

このドキュメントは

Why RDF model is different from the XML model

<http://www.w3.org/DesignIssues/RDF-XML.html>

の和訳です。

この文書には和訳上の誤りがありえます。

内容の保証はいたしかねますので、必ず W3C Web サイトの正式版文書を参照して下さい。

RDF モデルと XML モデルとの違い

この記事は、次の問いに答えることを試みている。

「何故、私が RDF を用い、何故、XML を用いないのか？」

この問い掛けは、RDF が始まってから色々な方面から、行われている。

W3C Query Language workshop において、文書を検索したい人と、色々な形式の文書から意味を抽出したり検索したりしたい人との間で、明らかな見解の違いがあった。

これは良くあるケースである。

私は、この記事で「XML モデルでなく RDF モデルが使われたか」をうまく説明できなかったため書いた。

この記事を書いた後、未だ XML モデルを指向している人がいると聞いたので、別の記事”Using XML for Data”を書いた。

この記事中においては、XML 全体を単純化して XML データモデルと記述し、RDF モデルおよびシンタックスを単純化して RDF シンタックスと記述する。

それは、RDF モデルを XML モデルにマッピングすることではなく、XML を用いて RDF モデルを表現することである。

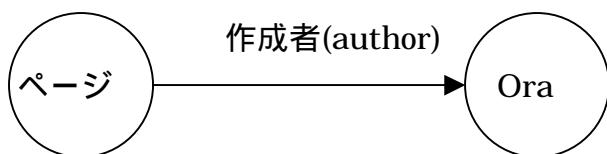
ここで、RDF アサーションの典型的な例を見てみよう。

例えば、“このページの作成者は Ora である”とする。

これを三元表示(トリプル)で表すと、

triple(author, page, Ora)

となり、これを図で表すと、次の様になる



これを XML で表記すると、次の様に記述できる。

```
<author>
  <uri>page</uri>
  <name>Ora</name>
</author>
```

または、

```
<document href="page">
  <author>Ora</author>
</document>
```

または、

```
<document>
  <details>
    <uri>href="page"</uri>
    <author>
      <name>Ora</name>
    </author>
  </details>
</document>
```

または、

```
<document>
  <author>
    <uri>href="page"</uri>
    <details>
      <name>Ora</name>
    </details>
  </author>
</document>
<document href="http://www.w3.org/test/page" author="Ora" />
```

XML グラフ

上記の記述は、何れも完全に正しい XML 文書であり、人間は同じ意味を表しているとして理解できる。これらを機械に解析させると、機械は異なる XML ツリーを生成する。

次の XML ツリーを仮定した場合、

```
<v>
  <x>
    <y a="ppppp">/y>
    <z>
      <w>qqqqq</w>
    </z>
  </x>
</v>
```

これは、何を意味するのか分からないが、人間がこれを読む場合、エレメント名がヒントを与えてくれる。

スキーマなしで見た場合、文書の構造については知ることが出来るが、それ以上何も知ることが出来ないし、何の推測をすることも出来ない。

ppppp が qqqqq の y なのか、qqqqq が ppppp の z なのか、もしくは、他の何かか知ることが出来ないし、何が問題なのか問うことも出来ない。

この xyz 例における混乱の原因は、沢山の疑問が生じることにある。

次の疑問がある。

- ・ ある詳細エレメント中に w エレメントが存在するか？
- ・ 最初の x エレメント中の w エレメントの内容は何か？
- ・ 属性値"ppppp"を有する x エレメントを含んでいる最初の y エレメントの次の w エレメントの内容は何か？
- ・ 等々

これら総ては、上記の文書に關対し生じる疑問である。

もしも、上記文書のスキーマが分かっているとして(この点が重要)、そして、もしも、そのスキーマが同一のものを表現する為の或る限定的な方法を示しているならば(この点も重要)、上記の疑問は、例えば、次の疑問とする事ができる。

- ・ ページの作成者は誰か？

これなら、明瞭であり、XML 文書からセマンティックグラフに写像できる。簡単に言うと、次の様に言えることは明らかである。

- ・ その写像は、多くから一つに写像する。
- ・ どのような写像か知るためのスキーマが必要。
- ・ (そのスキーマは、XML に関するものであり、これから付け加えられるであろう言語に関するものではない。)
- ・ XML ツリーの中の何かを検索するための表現は、RDF ツリーの中の何かを検索するのに必要な表現より複雑なものが必要とされる。

この最後のものが、特に重要である。

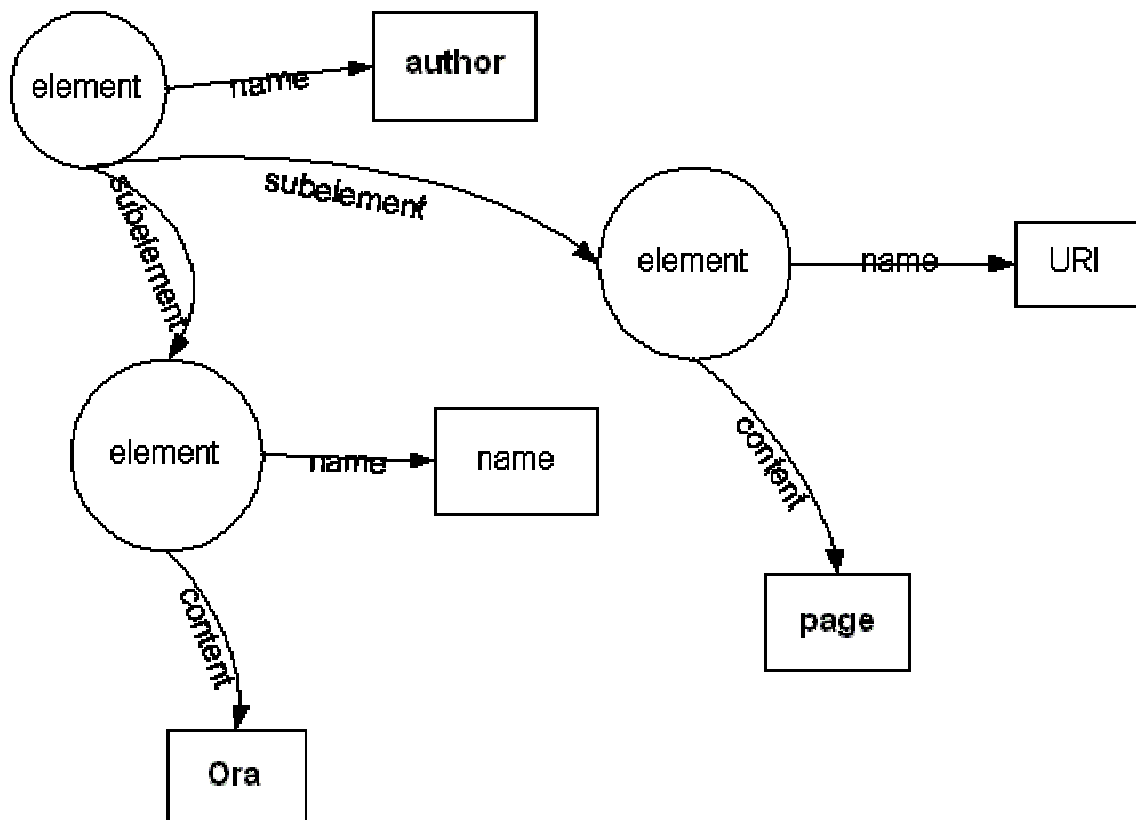
もしも、XML スキーマで示される文書の作成者を検索するための表現を作る場合、それを作ることは可能であり、それは、簡単なものかも知れないし、簡単でないかも知れない。しかし、もしも、他の一つ以上の属性の検索の為の表現を単一の表現に統合しようとするならば、(例えば、ある作者の本の一覧を示そうとする場合、) XML の場合、思う以上に酷く不恰好になる。

(これは、文字列に対する正規表現操作に数字を追加定義する場合を想像してもらえば良い、それを追加することは出来る、しかし、それを繰り返し表現によって、行うことは馬鹿らしいことである。この問題を解決するためには、分離した型として数字を取り扱うことを止めることである。)

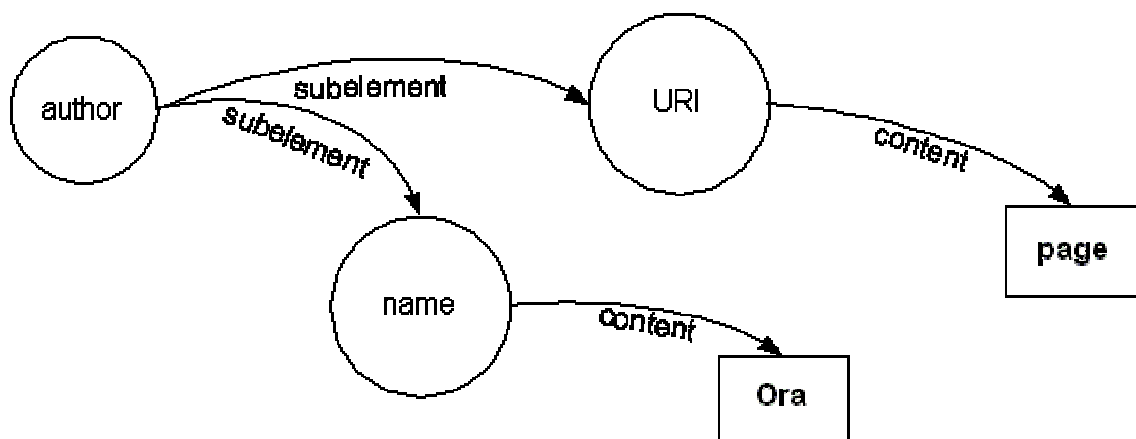
前記の簡単な XML エンコーディングを眺めて見よう。

```
<author>
  <uri>page</uri>
  <name>Ora</name>
</author>
```

これは、次の図によって示すことができる。



円の中にエレメントの名前を書けば、上記の図は、次の様に簡潔なもの
とすることができる。



これを RDF ツリーで表すと次の様になる。



この図を、各部分に対応する URI で置き換えた図にすることができる。

XML ツリーの検索が複雑になる原因は、一般に、XML を特定の論理ツリーに写像する方法に多くの方法があることであり、更に、検索が、それらの方法の選択とは、無関係であることである。

検索の大部分は、基本的に、ある事実に関する可能な総ての表現の集合をある一つのステートメントに変換しようとすることであり、これは、正に RDF が実現しようとしていることである。

すなわち、それは、ある文書に対して行いたいことを、ステートメントを書く標準的な方法を提供することであり、RDF の用語を用いて、同じ効果を生じさせることが出来、同等の RDF ツリーは、幾つかの XML ツリーによって得られる結果と同一の結果をもたらす。

では、もしも、XML にラベル付けすることが出来、特定のパーサがそれを読むならば、そのパーサは、アサーション(三元表記 ; triples)を知ることが出来、また、主語と目的語を区別することが出来るのではないか、すなわち、RDF を使わなくても、論理アサーションを推測できるのではないかと、言うであろう。

これは正に、RDF が目指していることである。

RDF グラフ

実際の所、RDF は非常な柔軟性を有している。 - RDF は、XML では多くの方法で表すことの出来る三元表示を、只一つの方法で記述でき、特定のアプリケーションに適合させることができる。

前記のものを次の様に記述できる。

```
<Description about="http://www.w3.org/test/page" Author ="Ora" />
```

ネームスペースの定義は省略してある。

実際には、誰もが自分の動詞、主語および目的語を分散したウェブの中に作り所有でき、そして、それらの任意の用語は、URI によって識別される。

次は、実際に用いられる実例である。

```
<?xml version="1.0"?>
```

```
<Description
```

```
xmlns="http://www.w3.org/TR/WD-rdf-syntax#"
```

```
xmlns:s="http://docs.r.us.com/bibliography-info/"
```

```
about="http://www.w3.org/test/page"
```

```
s:Author = "http://www.w3.org/staff/Ora" />
```

RDF エLEMENTの Description は、パーサに対し、そのデータの後に続く主語や目的語や動詞を見つける手掛かりを与える。

これは、XML 中に基本 RDF を用いる簡便な方法である。

これ以外に、より長くより効果的に記述する方法もある。例えば、同一のオブジェクトに沢山の属性集合を与える場合などである。

この場合、有益なことは、これら総てが、同一の三元表記を意味することである。



或る文書に関し、その文書が何意味しているかに付いて問うことを試みることを問う時、それは、複雑になる。

それは或る方法で可能であろう。しかし、文法に明示的にフラグ付けすること(RDF 構文はこれを実現する一つの方法である。)は、全体としては非常に望ましい。

RDF では可能だが、XML では不可能なことに、次のものがある。

セマンティック ツリーを解析でき、最終的に三元表記の集合を得ることが出来る、また、理解できない部分を無視することもできる。

上記の構造を理解したとすると、次の問題が内在していることが分かる。

- ・ スキーマを導入する事無く、または、特別な文書タイプを識別する手作りプログラムのアプリケーションも無いならば、文書から如何なるセマンティック情報を取り出すことが出来ない。
- ・ XML スキーマが変わる時、一般には、新たな中間要素(上記のツリーにおける "details" のようなものや、HTML における "div" など)を導入する。この場合、上記の文書の構造に基づいて行われる任意のクエリーを正しく処理

できるかも知れないし、出来ないかも知れない。

- ・ セマンティックモデルを作らない場合、適切に定義された物を得ることが出来ない。

最後の問題に関し後で、幾つかの例を示す、明らかに、RDF を用いなくても、たとえ XML システムであっても、慎重な設計でこれらの問題を回避することが出来る。しかし、RDF を用いれば、これらをより簡単に行うことが可能である。

正しく理解したか

「セマンティックモデルを作らないで問題に対処しようとした場合、適切に定義されたものを得ることが出来ない。」とは、何を意味するか？

実際に発生しうる曖昧な幾つかの事例を考えて見よう。

RDF において、何かについて述べている何かを理解しようとした場合、より深い考察を必要とすることがあるが、それは意図的なものではない。

“期限 1999 年” と書かれているジャムのジャーのラベルを見た時、何の期限を意味しているのか、それはラベルの期限なのか？ジャムの期限なのか？

或る文書についての記述に関する記述と或る文書に関するステートメントとの間に曖昧性がある。

他の事例は、他の要素を修飾する要素である。

情報が多くのレコードの中に別々に入れられた一つの集合から構成される場合、しばしば、論理性の欠如により、曖昧性が発生する。

HTTP ヘッダー(又は、email ヘッダー)が、良い例である。

これらは、只一つのプログラムが総てのレコードを操作する時は、うまく動作することが出来るが、しかし、混在レコードの場合、問題を生じる。

XML の場合、2つの要素の持つ問題の罠に簡単に陥る。その要素の一つが、著者を記述し、且つ、他の一つが「”著者”要素が直接著者を意味せずに、翻訳者を意味している。」を意味しているフラグの場合、唐突だが、最終文書の著者が最後であると結論付けることを可能にする”著者”タグは、その文書上の他のどれかの要素によって無効にさせられうる。

現実のセマンティックスが最初に見たのと同様に明らかでない仕様書の他の症状は、順序性である。

「レコードの集合の順序が重要であるが、それぞれのレコードが独立に定義されている」と我々が聞いた時、どう出来るであろうか？独立なアサーションは独立性を有する、または、ある順序を持っていることにおいて、常に正しい。例えば、或るサーバのコンフィギュレーションファイルの場合、「誰もが、その

ページにアクセス出来る」と言うようなステートメントは、実際は次のことを意味しているかも知れない。

「そのコンフィギュレーションファイルの中に当該ページに対する他の制約が無い場合、誰もがそのページにアクセス出来る。」

これは、仕様書に記述しない、しかし、規則は順番に処理されるということを示す。現実的には、論理記述は大規模な入れ子構造で表現される。

暗黙の順序性も存在する、例えば、メールヘッダーが、「このメールは暗号化されている」、「このメッセージは圧縮されている」、「このメッセージは ASCII コードで符号化されている」、「このメッセージは、HTML データである」等を示している場合。実際のメッセージは、HTML データの圧縮データの暗号データの ASCII 符号化データであり、メールヘッダーの中におけるこの論理は、仕様書の中に明記されるべきである。

文書の中の順序

機械にある知識ツリーを与えることと或る人にある文書を与えることとの間には幾つかの基本的な違いが存在する。

人の為の文書は一般にシリアライズされている、従って、人間が順番に読む時、その結果は人間の頭の中に構成要素のグラフとして組み立てられる。その順序は重要である。

知識グラフにおいて、異なるステートメント間で共通のノードが一貫性を持って識別される場合、順序は重要でない。(順序だてられたリストの概念は、存在する、その概念は、RDF においても重要であるが、もしも、例えば、「L の最初の要素が x であり」、「L の第三要素が z であり」と言う具合に、ステートメントの非順序集合まで細かいレベルまでブレイクダウンされるならば、最下位レベルでは、順序の必要性はなくなる。)

マシンリーダブルの文書の中で、順序が重要であると言われている表面的な独立ステートメントの表は、しばしば、順序性が需要でないステートメントになることが判明している。

幾人かは、RDF を使うことを考えるのに抵抗している、何故なら、彼らは、順序性を捨てることを望んでいない。しかし、私の仮説では、これは、ヒューマンリーダブル文書処理の上での制約から生じたものである。

これらの文書は RDF に変換するのに適していない典型的なものである。

結論

時々、セマンティックウェブはグラフだけのものであると考えている人々の集団が見られる、また、文書ツリー(または、リンクを含んでいる場合、グラフ)だけを考えている他の人々の集団がいる。しかし、その違いを認識することが

重要である。

このシリーズには、次のものがある。

- What the Semantic Web is not – 納得させるのが難しい質問に対する回答。
<http://www.w3.org/DesignIssues/RDFnot.html>
- Evolvability: 技術進化の為に必要となる言語の属性
<http://www.w3.org/DesignIssues/Evolution.html>
- Web Architecture from 50,000 feet
<http://www.w3.org/DesignIssues/Architecture.html>

以上、