:Title – DC 名前空間と同一
- "Title:コレクションに与えられた名前

定義はアプリケーションの状況により変化する。

➤ 局所的利用の為の DC:Title 付加の例

正式な Dublin Core 名前空間の定義では、

- "Title: リソースに与えられた名前"

英国の"アプリケーションプロファイル"の定義では、

- "Title:コレクションに与えられた名前"

この場合、定義が狭い。

➤ プロファイルは、多重実体のモデルの場合もある。

次の情報を持つ実体としてのリソース(物)

- 主題(dc:title)
- 作成日(dc:date dcq:created)
- 識別子(dc:identifier)

次の情報を持つエージェント(一人の人)

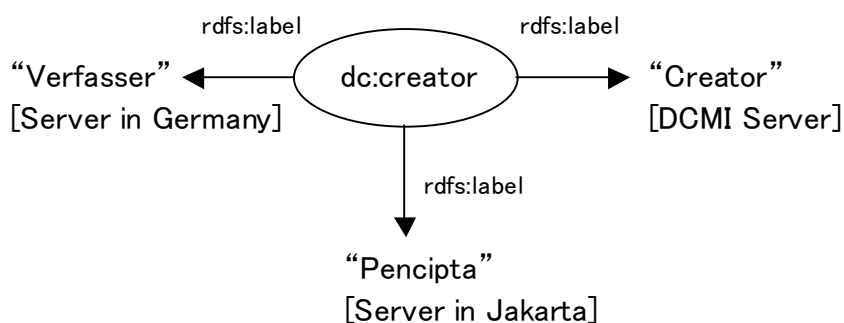
- 名前(vcard:fn)
- 誕生日(vcard:bday)
- 識別子(dc:identifier)

➤ 名前空間の翻訳

Dublin Core は、26ヶ国語に翻訳されている。

- マシンリーダブルトークンは全部で共有されている。
- ヒューマンリーダブルラベルは、各国毎に定義されている。
- 翻訳は、たくさんの国々で配布され、保守されている。
- 更に、DCMI レジストリにリンクされている。

➤ 1つのトークン 多国語の中のラベル



- RDF より強力な文章パターン

Dublin Core ステートメント

- リソースの作成者は"Tom Baker"である。
- リソースは識別子として `http://foo.org/bar.html` を持つ。

RDF の 3 元(triples)は、同じ事を言うにも、より強力な方法である。

- `http://foo.org/bar.htm` の作成者は、"Tom Baker"である。

2 Machine learning for information extraction from XML-marked up text on the Semantic Web

セッション名 : WF5 Semantic Web

発表者 : Nigel Collier(一橋大学)

XML マークアップされたテキストからの低レベルな情報抽出についての説明。隠れマルコフモデルをベースとしている。具体的には、すでにマークアップされているサンプルデータ(分子科学分野およびニュース記事)の一部を学習させた後に、残りのテキストに対して単語の分類を行わせ、その結果と問題点について述べた。

(ア) ポイント

- 大量の既存文書に対するメタ情報の生成をどのように行うかが課題になっている。あるドメインに限って文書群の一部をもとに学習を行かせた機械に、残りの文書を自動的にマークアップさせることで、効率のよいメタ情報の生成を可能にする
- 従来の辞書ベースによる解析では困難な、まだ辞書が存在していないようなドメインの文書群に対するアプローチ。
- 隠れマルコフモデルを用いた語彙の機械学習は、かなり高いパフォーマンスを見せたが、モデルにはいくつかの本質的な問題があり、より効率の良い学習にはまったく異なる機械学習のテクニックが必要になると思われる
- さらに、RDF や RDF スキーマで記述されるオントロジーの知識を統合するように、DTD 内に記述される、要素の種類や要素の属性を利用する必要がある。

3 An RDF Framework for Resource Discovery

セッション名 : WF5 Semantic Web

発表者 : Franklin Reynolds(Nokia)

各種の検索サービス (DHCP, SDP, SSDP, SLP, LDAP,...) 間におけるプロトコルの相違を、プロトコル独立な RDF ベースのフレームワークによって吸収し、共通のメタデータツールキット、検索言語によるプロトコル間の相互運用性の確保を狙う。

(ア) RDF を使う理由

- 様々な検索プロトコル(DHCP, DNS, X.500, LDAP, SLP, SSDP, SDP, Jini, Google)が存在しているが、データのフォーマットや検索言語などの再利用はされていない
- データベースの世界では、SQL を使って、実装と検索言語を分離している。
- RDF は W3C が採用している、柔軟度の高いデータモデルをベースにしたメタデータ言語であり、プロトコル固有のボキャブラリを自由に定めることができる。
- 各プロトコルを RDF 化することで、RDF に対応したツール(API, 検索言語)が、プロトコルに共通に利用できるようになる。

(イ) API と検索言語

現在、RDF には標準的な API や検索言語は存在していない。そこで、以下のものを実装した

- RDF API : RDF のデータモデル操作を行うための簡易 RDF API。Java で実装。RDF データの解析には SiRPAC の RDF パーサを利用。
- DGQL : Directed Graph Query Language の略。RDF グラフのマッチングを目的とした簡易検索言語(図 3)。スキーマ情報に基づいた検索などはできない。マッチングに成功した部分グラフを返す。

```
(N car (AND (A make (OR (N coupe)
                    (N wagon))
              (A owner (N Reynolds))))
```

「クーペあるいはワゴンで、Reynoldsが所有者である車」
(N: ノード, A: 述語)

図 3 DGQL の問い合わせサンプル

(ウ) 評価と課題

- 実装は、SLP に対する DGQL 拡張と、Web ベースの検索サービスとが行われた。
- 2つのプロトコル(SLP と Web ベースの検索サービス)は、同じメタデータを利用しており、これによって2つのプロトコル間でサービスの記述に矛盾が発生することを防げる。
- RDF には標準的な API が存在しておらず、各 RDF パーサの動作が異なっており、パーサに非依存の RDF アプリケーションを記述するのは非常に困難である
- RDF では、整数や日付などのプリミティブな型のデータをサポートしていな

い。そのため、検索言語 DGQL では、数の大小比較などの有益と思われる機能が実装されていない。しかし、型や RDF スキーマの導入は、システムを非常に複雑にしてしまう。

- プロトコルの応答モデルの違いが重要な場合がある（同期 / 非同期の通信、単一 / 複数の応答）

4 The RDFSuite: Managing Voluminous RDF Description Bases

セッション名：WF5 Semantic Web

発表者：Sofia Alexaki（ギリシャ）

大量の RDF データを扱うことを目的とする、ICS-FORTH RDFSuite というツールセットの紹介。RDFSuite には、RDF データの検証(VRP)、格納(RSSDB)、検索(RQL)といった機能が含まれている（図 4）。

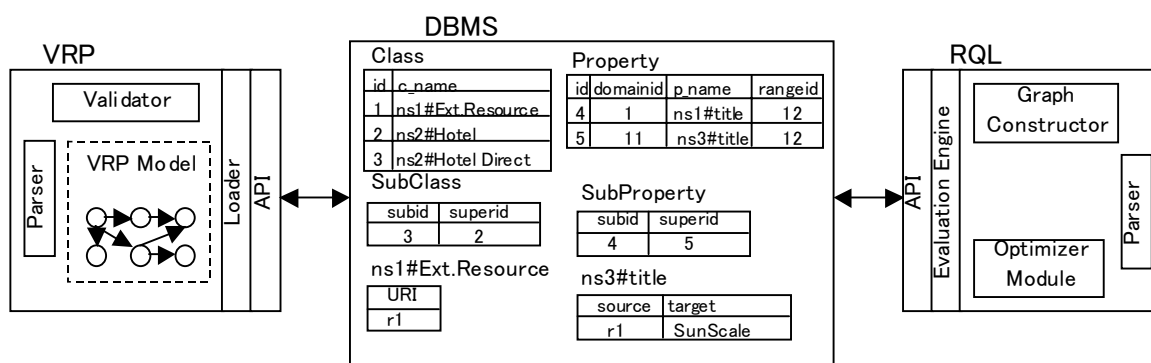


図 4 ICS FORTH RDFSuite の全体図

(ア) 背景

- いくつかのコンテンツプロバイダ(Yahoo!や Open Directory)は、そのコンテンツ管理に RDF を利用しており、そのデータ量は巨大なものになっている。
- Open Directory の場合、337085 個のトピックと、その下に分類される 2342978 ページの Web サイトの情報を管理しており、分類情報として 170M バイト、リソース記述に 700M バイトのデータを保持している。

(イ) The Validating RDF Parser(VRP)

RDF スキーマに記述された制約条件に基づいた検証が可能な、RDF パーサ。

(ウ) RSSDB

- RDF データを格納するデータベース。
- 特徴として、スキーマ情報をもとに、RDF メタデータにおける、オブジェクトの関連情報を自動的に生成する機能をもつ。
- これにより、スーパークラスや、サブクラス情報を利用したデータの格納が可能になり、クラス階層にもとづいたトラバースが効率よく行える。

(工) RQL

- SQL 風の検索言語
- クラス階層情報に基づいた、宣言的な検索が可能
- 命令の解釈時に、オブティマイザーによる最適化を行う

(オ) 課題

- RQL の最適化手法の改善
- データベースのトランザクション処理の実現
- スキーマ情報のより効率的な格納方法の開発

5 Semantic Web Languages: RDF vs. SOAP Serialization

セッション名 : WF5 Semantic Web

発表者 : StefanHaustein (Dortmund 大学 / ドイツ)

RDF はマシンリーダブルな言語であり、アプリケーション間で交換されるデータフォーマットである。データ交換用という観点から、同じような仕様として SOAP メッセージが挙げられる。この2つを比較して SOAP の利点を明らかにすることで、SOAP 上に Semantic Web のある部分を構築することを提案する

(ア) RDF の文法上の問題点

- XSLT との互換性 : 文法上のバリエーションが豊富なため、XSLT との相性が悪い
- 読みやすさ : 文法上のバリエーションに精通している必要がある。

(イ) RDF のデータモデル上の問題点

- オブジェクト指向システムとの互換性 :
プロパティが、ユニークでなくてはいけない ローカルなプロパティの存在を許していない

(ウ) SOAP の長所と短所

- オブジェクト指向システムとの親和性が高い
- 各種の W3C 標準技術(XSLT, XML Schema,...)との親和性が高い
- SOAP には、オブジェクトに URL を割り当てる仕組みがない

(エ) SOAP と RDF

- SOAP と RDF は、両方ともロジックレベルの表現力を持たない SOAP 上に推論システムを作るのは、RDF よりも難しいというわけではないように見える
- SOAP は、Web ページの注釈という観点からみると、ふさわしくない
- SOAP と RDF の両方を統合して、その上に論理層を構築すべきだと思われる

6 Jena: Implementing the RDF Model and Syntax Specification

セッション名 : WF5 Semantic Web

発表者 : Brian McBride (Hewlett Packard 研究所)

Java で実装された RDF の API である Jena の実装作業を通じて得られた RDF 仕様の躰きやすい点の説明と、Jena の紹介。

(ア) リソースと URI

RFC2396(URI)では、「リソース」は「概念」と対応するものとしている。RDF では、リソースは1つの URI で識別可能であると定義している。

- リソースは2つ以上の URI をもてない？ あるリソースとあるリソースが実は同じリソースであることを示す手段が、RDF では未定義である。
- 匿名リソースはありえない？ リソースの「表現」は何通りも存在しているが、これはリソースそのものを表わしているわけではない。「表現」には URI は必要ない。
- Jena では、匿名リソースには URI を割り当てない それでうまくいっている

(イ) プロパティ

- プロパティは、その属する名前空間 URI とローカル名を連結したもので識別される URI だけが与えられた状態では、プロパティのローカル名を決定することはできない。

(ウ) リテラル

- リテラルの言語は、リテラルの一部である リテラルは、単なる文字列ではない

(エ) ステートメントの具体化

- RDF のステートメント = リソースではない。
- RDF のステートメントを表現するリソースが存在しており、このメカニズムを具体化(Reification)と呼ぶ (図 5) このとき、このリソースを、「具体化されたステートメント」と呼ぶ。

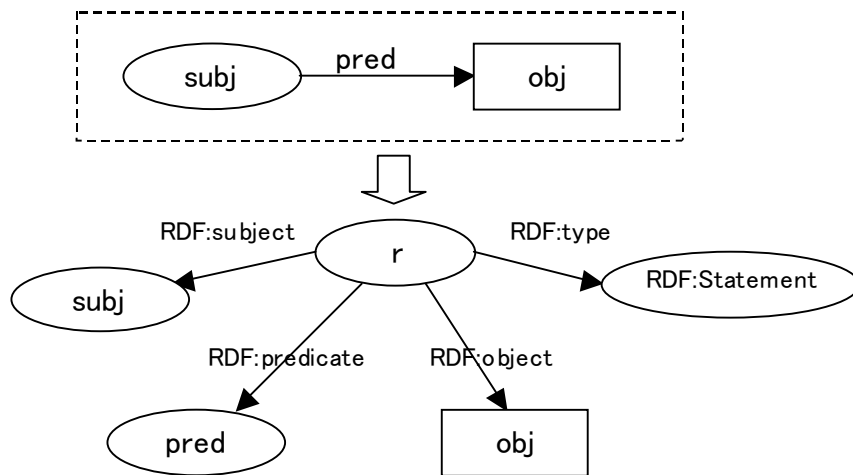


図 5 あるステートメントを「具体化」する

- あるステートメントに対応する具体化されたリソースは、無数に存在可能である。しかし、その個々のプロパティが指す先は、プロパティが同じなら、同じものを指す。

(オ) ステートメント、statements と出現

ステートメントの「出現」を表現したい場合がある。ステートメントの出現を *statements* と呼ぶ。

- 具体化されたステートメントによって、ステートメントの「出現」を表現することはできない。

(カ) RDF グラフ

RDF 仕様は、RDF グラフを形式的に表現する方法について述べていない

- RDF グラフは、形式的には具体化されたステートメントの集合として表わすことができる
- これによって、RDF グラフのマージが容易になる。

(キ) Jena

Jena の目標は、以下の 2 つである

- 使いやすい API であること
- RDF の仕様に適合すること

Jena の API は、以下の URL で入手可能

<http://www-uk.hpl.hp.com/people/bwm/rdf/jena>

(ク) Jena の実装アーキテクチャ

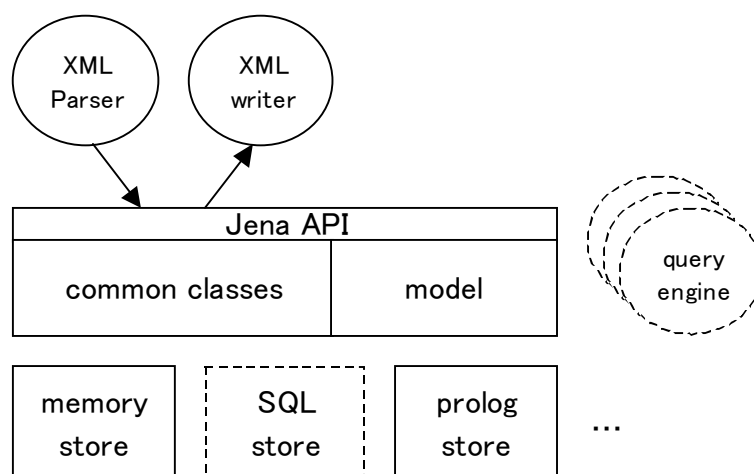


図 6 Jena の実装アーキテクチャ

7 Conceptual Open Hypermedia = The Semantic Web?

セッション名 : WF5 Semantic Web

発表者 : Carole Goble(Manchester 大学 / イギリス)

オントロジーとメタデータによる、概念情報にもとづいた動的なリンク生成システム COHSE (Conceptual Open Hypermedia ServiceE)の紹介。

(ア) COHSE で使われているテクノロジー

- ウェブベースのオープン・ハイパーメディア・リンクサービス :
コンテンツ中にリンク情報を埋め込むのではなく、コンテンツとリンク情報を切り離し、リンク情報はその種類ごとにリンクの生成・解決を行うエンジンによって提供されるようにする。これによって、いわばユーザ定義のリンク情報をコンテンツに重ね合わせることを可能にする。COHSE では、DLRS(Distributed Link Resolution Service)と呼ばれるシステムによって、複数のサーバ間にまたがってリンク解決エンジンが存在することを可能にしている。
- 概念ハイパーメディアシステム :
ドキュメント間のリンク情報を直接手で編集するのは、一貫性を保つのが難しく、エラーの原因となりやすい。概念ハイパーメディアシステムとは、コンテンツのメタ情報をもとにコンテンツを概念ドメインに分類し、概念ドメイン間のリンク情報を編集することで、一貫性のあるリンク情報を生成するシステムである。
- オントロジーによる推論サービス :

コンテンツ中の用語や、コンテンツ間の関係についての概念モデルを表現するために、オントロジーによる推論サービスを利用する。例えば、「このページはブードルについてのページです」というコンテンツがあったとき、「ブードル」という単語から、オントロジーによって、このページを「犬」に関するページとして分類するということが考えられる。

(イ) CHOSE のアーキテクチャ

- 全体図を図 7に示す（動作例を図 8に示す）
- メタデータサービス：文書に関連するメタデータを管理する。メタデータの外付けのリンク情報は、リンク・ベースに格納されている
- オントロジーサービス：オントロジーを利用して、コンテンツ中の用語の上位概念・下位概念を取得する。オントロジーには、OIL あるいは DAML+OIL ベースのものが使われる。
- リソースサービス：概念をリソース（URL）にマッピングする
- 編集知識：アプリケーションのフィルタ情報や、ユーザのプロファイル情報によって、リンクの生成をコントロールする
- リンク生成器：文書に付加すべきリンク情報を生成する。
- DLRS：DLRS クライアントエージェント。受け取った文書を解析して、リンク付きの文書を生成して返す。実装では、Java アプレット / Java スクリプトの形でブラウザに統合される。

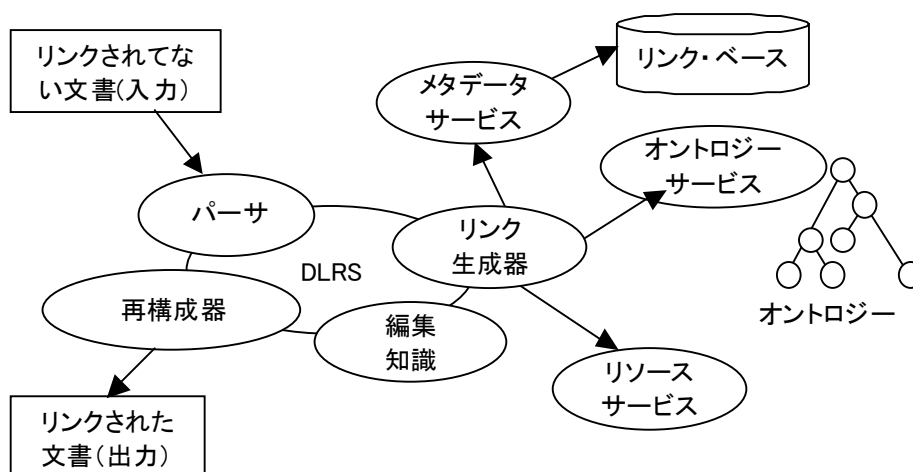


図 7 CHOSE のアーキテクチャ

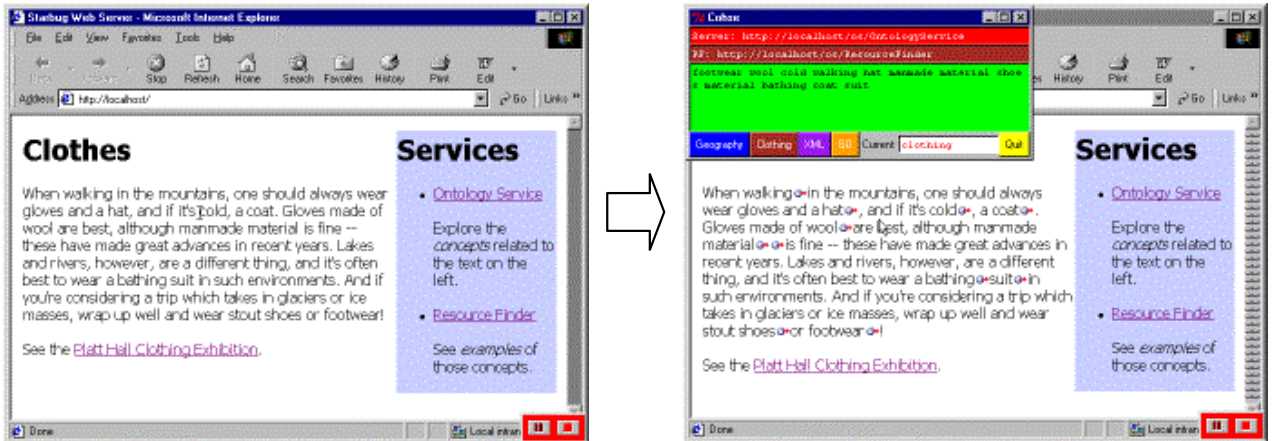


図 8 CHOSE の動作例

(民族衣装のサイトにおいて、服に関するオントロジーを利用してリンクを生成する例)

(ウ) 問題点

- 概念のレベルによっては、リンクの数が増えすぎてしまう。一度に表示するリンク数はどれくらいが望ましいのか？(潜在的なリンクはどのようにみせるべきか)
- 推論サービスは、どのように利用するのがベストなのか？
- 過去のコンテンツを考慮する場合は、どのように行うべきか？
- コンテンツの「途中」にリンクするのは、混乱の原因になるのではないか？

8 A comprehensive framework for learning ontologies for the Semantic Web

セッション名：WF5 Semantic Web

発表者：Alexander Maedche (Karlsruhe 大学 / ドイツ)

Semantic Web の成功のためには多くのオントロジーが必要であり、そのためにはオントロジーが簡単に作れるようにならなくてはならない。そこで、オントロジー作成環境 Text-To-Onto を提唱する。一般的なフレームワークやアーキテクチャに加えて、オントロジー作成の模範的なテクニックについても説明する。同様のものとして、Protégé (知識ベースシステムのためのモデリング環境) がある。

(ア) オントロジー構築のサイクル

新たなオントロジーを作成する手順は以下のようになる。

1. IMPORT/REUSE :
既存のオントロジーを、構造をそのまま、あるいは、新しい構造と古い構造のマッピングを定義した上で、新しいオントロジーに取り込む。
2. EXTRACT:
ツールの助けを借りて、ウェブドキュメントなどの入力データから目的のオントロジーを構築する
3. PRUNE:
オントロジーの目的にあわせて、作成しているオントロジーの内容を削って、オントロジーに含まれる情報の量に過不足がないようにする。
4. REFINE:
オントロジーがうまく働くように、細かな調整を加える。
5. APPLY:
アプリケーションから、作成したオントロジーを利用して、作成したオントロジーの評価を行う
6. 1に戻る

(イ) オントロジー作成のためのコンポーネント

Text-To-Onto (図 9) の各コンポーネントは以下のようにになっている

- 管理コンポーネント：入力データ(HTML, XML, DTD, データベース, オントロジー)の選択や、利用するリソース処理のメソッドおよびアルゴリズムの選択のために利用される
- リソース処理コンポーネント：リソース形式(構造化文書・非構造化文書)にあわせて、処理しやすいように変換を行う。
- 自然言語処理システム：SMES(Saarbrücken Message Extraction System)によって、自然言語テキストのトークン化などの処理を行う。SMES はドイツ語用の自然言語処理プロセッサ。
- アルゴリズムライブラリ：入力データの種類に応じて、適切な処理のアルゴリズムが利用できるようにする。通常、いくつかのアルゴリズムの結果を組み合わせる。
- OntoEdit：処理結果を GUI によってチェック・修正する(図 10)。

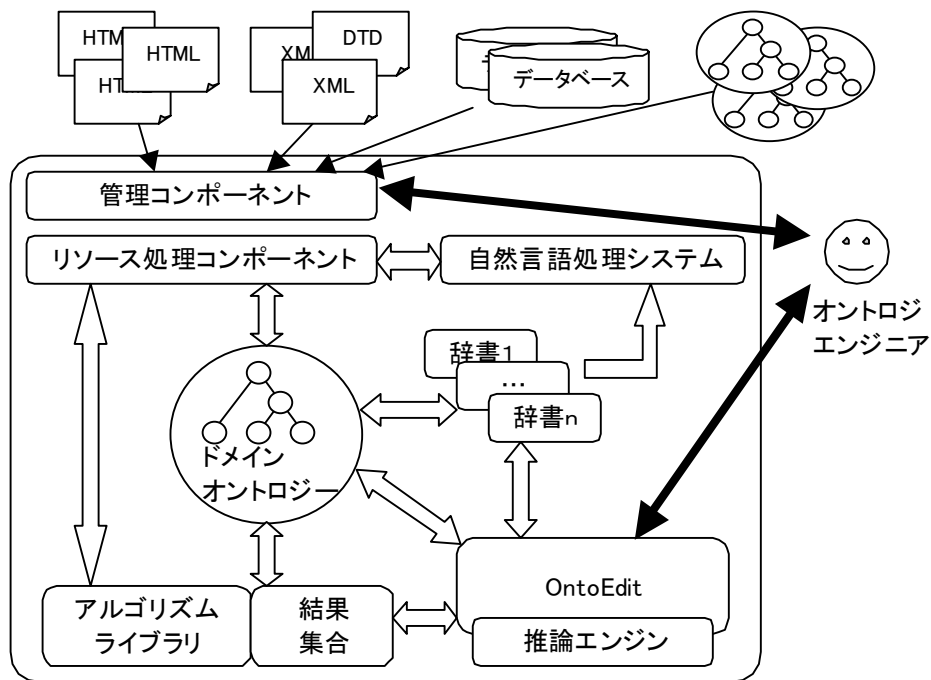


図 9 オントロジー作成環境 (Text-To-Onto) のアーキテクチャ

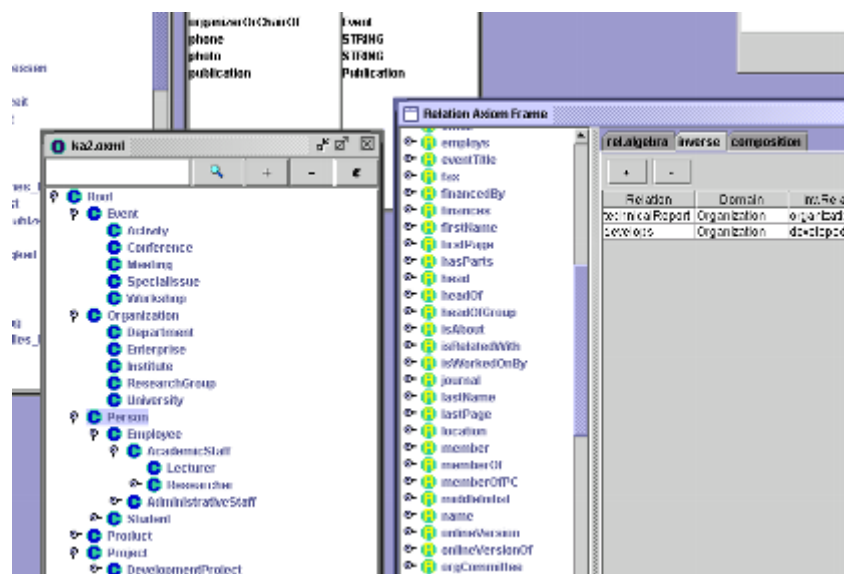


図 10 OntoEdit の実行画面

(ウ) 課題

今後、語彙 (オントロジー) の層の上に意味情報の層の作成が行われるようになれば、それに対応した (意味情報との矛盾を起こさないような) 新しいオントロジー作成技術が必要になってくると思われる。

9 Using RDF(S) to provide multiple views into a single ontology

セッション名：WF5 Semantic Web

発表者：Santtu Toivonen(Sonera / フィンランド)

RDF スキーマと、RDF を使って、一つのオントロジーから、文脈やケース固有の「ビュー」を生成することについての説明。これによって、ドメイン固有の事情をオントロジーから切り離すことが可能になり、分離された基本バックボーンとしてのオントロジーが実現できるというもの。

(ア) 背景

- RDF および RDF スキーマ(以下 RDF(S))では、セマンティックなカテゴリ全体を網羅するような用途は意図されていない。
- むしろ、RDF(S)では、ドメイン固有の事情に基づいたデータ(スキーマ)を提供することが意図されている。
- RDF(S)では、ある概念が定義される場合、クラスとして定義される(ある概念を定義するために使われる)か、属性として定義される(ある概念とは別の概念を説明するために使われる)かの、二者択一になる。(あるいは両方を別々に定義する必要がある)
- そのため、ある概念に対するスキーマがドメインにおけるビューによって異なってくる

これは、ある概念に対するドメインの「立場」の違いから生じている。

(イ) RDF スキーマとオントロジーの分離

- 2つのスキーマが、同じ概念を参照していたら、同じ概念を参照していることを識別できることが望ましい 同じオントロジーを参照する
- 単一の、巨大な、共有可能なオントロジーは、複雑で、動作が遅い。
- 巨大なオントロジーを使うことによるパフォーマンスの低下の影響は、RDF スキーマの使用によって(使用する概念の範囲を狭めることで)打ち消すことが可能になる
- 共通の概念を定義するものとしてのオントロジーと、ドメイン固有事情を現すスキーマを切り離すことが必要になってくる(図 11)。
- このことは、任意の RDF(S) データの集合から、オントロジーを構築することは不可能だと言っている? ある概念のオントロジー中の「位置」を決定できれば可能だが、どうすればそれが可能なのかは不明。逆の、オントロジーからあらゆる RDF(S) を生成することについては、ある概念に対する「立場」の情報がなければ、RDF(S)の仕様から考えると不可能。

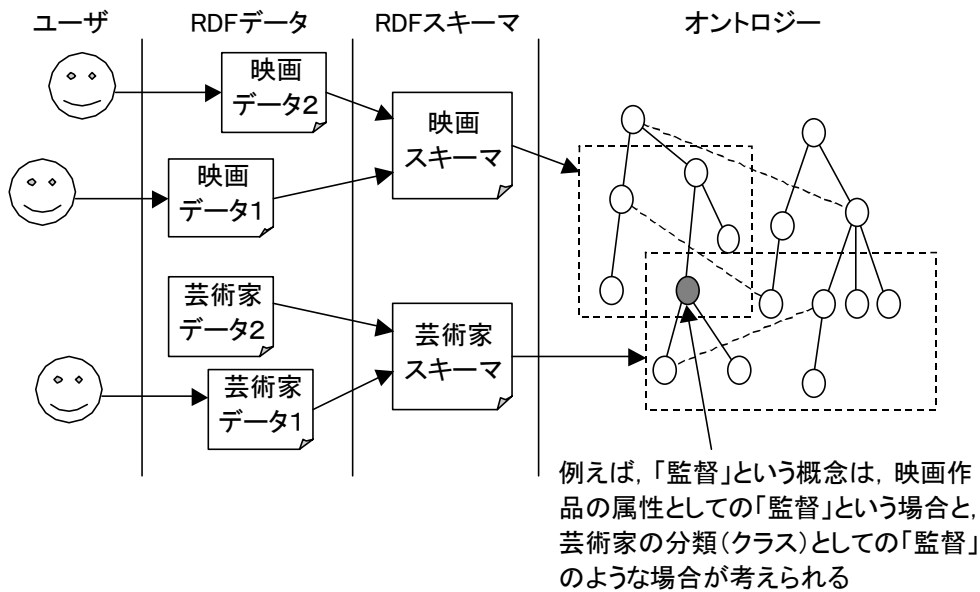


図 11 オントロジーとドメイン固有のスキーマを分離する例

1 0 Accessing Information and Services on the DAML-Enabled Web

セッション名：WF5 Semantic Web

発表者：Grit Denker (SRI / アメリカ)

DAML による検索とウェブサービスのアーキテクチャについての説明。

(ア) DAML とは

- エージェント記述のための仕様。アメリカ DARPA による、知的エージェント開発を目的としたプロジェクトにおいて開発されている。開発は 2000 年から開始 (正式な kickoff は 2000 年 8 月)、70MS / 5 年の規模。
- 階層的なアプローチを取っている (「RDF(S)」 「DAML+OIL」 「(DAML-L)」 「DAML-S」)(図 15)
- DAML+OIL：XML および RDF(S)をベースにしたマークアップ言語で、RDF(S)にはなかった制約条件を記述することが可能。(and, or, integer, string, equivalentTo, disjointWith などの制約条件を記述のためのタグが存在している。)
- DAML+OIL の仕様が、W3C に提案される予定。
- DAML-S：DAML ベースのウェブサービスオントロジーで、Web サービスの内容を明確に (機械が理解できる形で) 記述することを可能にするマークアップ言語を提供する。現在開発中。
- URL：http://www.daml.org/

(イ) DAML による知的な検索

Ex. 「James Hendler」が「共同著作者」である「SHOE」についての「最近の」
「論文」を検索する

- 時間の概念（「最近の」）について理解することが必要
- 「最近の」のような概念は、閉じた世界における仮定がない。（状況によって定義が変化する）

「最近の」の定義は複数存在する

複数のオントロジーの存在と、オントロジーのマッピングの必要性

Ex. Daniel Dennett が意識について書いた本があれば、それをスタンフォード図書館にとっておいてくれるように要求する

- 検索以外のサービスとの組み合わせ

(ウ) DAML における検索

複数のオントロジーの存在 オントロジー間のマッピングサービスを必要とする

(図 12)

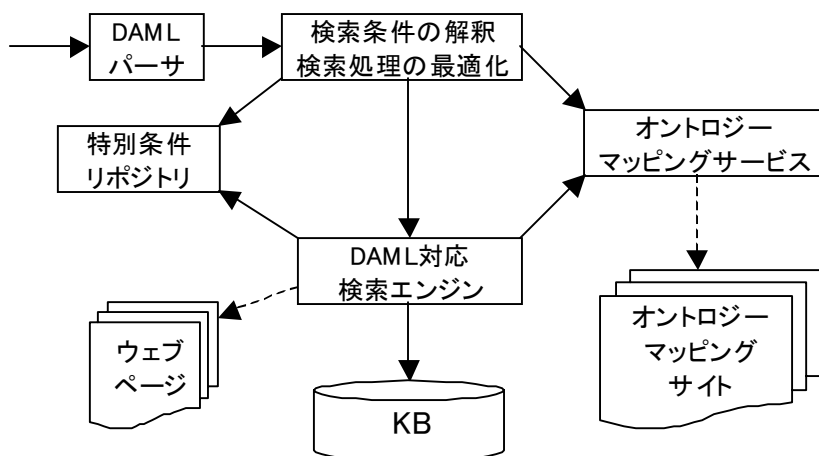


図 12 DAML における検索アーキテクチャ

(エ) DAML-S の目標

- サービスの発見・選択・利用を完全に自動化する
- 検索と、サービスの利用が、スムーズに係るようになる
- サービスの選択と利用における推論をサポートする

(オ) ウェブサービスのオントロジー

サービスを記述するオントロジーは、基本的に図 13の形をとる。

- サービスプロファイル: 使用条件・サービスから得られる結果についての記述。サービスを発見・選択するために必要。
- サービスモデル: サービスの意味的な記述。サービスが何をするのか、細かく記述する。（サービスのルール、状態変化、処理の流れなど）

- サービス基礎：サービスにアクセスするための詳細な実装情報。メッセージの形式や、使用する通信プロトコルなど。

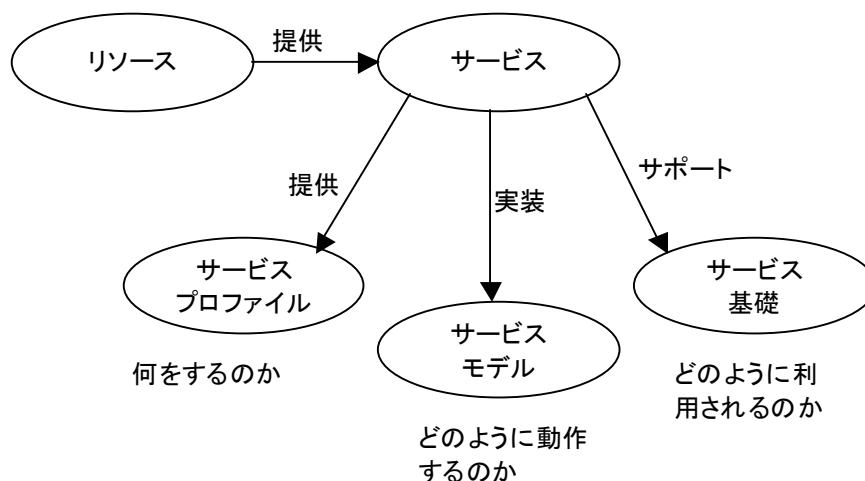


図 13 サービスのオントロジーの基本部分

- プロセスを表現する DAML エンコーディング

Event	precondition	parameter	effect
name	input	output	participant
startTime	duration	Sequence	Concurrent
- DAML の記述から，実行可能なプロセスモデルを構築するアルゴリズムを実装
 - 詳細は <http://www.ai.sri.com/daml/> を参照のこと．
- Stanford KSL, CMU, BBN, Nokia らと協力して開発

1 1 RDF Models for Dynamic Syndication and Wireless Application

セッション名：WF5 Semantic Web

発表者：Leon Shklar (情報技術者 / アメリカ)

RDF を使った、デバイスに応じたコンテンツの自動変換についての発表。メタデータにもとづく、デバイスの機能ベースの変換を行うことで、さまざまなデバイスに対して、低コストで一貫性のあるコンテンツ変換処理をすばやく用意することが可能になる。発表では、メタデータによる変換処理のコンテキストモデルと、コンテキストモデルに従って動作する Actor モジュールによる、コンテンツ変換処理のプロトタイプ実装についての説明を行った。

(ア) 従来への対応

デバイスや通信路を考慮したコンテンツの再利用が必要とされている

- デバイス依存の変換ライブラリ
- 新規デバイスへの対応が困難 高コスト・開発の遅れ・他のデバイスとの非一貫性

(イ) 新しいコンテンツ変換の流れ

RDF :

- 機械が理解できるメタデータ記述を目的とした設計
- すでに、コンテンツ変換を目的とした仕様がある (RSS)

CC/PP (Composite Capabilities / Preference Profiles) :

- WAP と W3C が協力して開発
- RDF ベース
- デバイスおよびユーザエージェントのプロファイル情報記述のための標準 (画面サイズ・キーボードの有無・表示可能な文字種類など。ユーザの好みも反映)
- 次世代サーバは、プロファイル情報と通信路のバンド幅の情報をもとに、適切なスタイルシートの選択・生成を行う
- 現在の (ベンダーなどによる) CC/PP サポートは不十分だが、将来的には改善される見込みがある

(ウ) プロトタイプ実装

CC/PP ベースのデバイスプロファイル情報のデータベースをもとに、コンテンツ変換を行うサービスのプロトタイプ実装を行った

- RDF によって変換処理コンテキストモデルを与える
- Java API によって、モデルのノードにおいて呼び出される、差し替え可能な処理モジュール (Actor) を実現している
- 処理コンテキストモデルは、3つのサブグラフから構成される (図 14) :
Content, Delivery, Transformation
- Content グラフ : コンテンツとコンテンツ処理の機能をモデル化している
- Delivery グラフ : デバイスやブラウザをモデル化している
- Transformation グラフ : デバイスの機能に対応した変換をモデル化している。
Delivery グラフとの関連をもっている。
- コンテンツ変換処理は以下ようになる
 1. Delivery グラフに基づいてデバイスおよびブラウザのプロファイル情報を計算する
 2. Transformation グラフに基づいて、計算したプロファイル情報からコンテンツの変換方法を計算する
 3. Content グラフに基づいて、デバイスおよびブラウザのプロファイル情報

- から、コンテンツ依存の変換を計算、実行する
4. プロファイル情報にもとづいて、コンテンツ非依存の変換処理を実行する

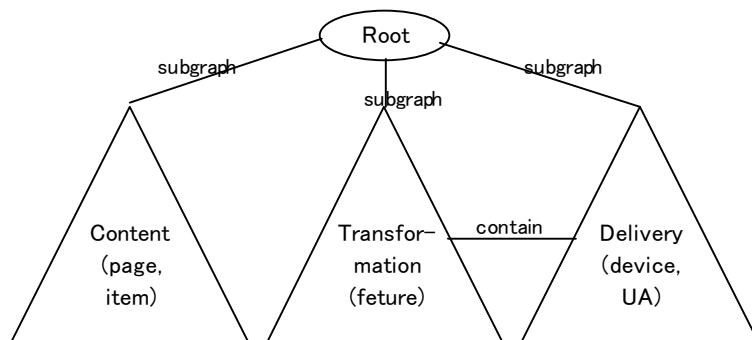


図 14 CC/PP サーバ内の処理コンテキストモデル

1 2 Mobilizing the Semantic Web with DAML-enabled Web Services

セッション名：WF5 Semantic Web

発表者：Shelia A. McIlraith (スタンフォード大学 / アメリカ)

複数の DAML によって記述されたウェブサービスを自動的に組み合わせて目的を解決するエージェント技術についての解説。

(ア) DAML とは

- RDF(S)の上に位置付けられる、意味情報を記述するための仕様 (図 15)
- RDF(S)では、意味情報を記述する能力が不足している
- DAML+OIL：オントロジー記述言語。記述論理とフレーム理論を元としている。
- DAML-L：明確な意味情報を備えた論理言語で、少なくともホーン節による命題論理を表現できる。(2002 年前半)
- DAML-S：サービス記述言語。

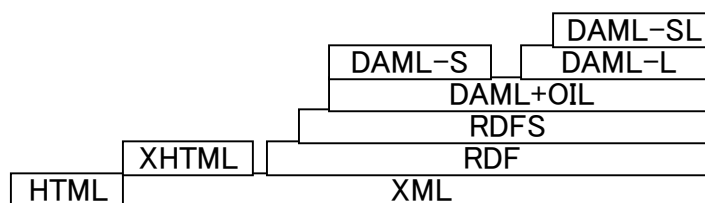


図 15 Semantic Web 関連仕様の階層構造

(イ) サービスを自動的に組み合わせるためのエージェント技術

- エージェントは、高い抽象度をもった、再利用可能な、「汎用手続き (generic

procedures)」にもとづいて動作する。

- 汎用手続きは、DAML-S によって記述される
- 汎用手続きは、ユーザの条件（会社のルール・個人の好みなど）によって制約される。
 - Ex. 汎用手続き「旅行の手続きをする」
パラメータ：出発地，旅行先（国外／国内），日程
条件：予算の上限，使用する旅行会社の種類，個人的な好みなど（「3時間以内なら自分で車を運転する」）
- エージェントの知識ベース（KB）（図 16）は、使用する汎用手続きとユーザの条件によって、自動的に構築される。
- 機械的な演繹処理によって、汎用手続きは条件とエージェントを囲む状況に関連付けられて、ウェブサービスに対する一連のリクエストが生成される。生成されたリクエストは、Agent Broker によって処理される。
- エージェントの知識ベースは、ウェブサービスからの応答内容に応じて変化する。
- 機械的な演繹処理は、Prolog によって実装された ConGolog 処理系によって行われる。（ConGolog：論理プログラム言語。シチュエーション・カリキュラス（状況計算）を動作理論とする）
- KB は ConGolog の状況計算式によって表現される

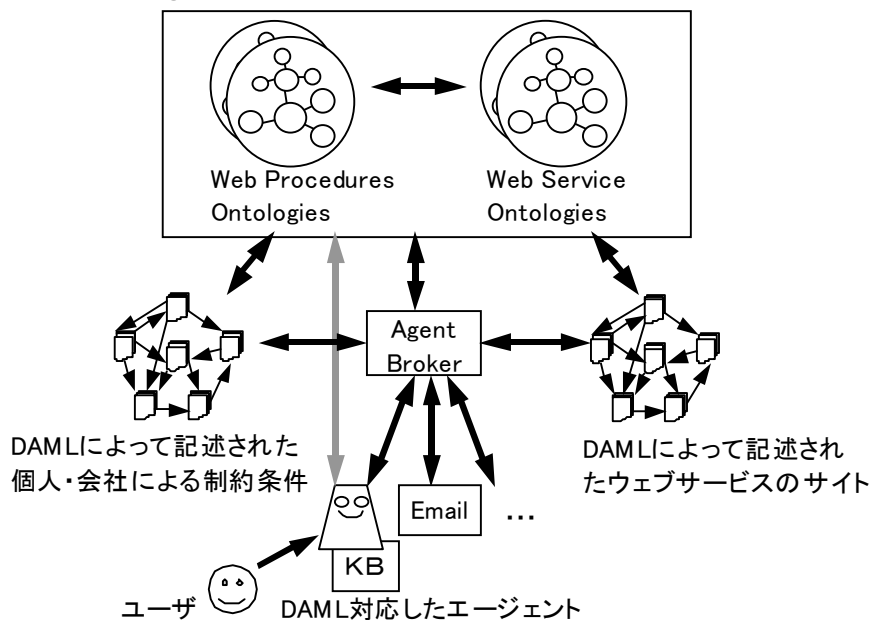


図 16 DAML をベースとした、Semantic Web のフレームワーク

1 3 The Design and Implementation of the Redland RDF Application Framework

セッション名 : P14 XML/RDF

発表者 : Dave Beckett(Bristol 大学)

RDF アプリケーションの開発フレームワーク、Redland の紹介。

(ア) はじめに

RDF は URI をベースとした汎用的な技術であり、その応用範囲は広い。RDF および RDF Schema は、ドメインにまたがって共通に利用できる機能を提供するものである。そのため、RDF を実装するアプリケーションには、高い柔軟性が必要とされる。

(イ) フレームワークの要件

- RDF の汎用性をサポート可能な、柔軟なシステム。
- RDF の使用目的に合わせたデータ格納手段が利用できる。RDBMS などの既存のデータ格納技術が利用できる
- 多くのアプリケーション、ユーザ、システムから使われるので、ポータビリティが重要。
- RDF(XML 形式)の解析は、トリッキーで面倒な作業なので、マークアップ部分を隠して、よりセマンティックな部分に集中できるようになっているべき
- 使用するプログラミング言語にあわせた使い方ができるべき

(ウ) フレームワークのデザイン

- 基本単位 : Statement(3 項組)、 Node(Subject, Object)、 Model(意味のある Statement の集合)、 Storage(Model を格納する)、 Stream(Statement の並び)、 Parser(RDF 文書を解析する)
- 階層 : Model, Storage といった抽象化により、RDF 応用アプリケーションは、低レベルな操作を気にする必要がなくなる。

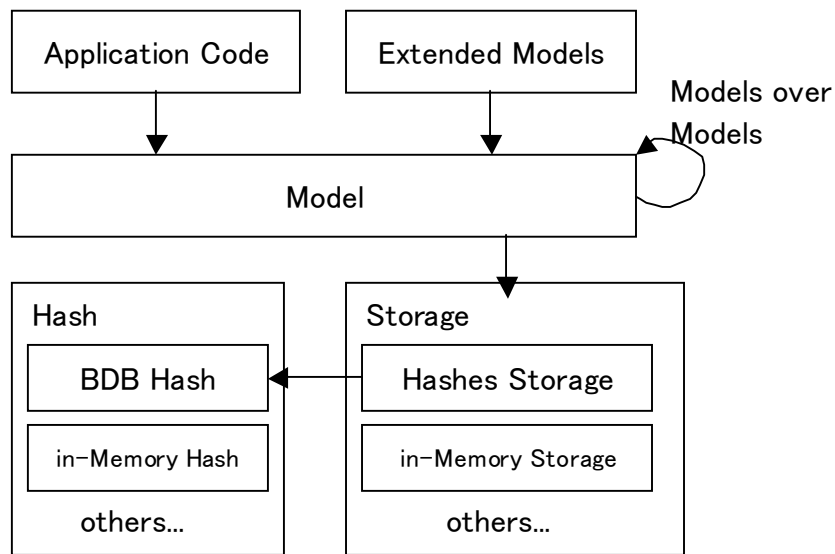


図 17 Redland のモデル階層

- モジュール：様々な既存の実装を利用するためのモジュールの仕組みが用意されている。動的バインドも可能。Storage(Berkley DB, in-memory(GDBM)、Parser(SiRPAC, libwww RDF Parser, Repat, Rapier)
- ポータビリティ：C 言語で実装。ただし、ツール (SWIG) によって Perl や Python インターフェースも用意 (自動生成) されている。また、オープンソースプロジェクトである。ライセンス形態は GPL/LGPL。

(エ) 応用アプリケーション

Redland は、開発開始から 6 ヶ月(2001/2 現在)程度のいまだ開発途中のフレームワークであるが、多くの機能が実装・テスト済みであり、安定しつつある。

- RDF Demonstration of Model with Persistent Store :
Berkley DB 上に構築した RDF のデータベースにアクセスするデモ。Web 上でアクセス可能。Redland の Perl インターフェースを利用。格納されているステートメントは 130 万以上、1 秒あたり 1800 のステートメントを結果として返すことができる。
- RSS 1.0 Demonstration :
RSS データを読み込んで HTML 形式に変換するデモを行った
- WSE Demonstration :
WSE (Web Search Environments) プロジェクトは、ILRT が出資するプロジェクトで、そこでは 3 つのシステムにまたがる RDF インターフェースとして、Redland を利用している。このシステムでは、およそ 11 万 Web ページを格納する非 RDF データベースが、4 万個の構造化データを格納する RDF データベースおよび Web ページとの関連データを格納する RDF データベースと接続されている。システムは、290 万個の RDF ステートメントを格納し、

そのときの Berkley DB のサイズはおよそ 2.1G バイトである。検索は、秒あたり 6000 ステートメントを返すことが可能である。

(オ) 今後

- RDF Query Language : モジュールとして追加
- Java インターフェース
- XHTML などのその他の形式のデータからも RDF データを抽出
- RDF データをより扱いやすくするための便利な API の追加

1 4 Combining RDF and XML Schemas to Enhance Interoperability Between Metadata Application Profiles

セッション名 : P14 XML/RDF

発表者 : Jane Hunter

様々なメタデータ言語間のマッピングの仕組み(MetaNet)を用意することで、メタデータ間の相互運用性の実現について発表した。

(ア) はじめに

- TV-Anytime や MPEG-21、OAI(Open Archives Initiative)は、既存の何種類かのメタデータスキーマを組み合わせた、アプリケーションプロファイルを必要としている。
- 種類の異なるメタデータ間にまたがった検索の必要性など、メタデータの相互運用性が必要になってきている

(イ) Semantic Web Metadata Architecture

- Web メタデータアーキテクチャの全体像(図 18)
- MetaNet - metadata のオントロジー。RDF スキーマとして表現され、共通の基礎をなす拡張性のあるボキャブラリをベースにしている。これは、特定ドメインのオントロジー (RDF Schema) から生成される。
- XSLT - XML 形式のメタデータ間の変形を行う。MetaNet のセマンティックな知識と結びつくことで、アプリケーションプロファイル間の柔軟で動的なマッピングが可能になる。
- アプリケーションプロファイル - 複数の既存の仕様をもとにした、XML Schema ベースの仕様。

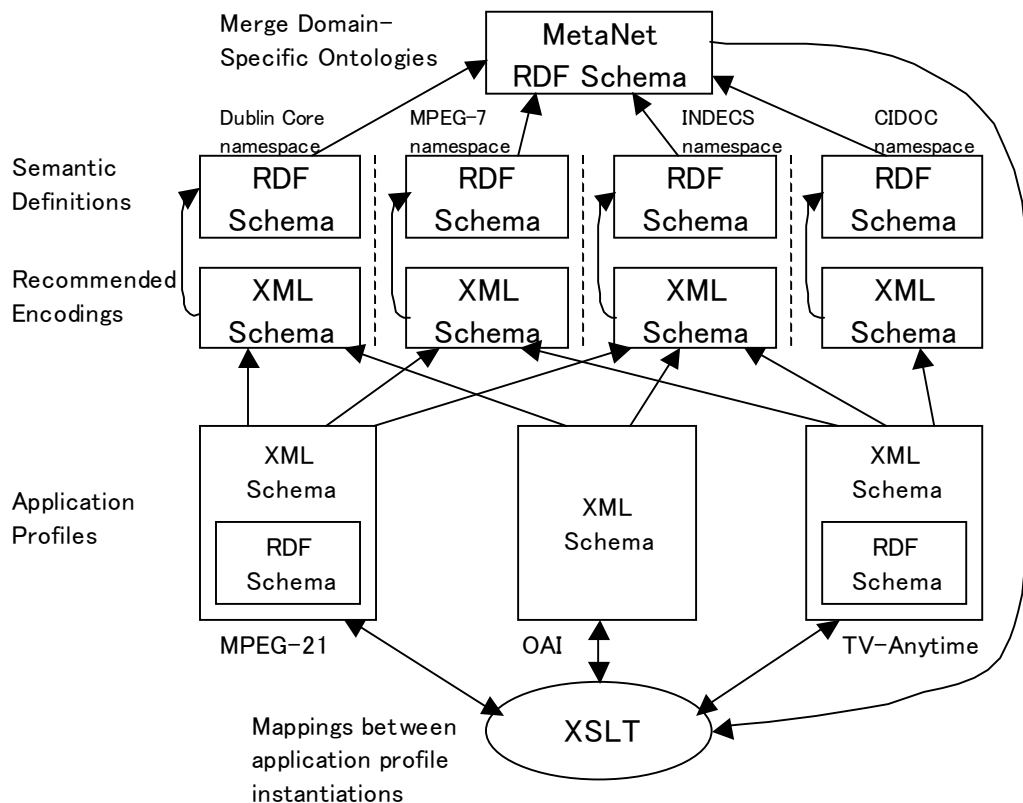


図 18 Web メタデータアーキテクチャの全体図

(ウ) RDF スキーマと XML スキーマを組み合わせる

- スキーマ言語には、RDF Schema と XML Schema がある。(DTD は namespace をサポートしていないので考慮しない)
- RDF Schema は意味的な表現に関する豊富なサポートを提供する。一方で、ローカルな使用(構造体や、基本型やデータ型制約)などのサポートが限られている
- XML Schema は、構造体や基本型、データ型制約などをはっきりとサポートしている。しかし、柔軟で動的なマッピングを可能にするために必要な、意味情報に対するサポートが限られている。
- RDF Schema と XML Schema を互いに補完する形で利用することがポイント

(エ) RDF スキーマと XML スキーマの組み合わせ方

RDF Schema と XML Schema を組み合わせる方法には、以下の 2 通りがある

- RDF Schema のデータを、XML Schema の annotation 要素内に埋め込む
- XML Schema から、外部の RDF Schema 文書へのリンクを追加する。

(オ) MetaNet – 意味情報の相互運用のための共通オントロジー

- ドメインごとに、RDF Schema が存在している状態。まずは、それらを一つの RDF Schema へまとめることが必要。
- さらに、まとめた RDF Schema を、柔軟かつ動的なマッピングが可能になるように、XSLT とリンクさせる必要がある。
- 理想的には、機械的に生成されるべきだが、MetaNet は人手で作成している。
- ABC ボキャブラリのコア部分を、ベースとなるオントロジーとしている。

(カ) 意味情報を XSLT に追加

- XSLT で変換を行うのには無理があった
- メタデータ間の、構造的・文法的な変換は非常にうまくいく
- 意味情報に基づくマッピングは、ハードコードされる必要がある。
- ゆるやかな制約あるいは変化する入力には適していない
- 意味情報のマッピングは、Java のメソッドを呼び出す拡張を、XSLT 実装に加えることで実現(図 19)

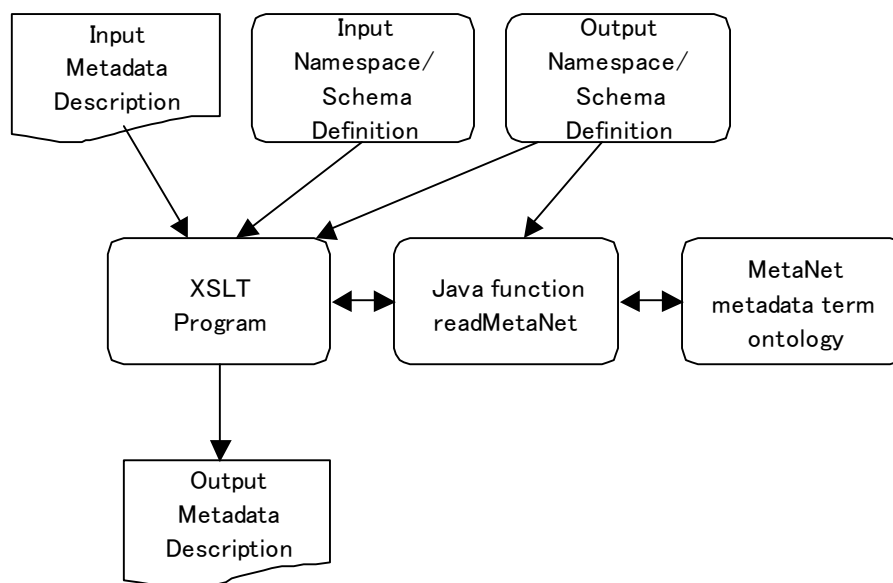


図 19 メタデータのマッピングにおける処理の流れ

(キ) 今後について

- DAML/OIL を使って MetaNet を表現する
- Schematron のような、軽量ルールベースアプローチによるアプリケーションの調査
- RELAX や TREX などの、その他のスキーマ言語への対応

1 5 Enabling knowledge representation on the Web by Extending RDF Schema

セッション名 : P14 XML/RDF

発表者 : Jeen Broekstra (Vrije 大学・オランダ)

Web 上でのオントロジーの利用が注目されている。RDF Schema(以下 RDFS)は Web リソースのメタ情報を表現するのに適しているといえるが、RDFS だけでは完全な知識表現を行うことは難しい。そのため、OIL(Ontology Interface Layer)を用いて RDFS を拡張する。これにより、RDFS との互換性を保ちつつ、高い表現力をもった意味情報の記述が可能になる。

(ア) RDFS の欠点

- 意味を簡潔に表現できない
- 推論モデルがない

(イ) OIL(現在は DAML + OIL)

- RDF をベースとしている
- フレームベースにおけるモデリングのための基本機能および、記述論理を提供する。
- 自動的推論サポートが提供される(マンチェスター大学の FaCT システム)。
- OIL の後継者として DAML+OIL が存在する。DAML+OIL は、W3c Ontology Working Group へ提出される。

(ウ) RDFS の拡張

- RDFS をオントロジー言語として利用できるようにすることが目的
 - RDFS に簡潔な意味情報を与える
 - RDFS にモデリングのための基本情報を追加する
- RDFS の拡張として OIL を利用する (図 20)

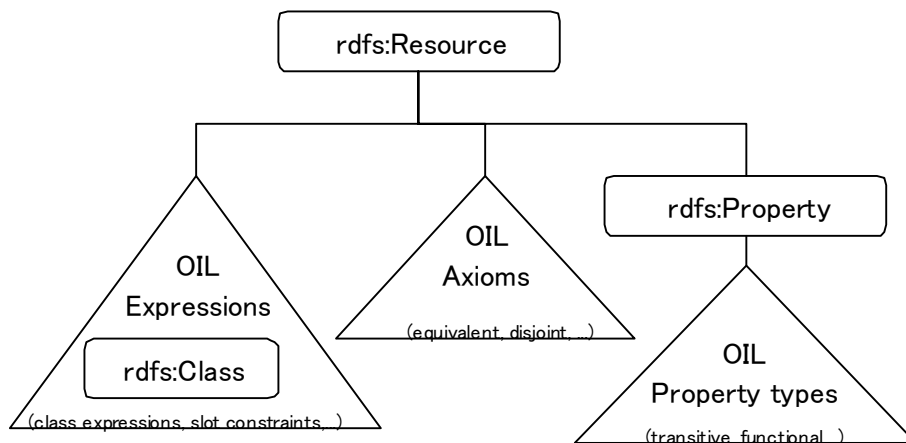


図 20 OIL による RDFS の拡張

1.6 Annotea: Shared Web Annotations for Building an Open RDF Infrastructure

セッション名 : P20 XML Tools 2

発表者 : Jose Kahan(W3C)

Annotea の紹介。Annotea は、Web ページに注釈をつけ、さらにつけた注釈を共有することを可能にするシステムであり、RDF をベースとして、XPointer、XLink といた W3C の技術を可能な限り再利用している。現在、Amaya(W3C)などで実装が行われている。

(ア) 基本アーキテクチャ

Annotea の基本アーキテクチャを図 21 に示す。注釈情報は RDF 形式のメタデータ (図 22) として、サーバに格納する。

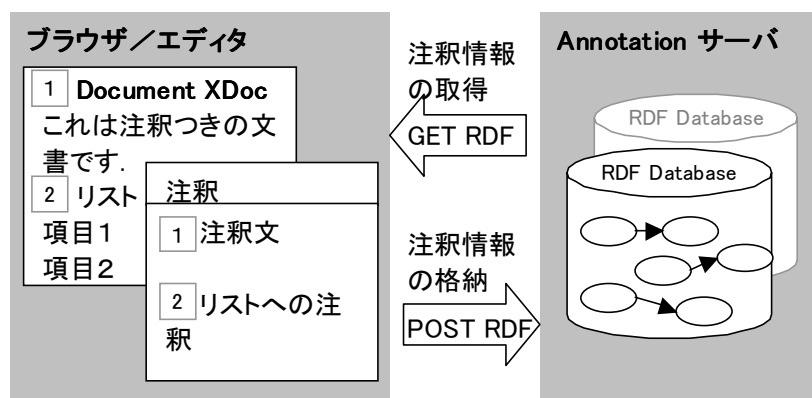


図 21 Annotea システムの基本アーキテクチャ

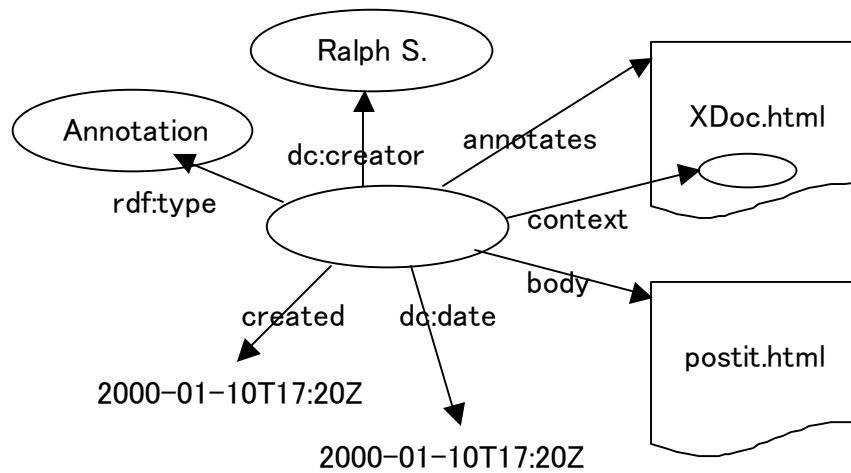


図 22 注釈情報の RDF モデル

(イ) 対応クライアント

Annotea はクライアント独立であるため、複数のクライアント実装が存在している (Amaya(図 23), Annozila, Bookmarklet,...)

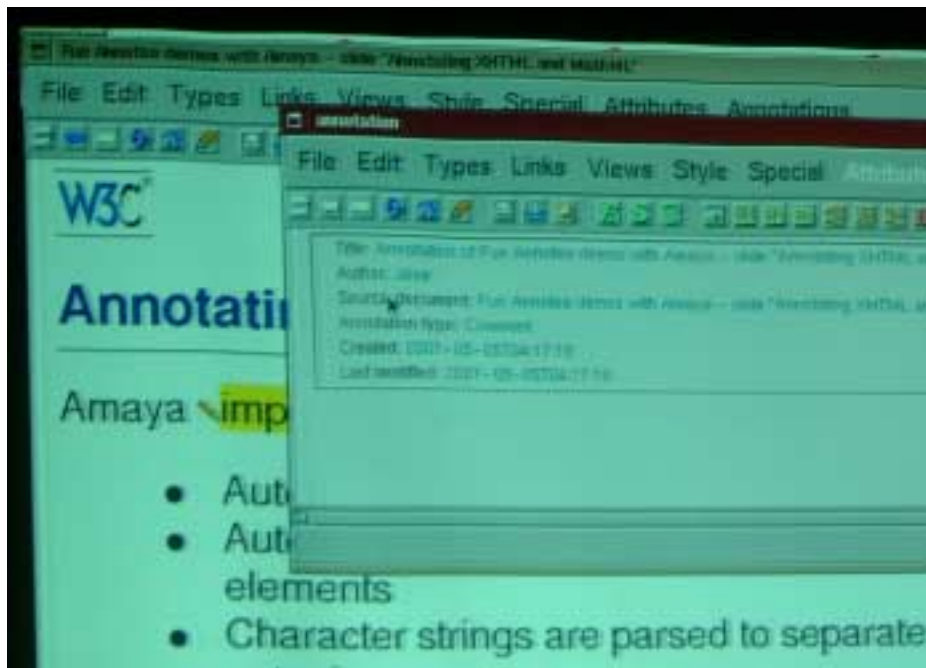


図 23 Amaya を使って注釈をつけている様子

(ウ) 問題点

- 誤った個所に注釈がついてしまう：一度注釈をつけた個所の前後に修正が入ると、修正によって文書中の位置がずれてしまうために、XPointer による指

定が正しくないものになってしまうことがある

- 注釈がみなしごになってしまう：注釈をつけた個所を削除すると、指し示すものがない注釈情報ができてしまう。

1 7 RDFdb and Algae

セッション名：D3 Semantic Web

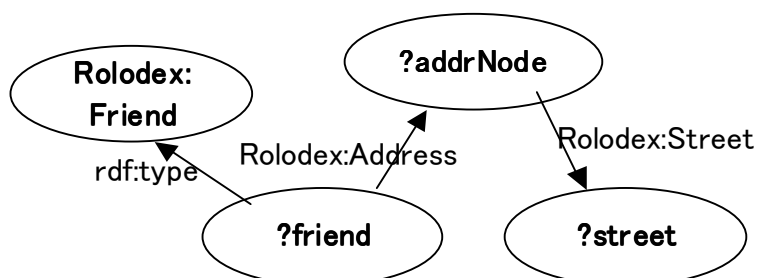
発表者：Eric Prudhommeux

RDF データベース RDFdb、ObjectDB と、問い合わせ言語 Algae の紹介。実装は Perl で行われている。

(ア) Algae

- S 式のようなクエリ文法。
- 宣言型の問い合わせを行う。

```
(ask '((rdf:type ?friend Rolodex:Friend)
      (Rolode:Address ?friend ?addrNode)
      (Rolodex:Street ?addrNode ?street)
      ) :collect '(?friend ?street))
```



1 8 Building a Semantic Web Site

セッション名：D3 Semantic Web

発表者：Eric van der Vlist

実際の Semantic Web 技術の応用例として、ニュースサイトでの RSS の利用について紹介した。RSS1.0 と、分類モジュールを使って、ニュース記事を RDF データベースへの格納し、さらに XML Topic Map の生成を行う。

(ア) 全体図

- RSS1.0 によって、サイトの構造を記述する
- ニュース記事中に埋め込まれた意味情報は、分類マークアップ情報に変換される

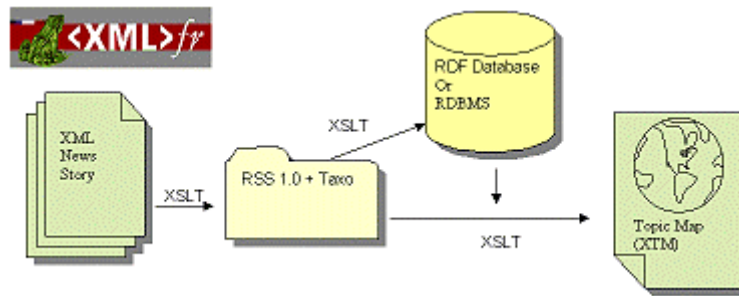


図 24 <XML>fr ニュースサイトにおけるニュース記事分類

(イ) RSS

- RDF(あるいは Rich) Site Summary の略
- もともとは、Netscape 社が自社ポータル(My Netscape)のために導入した。これが RSS 0.9 で、この時点では、RDF とは無関係だった。
- その後、RSS は他のポータルサイトでもりようされるようになるが、ボキャブラリの問題などから 2000 年中ごろには使われなくなっていた
- RDF 1.0 ワーキンググループが結成され、RDF を中心としてモジュールによる拡張機能をそなえた RSS1.0 仕様が、2000 年 12 月に発表された
- RSS1.0 といっしょに、Dublin Core モジュールと、いくつかのサポート用ツールが発表された
- 分類モジュールは現在開発中であり、ここではその開発中のものを利用している

(ウ) XTM Topic Map

- XML 文書(群)中の、トピックやトピック間の関係を記述するための仕様
- ニュースサイトでは、RSS のサイト構成情報と分類情報から生成している
- 参照 URL : <http://www.topicmaps.org/xtm/1.0/>

```

<topic id="hamlet">
  <instanceOf><topicRef xlink:href="#play"/></instanceOf>
  <baseName>
    <baseNameString>Hamlet, Prince of Denmark</baseNameString>
  </baseName>
  <occurrence>
    <instanceOf>
      <topicRef xlink:href="#plain-text-format"/>
    </instanceOf>
    <resourceRef
      xlink:href="ftp://www.gutenberg.org/pub/gutenberg/etext97/1ws2610.txt"/>
    </resourceRef>
  </occurrence>
</topic>

```

トピック : hamlet
 トピック名 : Hamlet, Prince of Denmark
 トピックの型 : 劇(play)
 関連リソースの型 : テキスト (plain-text-format)
 関連リソースへのリンク : ftp://www.gutenberg.org/pub/...

図 25 XTM 文書のサンプル(XTM 仕様より)

19 RDF Expert

セッション名 : D3 Semantic Web

RDF で記述されたデータをもとに動作する、簡易エキスパートシステムの紹介

(ア) 概要

- forward chaining あるいは backward chaining をサポート
- 入力データは RDF (知識と目的)
- Jena を API に利用

(イ) 動作の様子 (図 26)

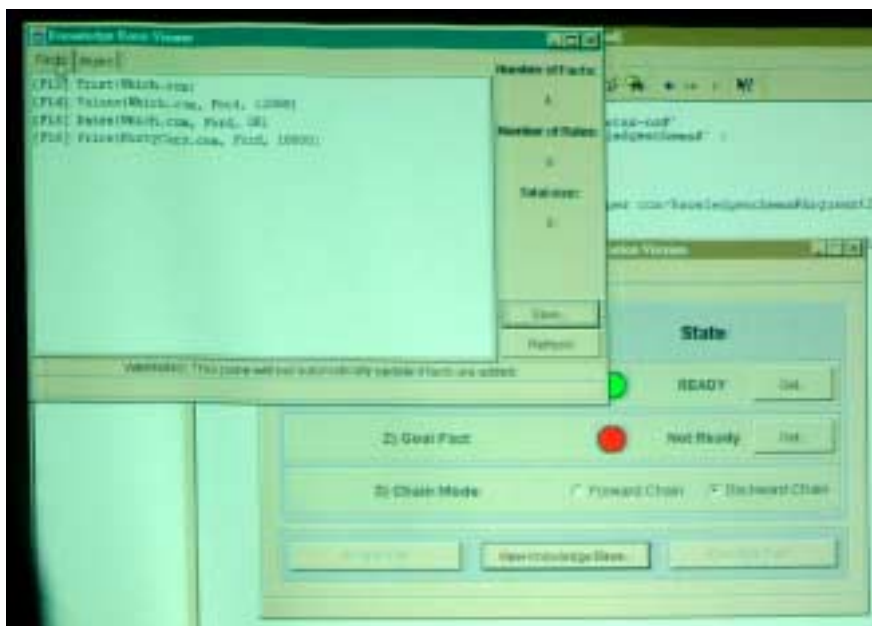


図 26 RDF Expert の動作中の様子

2 0 WebKB

セッション名 : D3 Semantic Web

知識ベース WebKB の紹介。Web 上で「知識」を登録し、共有できることが特徴。初期化のベースとして Wordnet (英語の語彙データベース。オンラインで入手できる)を使用。

- デモでは、「priest」と「minister」の2つの単語の比較を行っていた。(図 27)
- WebKB の URL は

<http://meganesia.int.gu.edu.au/~phmartin/WebKB/webKBshared.html>

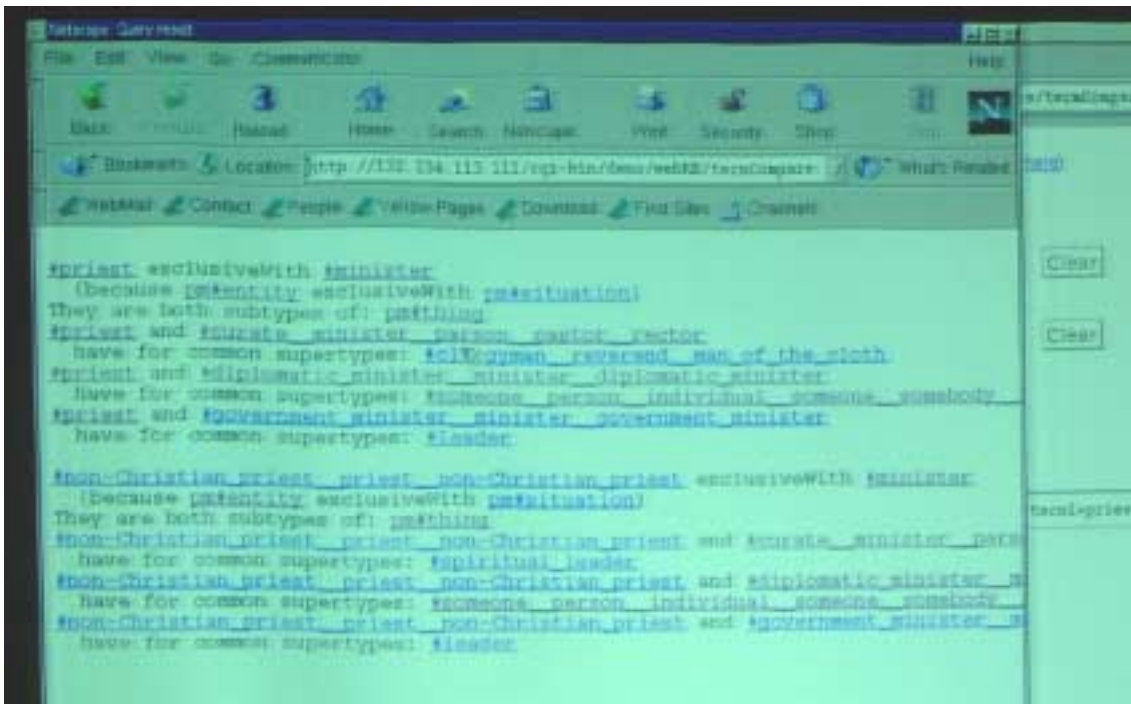


図 27 WebKB による単語の比較の実行例

2 1 Sesame

セッション名 : D3 Semantic Web

RDF を格納するスキーマベースのレポジトリとその検索機能によって構成される Sesame の紹介。

(ア) 概要

- Administrator Nederland bv で開発されている
- ヨーロッパ IST プロジェクト On-To-Knowledge の主要配布物の一つ
- 検索言語として、OQL スタイルの RQL を使用する
- データベースの内容をオントロジーとして取り出すことが可能
- まだ バージョンの段階
- URL : <http://sesame.administrator.nl/>

以上