

フレーム構造論理を用いた XML からの 知識ベースの構築

小 松 香 爾

1 はじめに

1990年代のインターネットの普及に WWW が果たした役割は大きい。WWW 上の WEB サイトを通じて、個人や組織は、様々な情報を得たり発信したりしている。得られる情報の、規模、種類、表現の豊かさという点で、WWW は書籍やテレビなどの他のメディアに対し圧倒的優位にあるといえる。WWW の規模が急速に拡大したのは、情報源としての利便性の高さによる。しかし、情報量の爆発とともに、高度な情報検索の必要性が生じ、検索エンジンの精度の改善は重要な問題となっている。現在、コンピュータが文書の意味を理解した上で行う高度な検索を実現するための新しいモデルとして、セマンティック Web が注目されている。また、古くから人工知能や自然言語処理の分野で、実用的な知識ベースが必要とされてきた。セマンティック Web は知識ベース構築の際のソースとなることも期待されている。従来の HTML で書かれた Web ページがコンピュータでの意味解析が困難であるのに対し、マークアップ言語 XML (Extensible Markup Language) で書かれた Web は意味解析がある程度は可能である。それゆえ、セマンティック Web は、高度な検索や計算機が理解可能な Web ページの構築を目指す、革新的な Web テクノロジと見なされている。本論文は、セマンティック Web を記述する XML から知識ベースを構築する手法について論じる。現在、セマンティック Web 上で推論を行うために、OWL DL (Web Ontology Language Description Logics) ⁽¹⁾ という記号論理が研究されている。しかし OWL DL には様々な制約があり、マークアップ言語から OWL DL への自動変換する研究も行われていない。その理由は、記述論理自体が一階述語論理のサブセットとして設計されたからである。自然言語から一階述語論理の論理式への機械的な変換は不可能である。一方、我々が開発したフレーム構造論理は、自然言語から論理式への変換を重視した論理体系であり、XML との構造的な類似性も高い。本論文では、XML からフレーム構造論理への自動変換の可能性を論じる。以降、2 節で XML をはじめとするマークアップ言語について、3 節でセマンティック Web の現状と応用技術について、4 節でフレーム構造論理の記述法と公理、5 節で XML からフレーム構造論理の論理式への変換について述べる。

2 マークアップ言語による文書記述

マークアップ言語は、文書の一部を「タグ」と呼ばれる特別な文字列で囲うことにより、文章の構造を文章中に記述していく記述言語である。SGML, HTML, XML, TeX は代表的なマークアップ言語であるが、それぞれ性質や目的は異なる。本節では、TeX 以外のマークアップ言語の特徴を、文書記述の観点で述べる。

2.1 SGML の特徴

XML は汎用マークアップ言語 SGML (Standard Generalized Markup Language) より派生したメタ言語である。SGML 自体は電子的な文書管理を目的として開発された。各種文書を長期間保存したり、人手を使わない自動管理を行ったりするために、論理的な文書構造を記述する言語である。SGML の大きな特徴は以下の二点である。

①メタ言語である

②マークアップ言語である

①は「言語を作成する言語である」ということを意味する。したがって、ユーザが文書管理のフォーマットを、目的や環境に応じて作成できる。②は「テキストファイルの中に内容と共に付加情報を記述する」ということを意味する。したがって、SGML のデータは通常のテキストであり、テキストエディタで操作可能である。テキストであることから、もちろん人間にも可読である。

2.2 HTML での文書記述

HTML は、SGML に基づいて書かれた実装の1つであり、インターネットを急速に普及させた大きな技術要因である。HTML は本来文章の論理構造を記述する言語であったが、主に Web ブラウザメーカーによる拡張の結果、Web ページの見栄え (フォントや文字の大きさ、文字飾り、行間、色など) を記述するタグが取り込まれた。HTML4.0. では文書の見栄えに関する記述は CSS (Cascading Style Sheet) を使って行なうように改められた。HTML では p 要素は段落を表すというように、要素や属性がどのような意味を持っているのかが決まっている。ウェブページの作成者は、意味付けされた要素や属性の中から、どのように表示されるウェブページを作成するかに応じて、必要な要素や属性を選択することによって、HTML テキストを記述する。しかし、HTML では要素や属性を用いて記述された情報の中身そのものに関しては、意味付けを行うことができない。したがって HTML は、Web ページを表示する為に特化された言語であるといえる。

2.3 XML での文書記述

HTML が Web ページ記述用の言語であるのに対し、XML は用途を制限されていない。それゆえ、要素や属性の意味付けの定義は XML の仕様の中に含まれない。XML の仕様では、⁽²⁾文法的なルールと記述方法だけが定められている。なお、HTML の後継言語である XHTML (Extensible Hyper Text Markup Language) という言語を作るために XML が使用されているが、XML そのものは SGML の後継言語である。XML は、SGML の役割であ

る電子的な文書管理に加えて、電子的なデータの交換という目的で設計された。そのため、文書だけでなく、単純な数値データの交換にも使用される。それゆえ「XMLはデータベースの情報を交換するための言語」あるいは「XMLは電子商取引のための言語」といった誤解がある。XMLがデータ交換などで利用されるのは、XMLの厳密性と柔軟性によるものであり、本来XMLは文書を扱うためのメタ言語である。以下に、XMLで記述された文書を例示する。

```
<?xml version="1.0" encoding="Shift_JIS"?>
<Data>
  <Book ID="1">
    <Title>XML 入門</Title>
    <Author>小松香爾</Author>
    <Summary>XMLの基本を理解するためのコンパクトな入門書</Summary>
  </Book>
</Data>
```

上記のXMLの文書は、HTMLで記述された文書と比較すると、見た目という点では、ほとんど変わらない。XMLもHTMLもSGMLから派生したマークアップ言語であり、要素や属性の存在といった、マークアップ言語の特徴は共通している。ただし、XMLには、「終了タグの省略が許されない」「空要素は、開始タグではなくて、空要素タグで記述する」「属性指定の際に属性値を引用符で必ずくくらずにはならない」などのような、HTMLにはない文法の厳密性がある。データ交換だけでなく、コンピュータを用いた文書管理や意味解析において厳密性は重要である。さらに、XMLの文書は、HTMLの文書とは本質的な違いがある。例えば上の例では、要素bookで1つの本を表し、その中の要素authorでその本の著者を表すなどのように、マークアップ要素の意味付けが設計されてなくてはならない。要素の意味付けの設計はユーザが行うこともでき、XMLが柔軟であるといわれる理由となっている。要素の意味付けを行う言語はスキーマ言語と呼ばれ、SGMLではDTD (Document Type Definition) と呼ばれるスキーマ言語が使われ、HTMLの定義などが実際にDTDで定義された。SGMLから派生したXMLでも、スキーマ言語として通常はDTDが使われる。しかし、DTDには「データ型が定義できない」、「文法がXMLと異なる」、「ネームスペースに対応していない」などの欠点があり、より強力なXML仕様と整合性のある言語の必要性がでてきた。W3Cでは、XMLスキーマと呼ばれる新たな言語の標準化を進めている。しかしその標準化には、業界の有力企業が参加しているため、仕様が複雑である。なお、データ交換という観点では、形式の設計の柔軟性は弱点にもなる。データ交換をするためには、形式が統一されている必要があり、XML文書の構造を別の形式に変形するための変換ルールを記述するための簡易言語XSLT (XML Stylesheet Language Transformations) が存在する。

3 セマンティック Web の概観

現在、Web 関連の技術として最も注目されているのが、セマンティック Web である。セマンティック Web は、WWW の提唱者であり、Web 技術の標準化団体である W3C (World Wide Web Consortium) の代表でもある Tim Berners-Lee が提唱する革新的なモデルである。

3.1 セマンティック Web の階層構造

セマンティック Web の特徴は、コンピュータで意味解析が可能な XML で記述されたメタデータの空間を構築することであり、階層構造によって表現されることである。(図 1 参照) まず第1層のセマンティック Web のソースは、URI で指し示される Web 資源である。ところが、現在 Web ページの記述で最もよく使われる HTML は知識表現言語としての記述力は⁽³⁾ ない。第 2 層のセマンティック Web のメタデータ記述には、SGML の改良版である XML が用いられる。XML スキーマはデータ型などを記述するものであるが、スキーマとして DTD が使われることも多い。第3層で、コンピュータ処理を前提とした知識表現のための言語として用いられるのが RDF (Resource Description Framework) である。その RDF の属性などを定義するのが、RDF スキーマである⁽⁵⁾。さらに第 4 層で、RDF に基づくオントロジを構築するための言語が OWL (Web Ontology Language) である⁽⁶⁾。第 5 層以上で使用される言語は現在のところ定まっていない。以降、本節ではセマンティック Web の現状とその応用について論じる。

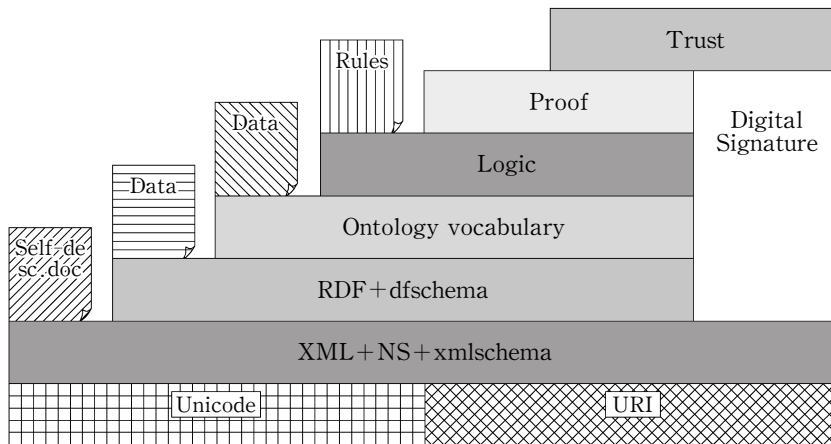


図 1 セマンティック Web の階層アーキテクチャ

3.2 セマンティック Web の現状

セマンティック Web は研究分野自体が新しく、特に知識ベース構築のための技術は非常に新しい。セマンティック Web 上でのオントロジ構築のリソースとなる RDF は XML で記述

される。図1で、XMLはデータ表現層に存在し、RDFは知識表現層に存在する。W3CによるRDFとOWLの正式勧告は2004年2月であり、図1のオントロジ層より上の層に関しては、正式勧告が出ていない。OWLによるオントロジの実装は研究段階であり、実用的な実装は存在しない。RDFによるメタデータ記述は既に実用レベルに達している実装がある⁽⁷⁾。例えばAdobe社によるオープンスタンダードのXMP (Extensible Metadata Platform) である。XMPは、RDFに準拠し、メタデータの取り込みと管理のための標準化されたフレームワークを提供する。報道関係者は、XMPのメタデータスキーマを使って、キャプション、クレジット、場所、時刻、日付などの詳細情報を写真や画像内に埋め込む。埋め込んだメタデータは、コンテンツ管理システムで読み取ることができ、マスメディアが利用できる情報になる。マスメディアは、メタデータを使って、記事に添付する写真を検討できる。このXMPはW3Cが勧告したRDFに準拠したメタデータであるが、RDFに準拠しないメタデータは、現在でも多数存在する。EUのMIReG (Managing Information Resources for e-Government)、英国のe-GMS (e-Government Metadata Standard)、デンマークのOIO-metadata (Offentlig Information Online-metadata)、オーストラリアのAGLS (Australian Government Locator Service)、米国のGILS (Government Information Locator Service) など、いわゆる電子政府が発行する文書である。ただし、これらのメタデータの記述は各国が独自で定義したものであり、メタデータの利用は国内に限定される。また、セマンティックWebには、知識を記述するメタデータの自体の記述が困難であるという問題もある。それゆえ「メタデータがないから有益なサービスを作れない」「有益なサービスがないからメタデータを作っても仕方がない」というデッドロックが生じている。現状では、人手でメタデータを記述することは、費用対効果の観点で、既存のWebに劣っていると見なされている。HTMLが多くの人の手によって書かれ、急速に普及したのは、費用対効果で優れていたのが大きな一因である。この問題を解消するため、大量のWeb文書(2004年現在、数十億のURLが存在すると推定されている)から、有益な知識を抽出するWebマイニングにより、メタデータを自動生成する試みがある。しかし、既存Webはゴミも多く、現在、メタデータの自動生成は研究レベルである。さらに、メタデータの種類の決定も問題になる。現在世界的に合意が取れているメタデータの要素は、Dublin Coreと呼ばれる、Title, Creator, Subject, Description, Publisher, Contributor, Date Type, Format, Identifier, Source, Language, Relation, Coverage, Rightsの14の属性しかない⁽⁸⁾。しかし、このような状況にもかかわらず、セマンティックWebの将来性は、研究者だけでなく一般企業まで広く知れ渡っており、今後、発展が期待されている。

3.3 Web 検索の重要性と精度の向上

検索エンジンはWebサイト検索の要求に応えるツールであり、1994年に開発された。今現在、YAHOO、MSNなどのポータルサイトのメインとなるコンテンツは検索エンジンであり、ビジネス面においても注目されている。ユーザは、検索エンジンにキーワードを入力することにより、興味のあるWebサイトにアクセスできる。検索エンジンの種類が異なる場合、精度

や使いやすさには差があり、検索エンジンの性能差が、ポータルサイトにアクセスする人数に大きく影響するといわれている。現在、ポータルサイトの収入のほとんどが、企業のバナー広告に頼っていることを考慮すると、検索エンジンのビジネス面での重要性も極めて大きいといえる。

しかし、検索の重要性にも関わらず、検索エンジンのコアとなる技術は、登場時のものから本質的には改良されていない。他のサイトからのリンク数による順位付けが行われたり、キーワードの関連語への拡張が研究されているが、キーワードと、Web ページの内容を表す索引語との類似性を比較し、類似性の高いページを上位に表示するという部分は変わらない。また、ユーザの検索要求を汲み取りながら、動的に索引語を決定するといった、いわゆる柔軟な検索の研究が行われているが、速度の面で実現が厳しいことに加え、コンテンツの意味解析を行わない適切な索引語の決定は困難である。劇的に検索精度を上げるためには、コンテンツの意味が計算機に扱える形で記述されなければならないといえる。例えば、現在の検索エンジンで、「渋谷にある美容院」を検索する場合、「渋谷」と「美容院」による検索を行う。しかし、Web ページ中の単語とのマッチングでは、検索の意図と異なるページがヒットするケースが多い。「渋谷」AND「美容院」という検索の結果は、「渋谷氏が経営する美容院」、「誰かが渋谷さんと美容院にいったという内容を含む日記」などの検索意図からかけ離れた情報もヒットしてしまう。問題点は「渋谷」の意味が考慮された検索がなされていないことであり、本質的な解決は、「渋谷」に関して「渋谷は地名である」というメタデータを付与しておく以外にない。

3.4 セマンティック Web の利用

セマンティック Web とは、コンテンツの意味情報を記述したメタデータを含むような Web である。メタデータを利用する検索が行えれば、Web ページの意味を考慮した検索が実現できることになる。しかし、セマンティック Web の研究分野で注目されているのは、検索への応用だけではない。古くから人工知能や自然言語処理の研究分野で、知識ベースの必要性が論じられてきた。これまで、電子辞書としてEDR⁽⁹⁾、オントロジとしてCYC⁽¹⁰⁾といった知識ベースが人手で構築されたが、規模や質に問題があり、実用化にはいたってない。各個人による認識のバラツキなどの問題もあり、実用的なデータベースの人手による構築は非常に困難であるといえる。しかし、WWW 上には、膨大な量の Web 文書が存在し、それらの文書から意味情報を抽出できれば、大規模な知識ベースをつくることができる。そして、現時点では、セマンティック Web のメタデータから意味情報を抽出することが、知識ベースを自動構築する唯一の方法であるといえる。なぜなら、従来の HTML で記述された Web は、構造的に文章を表示することに重点がおかれ、文書の意味を記述することは軽視されているからである。また、HTML は文法が曖昧に設計されており、そもそも機械処理には不向きな言語である。一方、セマンティック Web は、計算機による解釈の可能性が重視されている。セマンティック Web を構築する XML, RDF, OWL などはマシンリーダブル言語であり、人間だけでなく計算機にも解釈が可能であるように厳密に設計されている。

4 フレーム構造論理の構文と公理図式

本節では著者らが開発したフレーム構造論理の体系FS1⁽¹¹⁾の概要を説明する。フレーム構造論理は、もともと自然言語を記述するために構築された。ただし、コンテンツ記述言語の類ではなく、意味論が存在する純粋な記号論理の体系である。以下に、多様な属性関係を表現できるように拡張したフレーム構造論理の体系FS1を簡単に記述する。意味論は省略するが、集合論をベースとした意味論（初期のフレーム構造論理の意味論は束論をベースとしているが本質的な違いはない）の上で健全性、すなわち「定理ならば恒真式」と、完全性、「恒真式ならば定理」⁽¹²⁾が証明されている。

4.1 体系FS1の構文

〈基本記号〉

- (1) 属性名： l_1, l_2
- (2) オブジェクト記号： $\pi, a, b, c \dots$
- (3) オブジェクト演算子： \cdot
- (4) オブジェクト関係記号： $<, E_n$
- (5) 論理記号： \Rightarrow, \sim

オブジェクト記号のうち π を特別に全称オブジェクトと呼ぶ。直感的には「あらゆる概念を包含する概念」を表す。論理記号は命題論理で用いられる含意 (\Rightarrow) と否定 (\sim) である。

〈オブジェクト〉

- (1) t がオブジェクト記号であるとき、 t はオブジェクトである。
- (2) α, β がオブジェクトであるとき、 $(\alpha \cdot \beta)$ はオブジェクトである。
- (3) t がオブジェクト記号であるとき、 $t[ap_1, \dots, ap_n]$ (n は 1 以上) はオブジェクトである。各 ap_i は以下のいずれかの形をしたものである。
 - ① $L \rightarrow \alpha$
 - ② $L \rightarrow \{\alpha_1, \dots, \alpha_n\}$
 - ③ $L \leftarrow \{\alpha_1, \dots, \alpha_n\}$

ただし、上記で α はオブジェクト、 $\{\alpha_1, \dots, \alpha_n\}$ はオブジェクトの集合、 L は属性名を表すものとする。

〈オブジェクト関係〉

α, β がオブジェクトであるとき、以下はオブジェクト関係である。

- (1) $\alpha < \beta$
- (2) $E_n(\alpha)$

直感的な意味は、[1]は「 α は β である」、[2]は「 α に対応する概念が存在する」である。

〈論理式〉

- (1) P がオブジェクト関係のとき、 P は論理式である。

(2) P, Q が論理式するとき, $(P \Rightarrow Q), \sim(P)$ は論理式である。

ただし, 論理記号「 $\wedge, \vee, \Leftrightarrow$ 」は, 命題論理と同様に「 \Rightarrow 」と「 \sim 」から定義する。

4.2 公理体系 FS1

<公理図式>

- (1) 命題論理の公理
- (2) $\alpha < \pi$
- (3) $\alpha < \alpha$
- (4) $\alpha < \beta \wedge \beta < \gamma \Rightarrow \alpha < \gamma$
- (5) $\pi/[L \rightarrow \{\}]$
- (6) $\alpha/[L \rightarrow \{\beta\}] \wedge \beta < \gamma \Rightarrow \alpha/[L \rightarrow \{\gamma\}]$
- (7) $\alpha < (\beta \cdot \gamma) \Rightarrow \alpha < \beta \wedge \alpha < \gamma$
- (8) $\alpha/[L \rightarrow \{\beta_1, \dots, \beta_n\}] \Rightarrow \alpha/[L \rightarrow \{\beta_1\}] \wedge \dots \wedge \alpha/[L \rightarrow \{\beta_n\}]$
- (9) $\alpha < t[ap_1, \dots, ap_n] \Rightarrow \alpha < t \wedge \alpha/[ap_1] \wedge \dots \wedge \alpha/[ap_n]$
- (10) $\alpha < \beta \Rightarrow L(\alpha) < L(\beta)$
- (11) $L(\alpha) < L(\pi[L \rightarrow \{\pi\}] \cdot \alpha)$
- (12) $\alpha/[L \rightarrow \{\beta\}] \Rightarrow \alpha/[L \rightarrow \{L(\alpha) \cdot \beta\}]$
- (13) $\alpha < L(\beta) \Rightarrow \alpha < L(\pi[L \rightarrow \{\alpha\}] \cdot \beta)$
- (14) $\alpha/[L \rightarrow \beta] \Rightarrow L(\alpha) < \beta \wedge \alpha/[L \rightarrow \{\beta\}]$
- (15) $L(\alpha) < \beta \wedge \alpha/[L \rightarrow \{\gamma\}] \Rightarrow \alpha/[L \rightarrow \beta]$
- (16) $L(\alpha) < \beta \Rightarrow (\pi[L \rightarrow \{\pi\}] \cdot \alpha)/[L \rightarrow \beta]$
- (17) $En(t)$
- (18) $\alpha/[L \rightarrow \{\beta\}] \wedge En(\alpha) \Rightarrow En(\beta)$
- (19) $En(L(\alpha)) \Rightarrow En(\alpha)$
- (20) $En(\alpha) \wedge \alpha < \beta \Rightarrow En(\beta)$
- (21) $\sim En(\alpha) \Rightarrow \alpha < \beta$
- (22) $\pi/[L \leftarrow \{\}]$
- (23) $\alpha/[L \leftarrow \{\beta\}] \wedge \gamma < \beta \Rightarrow \alpha/[L \leftarrow \{\gamma\}]$
- (24) $\alpha/[L \leftarrow \{\beta_1, \dots, \beta_n\}] \Rightarrow \alpha/[L \leftarrow \{\beta_1\}] \wedge \dots \wedge \alpha/[L \leftarrow \{\beta_n\}]$
- (25) $En(\alpha) \wedge \alpha/[L \leftarrow \{\beta\}] \Rightarrow \beta < L(\alpha)$
- (26) $En(\beta) \wedge \alpha/[L \leftarrow \{\beta\}] \Rightarrow \alpha/[L \rightarrow \{\beta\}]$
- (27) $\sim En(\alpha) \Rightarrow \pi/[L \leftarrow \{\alpha\}]$

<推論規則>

P と $P \Rightarrow Q$ から Q を導く。

<定理>

公理あるいは定理に推論規則を有限回適用して得られる論理式。

推論規則と定理は命題論理と同様であり、フレーム構造論理の推論は決定可能である。体系 FS1の特徴は、さまざまな属性関係を記述できることである。

5 フレーム構造論理によるメタデータ記述

本節では、XML で記述されたメタデータを、フレーム構造論理の論理式の一部に変換する。DTD で定義された社員名簿を記述するためのサブセットを例として取り上げ、体系 FS1のオブジェクトへの変換例を示す。

5.1 DTD による文書型定義

DTD は XML のスキーマ言語である。XML は、現実の世界を階層構造にモデル化して、それを要素や属性の階層構造で表現する。以下に社員名簿を記述する言語の DTD による作成例を示す。社員の情報として、社員番号、名前、年齢、所属部署、所有資格を管理するものとする。

```
<! DOCTYPE 社員名簿 [
  <!ELEMENT 社員名簿 (社員+)>
  <!ELEMENT 社員 (社員番号, 年齢, 所属部署, 所有資格*)>
  <!ELEMENT 社員番号 (# PCDATA)>
  <!ELEMENT 年齢 (# PCDATA)>
  <!ELEMENT 所属部署 (所属部, 所属課)>
  <!ELEMENT 所属部 (# PCDATA)>
  <!ELEMENT 所属課 (# PCDATA)>
  <!ELEMENT 所有資格 (資格名, 取得日)>
  <!ELEMENT 資格名 (# PCDATA)>
  <!ELEMENT 取得日 (# PCDATA)>
]>
```

DTD では階層構造が記述される。上の例では、社員番号、年齢、所属部署、所有資格の各要素の親要素は社員要素であり、所属部署には、所属部と所属課要素が子供要素としてあるという情報が記述されている。また、一人の社員に対して、社員番号は一つに定まり、資格は、全くもっていなかったり、複数持っていたり、まちまちであるという情報も記述されている。

5.2 XML による文書記述

XML での文書記述の具体例をあげる。DTD で5.1の例のようにサブセット言語を定義したとする。このサブセット言語を用いて、社員番号7番で、年齢が34歳、所属部署が総務部財政課で、シスアドを2000年、情報処理技術者を2004年に取得したような社員は、XML 文書とし

て以下のように記述される。

```
<社員>
  <社員番号>7</社員番号>
  <年齢>34</年齢>
  <所属部署>
    <所属部>総務部</所属部>
    <所属課>財政課</所属課>
  </所属部署>
  <所有資格>
    <資格名>シスアド</資格名>
    <取得年>2000</取得年>
    <資格名>情報処理技術者</資格名>
    <取得年>2004</取得年>
  </所有資格>
</社員>
```

上記の例で、「所有資格」の要素はDTDで「資格所有*」と定義されており、複数個の要素が許されている。上記の例の場合は、要素が二つであるが、何個（0個でもよい）要素があってもよいことになっている。

5.3 体系FS1によるメタデータの記述

XMLでマークアップされた文書から、4節で示したフレーム構造論理への変換は容易である。例えば、5.2のXMLでの文書記述は、体系FS1で以下のように変換できる。

```
社員[社員番号→7,
  年齢→34,
  所属部署→ $\pi$ [所属部→総務部, 所属課→財務課],
  所有資格→{ $\pi$ [資格名→シスアド, 取得年→2000],
     $\pi$ [資格名→情報処理技術者, 取得年→2004]}
]
```

FS1では、上記のひとまとまりが一個のオブジェクト α に相当する。体系FS1のオブジェクト関係は、「 $\text{En}(\alpha)$ 」と「 $\alpha < \beta$ 」の二種類あるが、XMLからの変換の場合、 $\text{En}(\alpha)$ は必ず成り立っているといえる。しかし、 $\text{En}(\alpha)$ だけでは有意義な推論はできない。「 $\alpha < \beta$ 」の二つのオブジェクト間の関係は、XMLからは一般的には得ることができない。このオブジェクト

関係を得るためには、図1のオントロジ層，すなわちOWLで記述された知識が必要である。

5.4 変換の容易性の考察

XML文書からフレーム構造論理への変換は容易である。その理由として、以下の三点があげられる。

- ①フレーム構造とDTDの親和性
- ②フレーム構造論理は変数を持たない
- ③フレーム構造論理の属性関係の表現能力

①は、DTDは階層構造の記述ルールを定義する言語であり、記述された階層構造と、フレーム構造のフレームが本質的には同じ物であるということである。②に関しては、公理図式からわかるように、フレーム構造論理は一切の個体変数を持たない。自然言語の文書中には変数は現れないため、一階述語論理のような変数が論理式に含まれるような体系への自動変換は難しい。この自動変換の難しさは、述語論理をベースとした自然言語の推論システムが今日まで実現されない理由の一つになっている。フレーム構造論理は、元々、自然言語からの自動変換を目的とした論理であるため一切の個体変数を持たない。マークアップされた文書も変数を持たないため、XMLからフレーム構造論理への変換は比較的容易である。③は、フレーム構造論理では、一つしか属性値がない属性と、いくつもの属性値をとる属性を、属性対「 $L \rightarrow \alpha$ 」と「 $L \rightarrow \{\alpha_1, \dots, \alpha_n\}$ 」をそれぞれ用いて、簡単に区別できることによる。もちろん同様の属性関係の論理式への変換は、一階述語論理でも不可能ではないが、人間にとっても難しい変換となる。

6 まとめ

本論文で、フレーム構造論理によるメタデータ記述の一例が示された。また、XML文書からの論理式への変換が、一階述語論理と比較して容易であることが示された。フレーム構造論理による記述のメリットは、論理式を記述した時点で、推論まで可能であるという点にある。すなわち、図1のロジック層まで記述できていることになる。しかし、オントロジ層を参照しないと、推論の基礎であるオブジェクト関係を表現できない。従って、フレーム構造論理は、自動変換を目的とするならば、OWDLの代わりに使え、人間が記述するならば、OWLの代わりに使える。

参考文献

- (1) 兼岩 憲, 佐藤 健: D1: Description Logics, 人工知能学会誌, Vol. 18, No. 1, pp.73-82 (2003).
- (2) <http://www.w3.org/TR/2000/REC-xml-20001006>
- (3) 浦本 直彦: Webにおける情報統合, 情報処理, Vol. 44, No. 7, pp.707-712 (2003).
- (4) Lassila, O. and Swick, R.: Resource Description Framework (RDF) Model and Syntax Specification, Technical report, W3C(1999).

- (5) Brickley, D. and Guha, R.V. : Resource Description Framework Schema (RDF/S) Specification 1.0, Technical report, W3C (2000).
- (6) Patel-Schneider, P.F., Harrocks, I. and Harmelen, van F. : Proposed OWL Knowledge Base Language (2002).
- (7) Eberhart, A. : Survey of RDF data on the Web, <http://www.i-u.de/schools/eberhart>
- (8) Dublin Core Metadata Element Set, Version 1.1: Reference Description, 1999, DCMI Recommendation, <http://dublincore.org/documents/dces>
- (9) http://www.ijnet.or.jp/edr/J_index.html
- (10) Lenat, D.B. : CYC a large-scale investment in knowledge infrastructure, Communications of the ACM, Vol. 38, No. 11, pp.33–38 (1995).
- (11) 小松香爾, 西原典孝, 堀越浩司 : 多様な属性関係を表現できるフレーム構造論理の拡張体系, 第12回情報基礎論シンポジウム (2000).
- (12) Nishihara, N., Komatsu, K. and Yokoyama, S. : Frame-Structure Logic capable of structural knowledge representation, 平成10~12年度科学研究費, 基礎研究(C)(2), No.10680369.