

# セマンティックウェブ・プロセッサ開発マニフェスト

## A Manifesto of Semantic Web Processor Development

小出 誠二  
Seiji Koide

株式会社ギャラクシーエクスプレス  
Galaxy Express Corporation

In the last year, we have developed a prototype for the feasibility study of "a support system for large-scale operation systems using IT". The prototype demonstrated a scenario that the support system monitors behavior of an operation system, detects an anomaly, diagnoses the system, and suggests counteractions to operators. In the prototype, every subsystem was built as web services and invoked by a supervisor system that was programmed according to the demonstration scenario. In a couple of years, we are going to establish the practical system that dynamically discovers appropriate web services, and invokes them in dynamic context of operation using the Semantic Webs. In order to utilize the Semantic Web technology for our goal, we intend to develop a semantic web processor. This paper is a development manifesto for our semantic web processor. This project is prepared with the contract under the Japanese IT program of the Ministry of Education, Culture, Sports, and Technology.

### 1. はじめに

文科省 IT プロジェクト「IT を活用した大規模運用システムの支援システムの構築」[小出 2002][koide 2003]にて、ロケット打上を題材として、大規模運用システムの支援システムを開発している。このプロジェクトでは、全体システムを統合するための分散協調の枠組みとして、ウェブサービスとセマンティックウェブの活用を考えている。平成 14 年度研究「システム試作」では、作成した一つの不具合シナリオに基づいてシステムデモンストレーションを実施することを目的に、ウェブサービスとして各サブシステムを開発し、全体システムを統合して不具合監視、検知、診断、対応操作導出、オントロジによるメッセージ生成、および運用者によるマルチメディアデータ検索のデモンストレーションを行った。平成 15, 16 年度では、ロケット打上システムの運用タスク分析、要求分析から始め、本格システムを設計製作するが、ここではウェブサービスを静的に呼び出すのではなく、状況に対応して適切なウェブサービスを動的に発見し、複数のウェブサービスを動的に組み立てるしくみを開発する。それを可能にするための技術として DAML-S(Semantic Markup Language for Web Services)に代表される各種セマンティックウェブ技術を用いるつもりである。

セマンティックウェブを処理するためには、そのためのプロセッサが必要となる。すなわち、RDF (Resource Description Framework), RDFS (RDF Schema), OWL (Web Ontology Language), DAML-S などの表記を解釈処理するためのプロセッサが必要である。本報告では、リスブ言語を前提としたときのこれらプロセッサ開発に重要と考えられる技術開発ポイントと今後の開発計画について述べる。

### 2. Unicode および URI

ティム・バーナーズ＝リーによるセマンティックウェブ・タワーでは、その最下層はグローバルな文字識別のための Unicode とグローバルなウェブリソースの唯一性を可能にする URI(Uniform Resource Indicator)になっている。幸いにして市販のリスブ処理

系においてすでにその内部文字表現は Unicode になっており、外部表現としての JIS コードあるいは Shift-JIS への変換も自動的になされるかあるいはそのためのライブラリが用意されている。URI についても同様で、URI のためのライブラリが整理されてきており、セマンティックウェブ・プロセッサ開発にはこれらを有効に用いることができる。

### 3. XML および名前空間

XML は汎用を目的とした機械可読な記法であるが、その本質は開始タグと終了タグをセットとするエレメントの入れ子構造により、ものごとを表現することである。そのテキスト表現がリスブのS式と類似であることから、リスブで XML 処理系を開発することは比較的容易と思われるが、残念ながら現在満足できるような XML 処理系がリスブにはない。たとえば、FIPA 仕様では、ACL(Agent Communication Language)中に RDF コンテンツを記述しようとする、図1のようにS式と XML 表記をともに扱うことが必要となる[FIPA XC00011B]。

リスブを前提とすれば、S式パーザと XML パーザを共存させるために XML パーザを開発するだけでよい。しかもリスブのリードマクロ機能を用いれば、山形括弧を読み込むと同時にその開始タグからそれに対応する終了タグ読み込みまでを XML パーザで処理して、終了タグ読み込み完了とともにリスブのリーダーに戻るというようなことも可能となる。すなわち、リスブのトップレベルで図1に示すようなタイプ入力を実行することも、またリスブ処理系で XML 表現のみを入力することも可能となる。

XML の処理系として DOM や SAX があるが、ここでは XML 一般のパーザではなく、RDF や OWL の処理に特化した XML パーザを考え、XML エレメントの入れ子構造を DOM のようなノードの木構造ではなく、rdf:RDF 構造体や owl:Ontology 構造体と、その内部スロットに格納される rdf:Description 構造体や owl:Restriction 構造体などの入れ子構造として内部表現する。こうすることにより、これら構造体の印刷刷数をカスタマイズすることにより、内部表現と外部表現を一致させ、XML 表記の入力が完了すると内部にその内部表現が実現され、その実現によって印刷されるものは入力したとおりの XML 表記とすることがリスブのトップレベル上でも可能となる。

連絡先:小出誠二, (株)ギャラクシーエクスプレス, 港区浜松町  
1-18-16, Fax03-5733-7190, koide@galaxy-express.co.jp

```

(request
:sender Mary
:receiver John
:content (
  <?xml version="1.0"?>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:fipa="http://www.fipa.org/schemas#">

    <fipa:Action rdf:ID="JohnAction1">
      <fipa:actor>John</rdf:actor>
      <fipa:act>open</rdf:act>
      <fipa:argument>
        <rdf:bag>
          <rdf:li>door1</rdf:li>
          <rdf:li>door2</rdf:li>
        </rdf:bag>
      </fipa:argument>
    </fipa:Action>
  </rdf:RDF>
):language fipa-rdf0)

```

図1 RDFコンテンツのFIPA ACL

名前空間はセマンティックウェブのグラウンディングにとって非常に重要な機能である。すなわち、グローバルな URI の唯一性の特徴を利用して、XML タグのグローバルな唯一性を保証している。一方、リスブのシンボルに関する唯一性のための仕組みとしてパッケージがある。ここでパッケージとは Java のそれとは異なり、文字列からそれをシンボル名とするシンボルを生成するときに用いられる空間である。幸いなことに、リスブシンボルをエクスポートすると見かけ上 QName のように「galex:foo」などと印刷されて見やすい。そこでリスブパッケージの document 属性に URI のフラグメントを除いた情報をセットしておき、URI 入力の際に該当パッケージ内にて正しくリスブシンボルをインターンすることで URI からリスブシンボルへのマッピングを行う。

XML のスキーマ機能は一部のソフトウェア技術者にとって重要かもしれないが、一方あまりにも膨大な仕様であることからその完全な実装が難しいという批判もある。セマンティックウェブ開発にとって XML スキーマ開発は必要ではあるがそれほど決定的ではない。とりあえずの処置としてリスブのプリミティブデータを処理可能とするだけとする。

#### 4. RDF および RDFS

RDF はセマンティックウェブ実現のフレームワークとなる基礎の表現であり[Lassila 1999][Hjelm 2001][Brickley 2003], RDFS はオントロジ記述の基礎となるボキャブラリ定義である[Brickley 2002][Brickley 2003]. RDF の数学的な根拠はモデル論としてははっきりしているが[Hayes 2001][Hayes 2003], 一方そのモデル論の厳密さに比べて仕様に忠実な処理系として満足できるものは少ない。特にリスブ言語の処理系は見あたらない。

既に Schank らの事例ベース推論 MOP(Memory Organization Package)を用いて RDF および RDFS 処理系を開発した[小出 2003]. ここで処理系と呼んでいるのは、新しいリソースや新しいプロパティを定義してオントロジを開発でき、`rdfs:subClassOf` の推移律や属性値の継承機能を利用することができるという意味である。ただし、XML パーザはなく、MOP の記述方法に従って記述する。ここでの実現上の問題点はメタクラスである `rdfs:Class` および `rdfs:Datatype` のクラス定義におけるサーキュレーションと、プロパティ `rdfs:isDefinedBy` および `rdfs:member` のインスタンスでありながら、`rdfs:subPropertyOf` 関係というプロパティの派生クラスのような性質を持つところであった。今回の MOP による実装ではクラス定義サーキュレーションはなしとし、`rdfs:isDefinedBy` および `rdfs:member` はインスタンスではなくプロパティクラスとしたが、仕様に厳密という意味

では改善されるべき点である。詳細は[小出 2003]を参照されたい。

RDFS ではプロパティについて定義域と値域を定義できる。歴史的には Schema Representation Language (SRL)[Wright 1983]にて実現されていたものであるが、その機能についてはフレーム知識表現の立場からはスロット制約、すなわち定義域はそのプロパティを用いることのできるクラスを指定し、値域はそのプロパティの値が取り得る範囲を指定するという解釈と、モデル論の立場からはオントロジ生成、すなわちステートメントとしてあるプロパティに関係づけられたサブジェクトとオブジェクトのインスタンスを入力すると、インスタンスはそのプロパティの定義域と値域をクラスとするという知識を生成するという解釈がある。前述の RDF/S 処理系では前者の解釈を実装したが、後者についても考慮する必要がある。

#### 5. N-triple および Notation 3

RDF や RDFS は XML でよく表記されるが、RDF の本質はサブジェクト、プレディケイト(プロパティ)、オブジェクトの三つ組みで知識表現するところであり、XML 表記よりも本来 N-triple[Hayes 2001][Grant 2003]や Notation 3[TBL 1998][TBL 2000]で表記した方が自然である。そこで N-triple パーザを開発した。このパーザは N-triple 表記のファイルを読み込み、3節に述べたようにリスブシンボルにマッピングする。

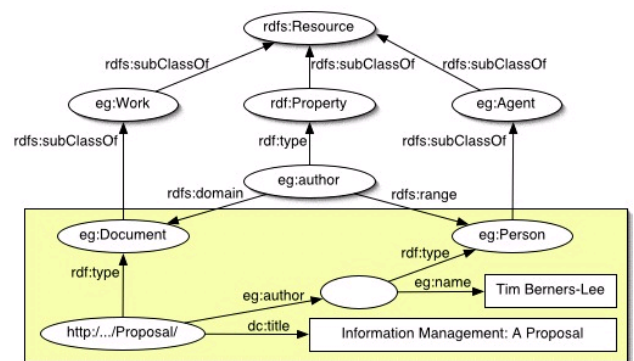


図2 RDFS の実例

図2は RDFS 仕様に関する古いドキュメント[Brickley2002]に記載されていた RDFS の例であるが、これを N-triple パーザで表記すると最後のページに記載したようになる。ただし、元々の N-triple の表記法ではリソースはすべて URI として山形括弧でくくられ、表記されねばならないが、これではあまりにも人にとって煩雑でありかつ非常に見にくい。そこで XML と同様に名前空間を導入し、XML 文法表現における QName を使ってよいことにした。このように拡張された N-triple 文法を最後のページに併記した。

人が断片的な知識を思いつづま三つ組みで表現することは、比較的容易であるが、それが全体として整合性あるとは限らない。一方、機械がモデル論に従って、三つ組みとして入力された知識の関係を整理し、矛盾や不備を指摘することは容易と考えられる。前述の URI からリスブシンボルへのマッピングにより、ばらばらに入力された三つ組みにおいて、同一の URI は同一のシンボルとなることから、リスブ内部でこれを有向グラフとして取り扱うことは可能である。さらに、三つ組みのプレディケイトを Prolog のそれと位置づけ、一つの三つ組み表現を Prolog における一つの事実と見なすことができる。これまでに、簡単な Prolog 処理系を実装し、ユニフィケーションによる入力データに

対するクエリを可能にしたが、`rdf:type`と`rdfs:subClassOf`の推移律がなく、これを実装することが課題となっている。

RDF と関係データベースの関係についてティム・バーナーズ＝リーも論じているが、三つ組み入力データの永続化にからんで関係データベースの利用を考えてもよい。その場合は効率の点から一つの大きなデータベースに収めるのではなく、プレディケイトに着目して、複数の副データベースに効率よく格納することが望ましく、採用する関係データベースと実装方法に注意が必要である。

Notation 3(N3)は N-triple に名前空間による QName 表記を許し、サブジェクトおよびオブジェクトに関して複数三つ組みの重複文を可能にしたものである。原理的には開発済み N-triple パーザを N3 用に拡張するのは容易ではあるが、3節に述べた QName からリスプシンボルへのマッピング方法にしたがうと、N3 における名前空間プレフィックスなしの表現(コロン+ローカル名)が、リスプ上ではキーワードとなってしまうことが悩ましい。

なお、XML 表記から RDF のモデル生成方法については資料[神崎]に詳しい。

## 6. OWL と DL

OWL には、OWL Lite、OWL DL、OWL Full の三階層があるが、Lite と言えどもかなりの機能実装が必要である。図3に OWL Lite および OWL DL のクラス階層を示す。前述の開発した RDF/S 処理系では CLOS(Common Lisp Object System)のクラス・サブクラス関係を `rdfs:subClassOf` 関係にマッピングし、`rdfs:subPropertyOf` については省略して `rdfs:subClassOf` と同様にクラス・サブクラス関係にマッピングしたが、OWL に取り組む前提としてこれらを共に実装する必要がある。現在の延長上で考えれば、CLOS のメタオブジェクトプロトコルを用いて、`rdf:Property` を抽象メタクラス `standard-slot-definition` のサブクラスとし、`rdfs:subPropertyOf` を特殊なクラス・サブクラス関係として実装する手法が考えられる。

`owl:TransitiveProperty` は `rdfs:subClassOf` や `rdfs:subPropertyOf` のような推移律を持つプロパティを定義するためのプロパティクラスである。これも CLOS のメタオブジェクトプロトコルを用いて実装できるかも知れない。詳細に検討する余地がある。

`owl:allValuesFrom` はある属性についてインスタンスの取り得る値を制限したクラスをローカルに指定するためのものであるが、これは使い方は少し異なるが、Schank らの MOP で実装されていたものである。`owl:allValuesFrom` は全称記号(for-all)と類似であるとの説明がある。

`owl:someValuesFrom` はある属性についてインスタンスの取り得る値についての少なくとも一つはそうであることをローカルに指定するためのものである。これは存在記号と類似であるとの説明がある。

OWL DL はその背景に Description Logic(DL)を持つ。DL については[兼岩]に解説記事があり、その実装法の一つであるタブロー法については[長谷川]に解説記事がある。タブロー法によれば他の方法よりも OWL DL への適用が容易と思われるが、OWL DL の実装においてどれほどの推論機能が必要とされるか、またどのような実装方法が望ましいか見極めが必要である。一方、CLOS ではクラス・サブクラス関係は `initialize-instance` されたときにクラス優先順位リストが計算されるが、これは DL において概念間の包摂関係を計算していることと同じである。これを極端に拡張すると、RDF のモデル生成[Lassila 2002]で行われたように、OWL において知識が入力されるたびに、推論規則により必要なクロージャを生成しておけばよい。課題は OWL DL における完備な推論規則を明らかにすること( RDF グラフモデルへのマッピングについては <http://www.w3.org/TR/owl-absyn/mapping.html> に詳しいが)、大規模な実用規模のオントロジにおいて、どのような実装方法が大規模ストレージと計算効率上望ましいのか明らかにすることである。CLOS を用いた場合にはその永続化においてオブジェクトデータベースの利用が考えられる。

`owl:Nothing` はインスタンスを持たないクラスであるが、`owl:Thing` がすべてのクラスのスーパークラスであるのに対して、`owl:Nothing` はすべてのクラスのサブクラスである。CLOS におけるクラス名はリスプの型と見なされるが、`owl:Thing` から `owl:Nothing` までのスーパー・サブクラス関係に加えて、クラス定義として OWL のクラスIDのみならず、インスタンス列挙、属性制約、クラス定義の共通集合(intersection, AND に相当)、和集合(union, OR に相当)、補集合(complement, NOT に相当)が許されており、将来は型理論の成果までセマンティックウェブ構築

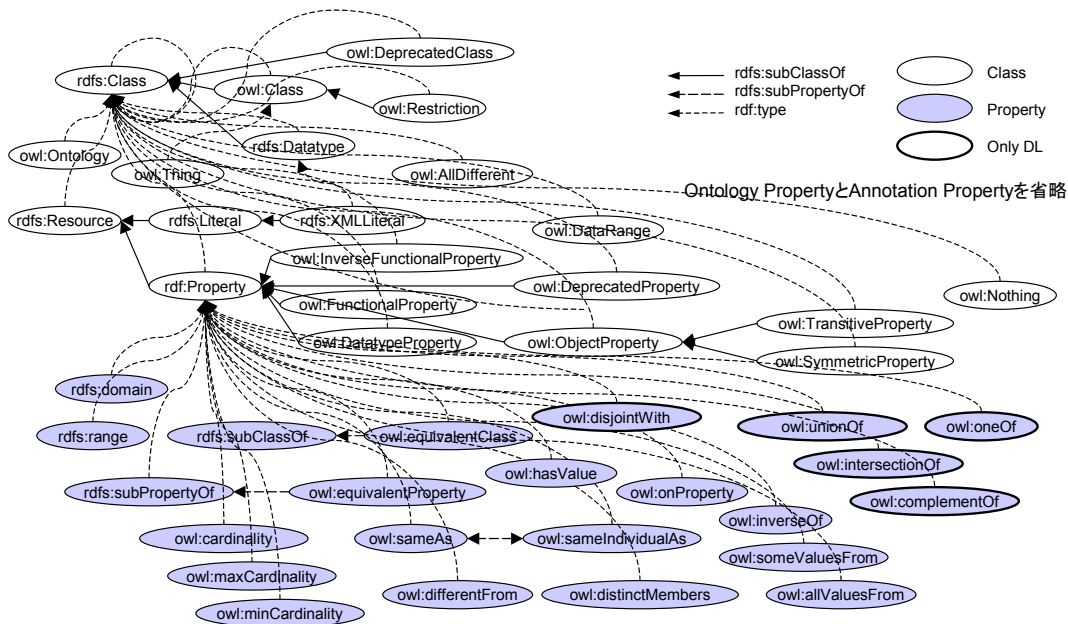


図3 OWL Lite/DLにおけるクラスとプロパティの階層構造

に取り入れることが可能となっている。

OWL Full に登場するエンティティは OWL DL と同じであるが、その定義は拡張されている。特に、`rdfs:Resource` は `owl:Thing` に、`rdfs:Class` は `owl:Class` に、`rdfs:Property` は `owl:ObjectProperty` に同一となり、このとき `owl:Class` のメタクラス定義を厳密に考えると、自分で自分を定義するというメタサーキュレーションを実現しなければならなくなる。

## 7. セマンティックウェブとブートストラッピング

よく知られたセマンティックウェブの論説では[TBL 2001], ルーシーのパーソナルエージェントが医者エージェントからお母さんの処方箋をもらい、ルーシーの自宅20マイル四方にあるお母さんの保険が利く病院を選び出し、兄弟であるピートとルーシーのエージェントは協力してスケジュールを調整し、病院の予約をとる。ここで各エージェントはウェブ上に記載されている用語を理解するために、ウェブ上のどこかにあるオントロジを参照することになっている。このエージェントはウェブを訪問しそのコンテンツをとってくるクローラのようなものと理解できるが、セマンティックウェブ技術は、現在人間向けのウェブコンテンツをこの論説のようにソフトウェアエージェントが理解できるようにするためのものであり、そのための基盤技術としてオントロジ技術に期待がよせられている。RDF, RDFS, OWL, DAML-Sなどはセマンティックウェブ用オントロジ記述のための知識表記法である。

セマンティックウェブは人工知能ではないとティム・バーナーズ＝リーは言うが、上記論説においてウェブコンテンツとオントロジを参照して主人の代理として行動するエージェントは、明らかに実現された AI である。エージェントが新しいオントロジを発見したとき、新しい推論能力を「ブートストラップ」することさえできると述べられているが、それこそ究極の AI と言ってもよいものであろう。しかし現在のセマンティックウェブ技術からどのようにそれが可能になるか、その道筋は明らかになっていない。乳幼児が生得的に持っていると言われる上位オントロジと言語獲得のためのバイアスから言語をブートストラップするように、RDF や RDFS で記述されたウェブコンテンツやオントロジからエージェントが能力をブートストラップする仕組みについての研究が重要である。

## 8. おわりに

今後2年間におけるセマンティックウェブ・プロセッサの開発計画について述べた。できれば、多くの方々の協力を得て開発スピードと品質を上げ、成果をオープンにして我が国のこの分野の発展に寄与したい。

## 9. 謝辞

本報告は、文科省ITプログラム「ITを活用した大規模システムの運用支援システムの構築」、の一部として実施されたものである。本プロジェクト実施では大須賀節雄東京大学名誉教授に技術評価委員長をお願いし、オントロジ構築に関して大阪大学溝口理一郎教授にご協力戴いている。記して感謝の意を表する。

## 参考文献

[Brickley 2002] Brickley, D., and R.V. Guha, “RDF Vocabulary Description Language 1.0: RDF Schema”, <http://www.w3.org/TR/2002/WD-rdf-schema-20021112/>, W3C, 2002.

- [Brickley 2003] Brickley, D., and R.V. Guha, “RDF Vocabulary Description Language 1.0: RDF Schema”, <http://www.w3.org/TR/rdf-schema/>, W3C, 2003.
- [FIPA XC00011B] FIPA RDF Content Language Specification, <http://www.fipa.org/>, 2001.
- [Grant 2003] Grant, J., and D. Beckett, “RDF Test Cases”, <http://www.w3.org/TR/rdf-testcases/>, W3C, 2003.
- [長谷川] 長谷川ほか, 「タブロー法とモデル生成型定理証明」, 人工知能学会誌, Vol.16, No.5, pp661-667, 2001.
- [Hayes 2001] Hayes, P., “RDF Model Theory”, <http://blogspace.com/rdf/modeltheory/>, W3C, 2001.
- [Hayes 2003] Hayes, P., “RDF Model Theory”, <http://www.w3.org/TR/rdf-mt/>, W3C, 2003.
- [Hjelm 2001] Hjelm, J., Creating the Semantic Web with RDF, WILEY, New York, 2001.
- [兼岩] 兼岩, 佐藤, 「DL: Description Logics」, 人工知能学会誌, Vol.18, No.1, pp.73-82. 2003.
- [神崎] 神崎, <http://www.kanzaki.com/docs/sw/rdf-xml.html>.
- [Koide 2000] Koide, S., and M. Kawamura, “An Implementation of Case-based Memory of an Interface Agent by Lisp, Java, and C++”, Proceedings of the Japan Lisp User Group, Franz, Chofu, 2000.
- [小出 2002] 小出ほか, 「ITを活用した大規模システムの運用支援システム」, SICE システムインテグレーション部門講演会, pp.399-400, 2002.
- [Koide 2003] Koide, S., et al., “Operation-Support System for Large-Scale System Using Information Technology”, Proceedings of 5th Int. Conf. Enterprise Information Systems (ICEIS2003), Vol.4, pp.430-437, ESEO, Anger, 2003.
- [小出 2003] 小出, 北村, 「MOP3: セマンティックウェブプロセッサ」, 第17回人工知能学会全国大会, 1D4-03, 2003.
- [Lassila 1999] Lassila, O., and R.R. Swick, “Resource Description Framework (RDF) Model and Syntax Specification”, <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>, W3C, 1999.
- [Lassila 2002] Lassila, O., “Taking the RDF Model Theory Out for a Spin”, First Int. Semantic Web Conf. (ISWC2002), pp.307-317, 2002.
- [Riesbeck 1989] Riesbeck, C.K., and R.C. Schank, Inside Case-Based Reasoning, LEA, Hillsdale, 1989.
- [TBL 1998] Berners-Lee, J., T., “Notation3: Ideas about Web Architecture – yet another notation”, <http://www.w3.org/DesignIssues/Notation3>, 1998.
- [TBL 2000] Berners-Lee, J., T., “Primer: Getting into RDF & Semantic Web using N3”, <http://www.w3.org/2000/10/swap/Primer.html>, 2000.
- [TBL 2001] Berners-Lee, J., T., Hendler, and O. Lassila, “The Semantic Web”, SCIENTIFIC AMERICAN, Scientific American, May 2001.
- [Wright 1983] Wright, J.M., and M.S. Fox, SRL/1.5 User Manual, Carnegie-Mellon Univ., 1983.
- [DAML-S] The DAML Services Coalition, “DAML-S: Semantic Markup for Web Services”, <http://www.daml.org/services/daml-s/0.7/>.

```

# Since QName syntax is not legal N-triples syntax, and in the interests of brevity and readability, we will use the convention
# whereby a QName is used without surrounding angle brackets to indicate the corresponding uriref enclosed in angle brackets, eg the
# triple
#
# <ex:a> rdf:type rdfs:Property .
#
# should be read as an abbreviation for the N-triples syntax
#
# <ex:a> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/2000/01/rdf-schema#Property> .
#
<http://somewhere-for-eg/eg#Work> <http://www.w3.org/2000/01/rdf-schema#subClassOf> <http://www.w3.org/2000/01/rdf-schema#Resource> .
<http://www.w3.org/1999/02/22-rdf-syntax-ns#Property> rdfs:subClassOf rdfs:Resource .
<http://somewhere-for-eg/eg#Agent>
  rdfs:subClassOf rdfs:Resource .
<http://somewhere-for-eg/eg#Person>
  rdfs:subClassOf <http://somewhere-for-eg/eg#Agent> .
<http://somewhere-for-eg/eg#Document>
  rdfs:subClassOf <http://somewhere-for-eg/eg#Work> .
<http://somewhere-for-eg/eg#author>
  rdf:type
  rdfs:domain
  rdfs:range
  <http://somewhere-for-eg/eg#Document> .
<http://.../Proposal/>
  rdf:type
  rdfs:title
  "Information Management: A Proposal" .
<http://.../Proposal/>
  rdfs:author
  <http://somewhere-for-eg/eg#author>
  :a .
<http://.../Proposal/>
  rdf:type
  rdfs:author
  <http://somewhere-for-eg/eg#name>
  :a
  <http://somewhere-for-eg/eg#name>
  "Tim Berners-Lee" .

```

### Extended N-triple Syntax

```

ntripleDoc ::= line*
line ::= ws* ( comment | triple )? eoln
comment ::= # ( character - ( cr | lf ) )*
triple ::= subject ws+ predicate ws+ object ws* ' ' ws*
subject ::= uriref | nodeID | QName
predicate ::= uriref | QName
object ::= uriref | nodeID | literal | QName
uriref ::= '<' absoluteURI '>'
nodeID ::= ' ' name
literal ::= langString | datatypeString
langString ::= lang string '@' language ?
datatypeString ::= langString '^^' uriref
language ::= [a-z0-9]+ ( '-' [a-z0-9]+ )?
ws ::= space | tab
eoln ::= cr | lf | cr lf
space ::= #x20
cr ::= #xD
lf ::= #xA
tab ::= #x9
string ::= character* with escapes
name ::= [A-Za-z][A-Za-z0-9]*
absoluteURI ::= character+ with escapes
character ::= [#x20-#x7E]

```