

SWAD–Europe Deliverable 5.1: Schema Technology Survey

Project name:

Semantic Web Advanced Development for Europe (SWAD-Europe)

Project Number:

IST-2001-34732

Workpackage name:

5. Integration with XML Technology

Workpackage description:

<http://www.w3.org/2001/sw/Europe/plan/workpackages/live/esw-wp-5.html>

Deliverable title:

SWAD-Europe: Schema Technology Survey

URI:

http://www.w3.org/2001/sw/Europe/reports/xml_schema_tools_techniques_report

Authors:

Stephen Buswell, Dan Brickley, Brian Matthews

Abstract:

This report surveys the state of schema annotation and mapping technology. It takes a practical approach by targeting the work to the needs of developers, providing background to support our attempts to answer frequently asked questions on this subject. The report first reviews previous work on 'bridging languages', giving an overview of the major approaches and uses that to motivate further technical work to progress the state of the art in this area.

Status:

Snapshot release for discussion and editorial work. Further revisions are planned during WP4.

Comments on this document are welcome and should be sent to the public-esw@w3.org list. An archive of this list is available at <http://lists.w3.org/Archives/Public/public-esw/>

This report is part of [SWAD-Europe](#) [Work package 5: Integration with XML Technology](#) and addresses the topic of Schema annotation, and the relationship(s) between RDF and XML technologies. The variety of so-called 'schema languages' for the Web has caused some confusion. This document attempts to place them in context, and explore the state of the art in tools for mapping data between the different approaches.

To do this, we need to draw on a variety of examples. The following diagram depicts a very simple RDF Schema, as well as some instance data that uses it. The example was originally created for the RDF Schema specification, and is used here as a basis for explaining the RDF 'world view', and contrasting that with the perspective implicit in XML, XML DTDs and XML-based schema languages.

RDF Schema example 1:

"Bridging Languages", An Introduction

XML encodings of information models fall largely into two groups:

- Syntactic definitions such as DTD, XDR, XSD which define XML structures and validation rules but which do not define the semantics of the relationships between the elements.
- Semantic definitions such as RDF, DAML+OIL, OWL which define the information from a 'knowledge representation' standpoint but which do not normally provide for validation or completeness checking.

This paper discusses ways in which some meaning can be attached to, or inferred from, XML structures. It then looks at two languages, Schema Adjunct Framework (SAF) and Meaning Definition Language (MDL) which build on this approach.

NB: Part of this analysis is based on ideas found Robert Worden's paper on MDL [\[MDL\]](#)

A Simple Model of Meanings

Broadly speaking, when we describe our universe of discourse, we want to make statements of three types:

- Existence/Type statements "an object X of type T exists"; "there is a winegrower"; "there are two wines"
- Attribute/Value statements "the colour of this wine is red"
- Relationship statements "this wine is produced by that winegrower"

These three concepts can be found in many modelling paradigms from UML to Entity-Relationship diagrams. In RDF Schema the single concept 'property' covers both 'Attribute/Value' and 'Relationship'.

The implied semantics of XML structures

'Structural' XML (DTD, XSD and the like) does not explicitly encode the information in the manner discussed above. However, inspecting XML instances, we can see some patterns.

Objects and Instances -

In general, objects are represented by XML elements:

```
<winegrower name="Chateau Verpriced" >

  </winegrower>
```

implies the existence of a winegrower object. This could also be represented as, for example:

```
<organisation orgtype = "winegrower" name="Chateau Verpriced" >

  </organisation >
```

where the orgtype attribute is used to select a subclass of a more generic superclass, or even:

```
<organisation name="Chateau Verpriced" >
  <orgtype >winegrower</orgtype >

  </organisation >
```

where the orgtype subelement is used to select a subclass of a more generic superclass. Note that not every XML element corresponds necessarily to an object. Here 'name' corresponds to an Attribute/Value.

```
<winegrower >
  <name>Chateau Verpriced</name>

  </winegrower>
```

Note that one cannot distinguish between this and the preceding example relying purely on syntax of the source document – some higher-level interpretation is always required. A further complication is that there is not necessarily a simple relation between elements and objects: this may be context-dependent. Here the type of object represented by 'organisation' is modified by the context.

```
<winegrowers>
  <organisation name="Chateau Verpriced" >
  <organisation name="Chateau Verdrawn" >
</winegrowers>

<winemerchants>
  <organisation name="Cheap+Cheerful" >
  <organisation name="Rough+Ready" >
</ winemerchants >
```

So in general we can say something like "An element with name E represents an object of type T", where this may be further qualified by

- Context - a particular XPath within the document
- Subselection - dependency of a particular attribute or subelement value.

Attribute Values - In the majority of cases, object attribute values are represented by the contents of XML attributes or subelements. The semantics of the following are indistinguishable:

```
<wine>
<name>Vielles Bottes</name >
</wine>

<wine name = "Vielles Bottes" >
</wine>
```

There may a level of conditionality, for example in the generic uncommitted-schema style

```
<wine>
  <wine-prop prop-name = "name" prop-value = "Vielles Bottes" / >
  <wine-prop prop-name = "colour" prop-value = "noir" / >
</wine>
```

Here the meaning of the 'prop-value' attribute depends on the contents of 'prop-name' attribute.

Relationships - Relationships can be represented in XML structures in various ways. The simplest of these is nesting:

```
<winemerchant name = "Bristol Bottlers" >
  <wine>
    <name>Vielles Bottes</name>
    <colour>black</colour>
  </wine>
  <wine>
    <name>Weston's Finest</name>
    <colour>red</colour>
```

```

    </wine>
</winemerchant>

```

Here the nesting establishes a relationship between the "Bristol Bottlers" winemerchant and the wines they sell. Note that there is no fixed semantics to the parent-child element relationship - we could as easily list the winemerchants by wine, as in:

```

<wine>
  <name>Vielles Bottes</name>
  <colour>black</colour>
<winemerchant name = "Bristol Bottlers" />
<winemerchant name = "Bath Brewers" />
</wine>

```

Section to write on relationships implied by shared values (strings, ID-IDREF, URI, ..)

The Schema Adjunct Framework

The Schema Adjunct Framework (SAF, [2]) tries to extend the structural model of a document given by the schema with additional information about the 'meaning' of pieces of the information within the instance. In the SAF, this meaning is specified by adding information about the processing which should be applied when a particular item is received. One could see this as definition of meaning in terms of the operational semantics of the target system.

Such information includes mappings to relational databases, indexing parameters for native XML databases, business rules for additional validation, internationalization and localization parameters, or parameters used for presentation and input forms. Some of this information is used for domain-specific validation, some to provide information for domain-specific processing.

Information items are selected by means of XPath expressions; the processing information is given by reference to an external schema which represents the processing functionality of the target system.

SAF Examples (from the SAF draft) - <schema-adjunct target="http://www.example.com/pat-admit.xsd" xmlns:sql="http://www.example.com/sql-map.xsd" ...>

```

<document>
  <sql:server>192.168.0.6</sql:server>
  <sql:database>PATIENT_RECORDS_DB</sql:database>
</document>

<element context='admission'>
  <sql:table>TBL_ADMISSIONS</sql:table>
</element>

<element context='age'>
  <sql:column>COL_AGE</sql:column>
</element>

<attribute context='admission/@id'>
  <sql:column>COL_ADMIT_NBR</sql:column>
</attribute>

</schema-adjunct>

```

Here we are specifying storage/retrieval rules for the information in a relational database. Note that we give rules at all levels: document, element, attribute. The 'context' attribute selects the instance data by means of an XPath expression.

Meaning Definition Language

Schema Languages such as XSD Schema and RELAXNG are concerned with the structure of XML documents. UML, DAML+OIL and RDF-Schema are concerned with meaning. Meaning Definition Language (MDL) is a bridge between structure and meaning. Using MDL, an XML language designer can express how the structure of an XML document conveys its meaning.

Meaning Definition Language (MDL, [1][3]) is a SAF implementation which extends the ideas behind SAF by specifying the processing rules for the schema instance information in terms of a formal representation of the knowledge by means of a UML class model or XSD Schema.

MDL provides:

- Meaning-Level Java API for XML
- Meaning-Level XML Query language
- Automated XML translation based on meaning

Literature Overview

This literature review is intended to provide a useful central starting point for locating resources on the Web relevant to the workpackage. It was developed by taking a base set of resources and following the links, verifying links lead to the correct resource and removing those links which lead to subject matter too far removed from the

subject of the workpackage (a necessarily subjective decision). The process is then repeated with the next set of links. This list makes no claim to be inclusive.

Resources have been classified under 6 top-level categories as below. Necessarily, many resources discuss issues related to more than one category; a more detailed analysis is given by the associated cross-reference table

Resource Cross Reference:

Subject Area	XML Core Technology	Schemas									Bridging				KN. Rep				Tools & Methods	
			DTD	XSD	XDR	SOX	Schema-tron	DSD	RELAX-NG		SAF	MDL	Interop		RDF	DAML+OIL	OWL	Topic Maps		
Resource																				
Extensible Markup Language (XML) 1.0	x																			
XML Schema Part 0: Primer		x		x																
XML Schema Part 1: Structures		x		x																
XML Schema Part 2: Datatypes		x		x																
XML-Namespaces Namespaces in XML	x																			
XML Information Set	x																			
XML Transformations (XSLT) Version 1.0	x																			
XML Path Language (XPath) Version 1.0	x																			
Document Definition Markup Language (DDML) Specification - Version 1.0		x																		
Document Content Description for XML		x																		
XML-Data		x																		
XML-Data Reduced (XDR)		x			x															
Docu- ment Struct- ure Descrip- tion 1.0		x																		
RELAX		x							x											
RELAX NG		x							x											
Sch- ema for Object-Oriented XML 2.0		x				x														
RDF Model and Syntax														x	x					
RDF Schemas														x	x					
An RDF Schema for the XML Inform- ation Set		x											x		x					
Schema Adjunct Frame- work										x	x		x							
Meaning Defini- tion Lang- uage										x	x		x							
Schem- atron		x					x													
ISO/IEC 13250: 2000 Topic Maps															x					x
XML Topic Maps (XTM) 1.0															x					x
Comp- arative Analysis of 6 XML schema lang- uages		x	x	x	x	x	x	x												
Descr- ibing Your Data: DTDs and XML Schemas																				

<http://www.brics.dk/DSD/>

See also "DSD: A Schema Language for XML", N. Klarlund, A. Moller, M. I. Schwatzbach, Proc. 3rd ACM Workshop on Formal Methods in Software Practice, 2000.

RELAX (REgular LAnguage description for XML) -

<http://www.xml.gr.jp/relax/>

RELAX NG -

<http://www.oasis-open.org/committees/relax-ng/>

RELAX NG, the next generation schema language for XML: clean, simple and powerful.

Schema for Object-Oriented XML 2.0 -

<http://www.w3.org/TR/NOTE-SOX>

RDFMS Resource Description Framework (RDF) Model and Syntax -

<http://www.w3.org/TR/REC-rdf-syntax>

RDF Schema Resource Description Framework (RDF) Schemas -

<http://www.w3.org/TR/rdf-schema/>

An RDF Schema for the XML Information Set -

<http://www.w3.org/TR/xml-infoset-rdfs.....>

This W3C Note defines an RDF schema for the XML Infoset.

Schema Adjunct Framework -

http://www.tibco.com/solutions/products/extensibility/resources/saf_dec2000.htm

Meaning Definition Language -

<http://www.charteris.com/mdl/>

Schematron -

<http://www.ascc.net/xml/resource/schematron/>

An XML Structure Validation Language using Patterns in Trees

ISO/IEC 13250:2000 Topic Maps -

<http://www.y12.doe.gov/sgml/sc34/document/0129.pdf>

XML Topic Maps (XTM) 1.0 -

<http://www.topicmaps.org/xtm/1.0/>

This specification provides a model and grammar for representing the structure of information resources used to define topics, and the associations (relationships) between topics. Names, resources, and relationships are said to be characteristics of abstract subjects, which are called topics. Topics have their characteristics within scopes: i.e. the limited contexts within which the names and resources are regarded as their name, resource, and relationship characteristics. One or more interrelated documents employing this grammar is called a "topic map."

Schema Language Reviews, Validation Issues

Comparative Analysis of 6 XML schema languages -

<http://www.cobase.cs.ucla.edu/tech-docs/dongwon/sigmod-record-00.html>

As XML is emerging as the data format of the internet era, there is an substantial increase of the amount of data in XML format. To better describe such XML data structures and constraints, several XML schema languages have been proposed. This paper presents a comparative analysis of six noteworthy XML schema languages: XML DTD; XML Schema; XDR; SOX ;Schematron; DSD.

Describing Your Data: DTDs and XML Schemas -

<http://www.xml.com/pub/1999/12/dtd/>

Document Type Definitions and XML Schemas both provide descriptions of document structures. The emphasis is on making those descriptions readable to automated processors such as parsers, editors, and other XML-based tools. They may also carry information for human consumption, describing what different elements should contain, how they should be used, and what interactions may take place between parts of a document

Schematron Tutorial -

<http://www.zvon.org/HTMLOnly/SchematronTutorial/General/contents.html>

<http://www.zvon.org/HTMLOnly/SchematronTutorial/General/contents.html>

The Schematron is a simple and powerful Structural Schema Language

Validating XML with schematron -

<http://www.xml.com/pub/a/2000/11/22/schematron.html>

Schematron is an XML schema language, and it can be used to validate XML. (Requires familiarity with XML 1.0, DTDs, XSLT, and Xpath).

Schemachine -

<http://www.topologi.com/public/Schemachine.pdf>

This note proposes a possible framework for supporting modular XML validation.

XML Pipeline Definition Language Version 1.0 -

<http://www.w3.org/TR/xml-pipeline>

This Note describes the features and syntax for XML Pipeline Definition Language. Pipeline is an XML vocabulary for describing the processing relationships between XML resources. A pipeline document specifies the inputs and outputs to XML processes and a pipeline controller uses this document to figure out the chain of processing that must be executed in order to get a particular result.

Semantic-Syntactic Relationship, Differences, Interoperability

The 'Cambridge Communiqué' -

<http://www.w3.org/TR/schema-arch>

This discusses the architectural relationship between the schema work being undertaken within XML and RDF activities.

Markup Languages: Comparison and Examples - XML, RDF, DAML -

<http://trellis.semanticweb.org/expect/web/semanticweb/comparison.html>

Contains a comparison table showing the tradeoffs and differences among these markup languages.

Dublin Core in RDF/XML -

<http://dublincore.org/documents/2001/09/20/dcmes-xml/>

The Dublin Core Metadata Element Set V1.1 (DCMES) can be represented in many syntax formats. This document explains how to encode the DCMES in RDF/XML, provides a DTD to validate the documents and describes a method to link them from web pages.

DTD's for the Dublin Core Element Set -

<http://dublincore.org/documents/2001/09/20/dcmes-xml/dcmes-xml-dtd.shtml>

Harvesting RDF from XLINKs -

<http://www.w3.org/TR/2000/NOTE-xlink2rdf-20000929/>

Both XLink and RDF provide a way of asserting relations between resources. RDF is primarily for describing resources and their relations, while XLink is primarily for specifying and traversing hyperlinks. However, the overlap between the two is sufficient that a mapping from XLink links to statements in an RDF model can be defined. Such a mapping allows XLink elements to be harvested as a source of RDF statements. XLink links (hereafter, "links") thus provide an alternate syntax for RDF information that may be useful in some situations.

This Note specifies such a mapping, so that links can be harvested and RDF statements generated. The purpose of this harvesting is to create RDF models that, in some sense, represent the intent of the XML document.

Language Comparisons - XML, RDF, DAML+OIL, OWL -

<http://www.daml.org/language/features>

Contains a table summarizes differentiating language features available in XML, RDF, DAML+OIL, and OWL.

XML Processing: position paper -

<http://www.w3.org/2001/06/rdf-xproc/1>

This paper outlines how RDF fits into the XML family of specifications; how RDF software components and vocabularies might relate to the XML processing environment. XML Documents represent the XML Infoset; RDF graphs represent what those Infosets are trying to tell us about objects, their inter-relationships and properties.

Bridging the Gap between RDF and XML -

<http://www-db.stanford.edu/~melnik/rdf/fusion.html>

The convoluted syntax of the RDF 1.0 specification is a major obstacle for the broad acceptance of RDF. The goal of this proposal is to allow every "legacy" XML document to have an RDF model. The advantages of this approach include:

1. The semantics of XML documents can be made explicit. Both structural and semantic markup can coexist in the same document.
2. RDF can be used to annotate existing XML documents.
3. "RDF-enabled" XML can still be rendered and transformed using XSLT.
4. Using small changes in XML DTDs, meaningful RDF documents can be produced from original XML documents. But every XML document (even those without DTDs) has a default RDF interpretation.

Connecting XML, RDF and Web Technologies for Representing Knowledge on the Semantic Web -

<http://ilrt.org/people/cmdjib/talks/xmleurope2002/print.html>

In order to represent knowledge for it to be usable web-wide and in interoperable ways, it should be done using

well-known and appropriate web technologies. These include XML and RDF. RDF can be used with many XML technologies such as XML Namespaces, XML Schema, RELAX NG, XSLT and is related to many more.

This paper describes how these technologies are best used together, their relationships and where each of them can be appropriately applied.

Why RDF model is different from the XML model -

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

This note is an attempt to answer the question, "Why should I use RDF - why not just XML?". This note assumes that the XML data model in all its complexity, and the RDF syntax as in RDF Model and Syntax, in all its complexity. It doesn't try to map one directly onto the other -- it expresses the RDF model using XML.

The Yin Yang Web: XML Syntax and RDF Semantics -

<http://www2002.org/CDROM/refereed/231/>

XML is the W3C standard document format for writing and exchanging information on the Web. RDF is the W3C standard model for describing the semantics and reasoning about information on the Web. RDF and XML are based on two different paradigms.

This paper develops a model-theoretic semantics for the XML XQuery 1.0 and XPath 2.0 Data Model, which provides a unified model for both XML and RDF. This unified model can serve as the basis for Web applications that deal with both data and semantics. The paper shows how the RDF world can take advantage of XML query languages, and how the XML world can take advantage of the reasoning capabilities available for RDF.

Knowledge Representation Language Reviews

Comparison of ontology languages -

<http://www.fipa.org/docs/input/f-in-00045/f-in-00045.pdf>

Ontology Overview from Motorola Labs with a comparison of ontology languages

Topic maps, RDF, DAML, OIL -

<http://www.ontopia.net/topicmaps/materials/tmrdfoidaml.html>

This paper provides quick introductions to each of the technologies, highlighting the similarities and differences between them. These technologies all come from very different backgrounds, and tend to be presented in very different ways, and yet on closer examination their anatomies turn out to be surprisingly similar.

The starting point of the paper is the observation that all these technologies provide a restricted set of mechanisms for making statements about the Universe of Discourse. The introductions to the technologies are done by describing how concepts such as 'thing', 'relationships between things', 'properties of things', 'types of things', 'names of things', 'kinds of things' and so on are represented within each technology.

After comparing the technologies it is shown how each of these technologies relate to one another and to what extent they build on each other or compete with each other. More importantly, the paper also shows how data can be moved from one representation to another, and how tools implementing the various technologies can be made to work together.

A Topic Map Data Model: An Infoset-based Proposal -

<http://www.ontopia.net/topicmaps/materials/proc-model.html>

This document defines an abstract model for topic maps which makes explicit the implicit data models of ISO 13250 and XTM 1.0. It also defines a processing model for XTM 1.0 based on the data model.

tolog: A topic map query language -

<http://www.ontopia.net/topicmaps/materials/tolog.html>

This paper describes a query language for topic maps.

On the integration of Topic Map data and RDF data -

<http://www.semanticweb.org/SWWS/program/full/paper53.pdf>

RDF and TopicMaps: An Exercise in Convergence -

<http://www.topicmaps.com/topicmapsrdf.pdf>

XML Topic Maps through RDF glasses -

<http://www.cogx.com/rdfglasses.html>

The information represented in a topic map, expressed in one of the XML interchange syntaxes for topic maps, can, at some level of detail, be translated into information that is expressed in one of the XML interchange syntaxes for RDF information. The translated information can then be used in the context of RDF applications that would otherwise not be able to use it.

Bridging Language Reviews

Meaning Definition Language -

<http://www.charteris.com/mdl/>

Schema Languages such as XSD Schema and RELAXNG are concerned with the structure of XML documents. UML, DAML+OIL and RDF-Schema are concerned with meaning. Meaning Definition Language (MDL) is a bridge between structure and meaning. Using MDL, an XML language designer can express how the structure of an

XML document conveys its meaning.

Schema Adjunct Framework -

http://www.tibco.com/solutions/products/extensibility/resources/saf_dec2000.htm

Schema adjuncts are a mechanism for extending XML schema languages, and for providing information from such extensions to programs that process XML instances. To process XML instances for a given schema, many environments need additional information which is typically not available in the schema itself. Such information includes mappings to relational databases, indexing parameters for native XML databases, business rules for additional validation, internationalization and localization parameters, or parameters used for presentation and input forms.

The Schema Adjunct Framework is an XML-based language used to associate domain-specific data with schemas and their instances, effectively extending the power of existing XML schema languages such as DTDs or XML Schema.

Ontologies, Methodologies, Tools, Architecture

Ontology Development 101: A Guide to Creating Your First Ontology -



<http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-nov-mcguinness.html>

Ontology Editing Tools -

<http://www.xml.com/pub/a/2002/11/06/ontologies.html>

http://www.xml.com/2002/11/06/Ontology_Editor_Survey.html

This survey covers software tools that have ontology editing capabilities and are in use today. The tools may be useful for building ontology schemas (terminological component) alone or together with instance data. Ontology browsers without an editing focus and other types of ontology building tools are not included. Otherwise, the objective was to identify as broad a cross-section of editing software as possible. The editing tools are not necessarily production level development tools, and some may offer only limited functionality and user support.

Report on ontology tools -

http://ontoweb.aifb.uni-karlsruhe.de/About/Deliverables/D13_v1-0.zip

OntoWeb report on a comparative study of 11 ontology editors plus several other ontology tools

Evaluation of Ontology-based Tools -

<http://km.aifb.uni-karlsruhe.de/eon2002/>

Common European Research Information Format (CERIF) -

<http://www.ub.uib.no/avdeling/fdok/cris/UKmeeting/The%20CERIF.htm>

The CERIF- SW project brings together expertise of research institution in developing distributed information system of heterogeneous data sources, database harmonization, providing data access to research data, developing Knowledge Management and Semantic Web solutions, metadata creation and use to develop a set of tools for building next generation of research information systems, and to integrate using new technologies a set of European Research Information Systems for advanced research information retrieval capabilities.

Architectural Principles of the World Wide Web -

<http://www.w3.org/TR/2002/WD-webarch-20020830/>

The World Wide Web is a networked information system. Web Architecture is the set of principles that all agents in the system follow to create the large-scale effect of a shared information space. Identification, data formats, and protocols are the main technical components of Web Architecture, but the large-scale effect depends on social behavior as well.

This document strives to establish a reference set of principles for Web architecture

RSS 1.0+ Sample Scenario (draft)

Some in the RSS community would prefer to use a non-RDF serialization for RSS feeds, to avoid the perceived syntactic burden of using RDF (eg. serialization rules). Others have been creating extension vocabularies for use in RSS feeds, to augment the basic structure of a feed with additional information, about the channel, the documents mentioned in the channel, or the things described by those examples. The goal here will be to explore the applicability of schema annotation to this problem: is it possible to deploy mixed-namespace RSS using annotated XML schemas, instead of RDF's XML syntax. If so, what does this mean for practicalities of defining extension vocabularies for use in RSS? eg. are those vocabularies also re-usable in non-RSS RDF documents (eg. Dublin Core, Creative Commons, etc.).

reagleMIT thinks a response to DanC is scaling the SW, most of the RSS people won't care and won't do the work, but the SW folks need to spread some pixie-dust and suck in/import whole realms of XML apps into the SW for them (and part of this importation is probably this chunk is ambiguous).

WSDL Sample Scenario

- ws desc reqs: <http://www.w3.org/TR/2002/WD-ws-desc-reqs-20021028/> desc wg: <http://www.w3.org/2002/ws/desc/>

See EricP's work on this, and other...

P3P: Brian's mapping

Apple plist files

[PropertyList-1.0.dtd](#) , and XSLT to convert it to RDF. Also [sample data](#).

RDF calendar tests

See [RDF Calendar workspace](#), esp [test file collection](#): xcal to rdf scenario.

References

- [MDL] A Meaning Definition Language, R. Worden, Charteris 2001 <http://www.charteris.com/mdl/>
 [SAF] Schema Adjunct Framework
http://www.tibco.com/solutions/products/extensibility/resources/saf_dec2000.htm
 [XMD] Professional XML Meta Data, Ahmed et al, Wrox. ISBN 1-861004-51-6 (MDL - ch 8)
- [RDFMS]
[Resource Description Framework \(RDF\) Model and Syntax Specification](#), O. Lassies and R. Swick, Editors. World Wide Web Consortium. 22 February 1999. This version is <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>. The [latest version of RDF M&S](#) is available at <http://www.w3.org/TR/REC-rdf-syntax>.
- [RDFSCHEMA]
[RDF Vocabulary Description Language 1.0: RDF Schema](#), D. Brickley, E.V. Guha, Editors, World Wide Web Consortium W3C Working Draft, work in progress, 19 March 2002. This version of the RDF Primer is <http://www.w3.org/TR/2002/WD-rdf-schema-20020430/>. The [latest version of the RDF Primer](#) is at <http://www.w3.org/TR/rdf-schema/>.
- [RDFRELATIONAL1]
[Storing RDF in a relational database](#), Sergey Melnik, Stanford University, 2000-2001
- [SYNTACTICWEB]
[The Syntactic Web](#) - Syntax and Semantics on the Web, Jonathan Robie, Software AG, USA in proceedings XML Conference, December 9-14 2001, Orlando Florida, USA.
- [RSS10]
[RDF Site Summary \(RSS\) 1.0](#), RSS-DEV Working Group, 2001-05-30

Further References

RAW NOTES: - This section tracks links, content and ideas that should be integrated into the document. For now I'm just hoarding links I don't want to miss in the final report. (mail [me](#) if I've missed anything obvious from this survey.

- content from [workpackage description](#)
- [The Cambridge Communiqué](#), W3C NOTE 7 October 1999
- [Web Architecture: Describing and Exchanging Data](#), W3C Note 7 June 1999
- [XML Information Set](#), W3C Recommendation 24 October 2001
- [An RDF Schema for the XML Information Set](#), W3C Note 6 April 2001
- [Research notebook: On edge-labelled graphs in XML](#)
- [XML Syntax Recommendation for Serializing Graphs of Data](#), submission to QL'98 workshop by Andrew Layman (andrewl@microsoft.com), December 2, 1998.
- [XML and RDF Schema](#), notes by Dan Brickley
- [Xemantics: Adding object Identity and Explicit Relations to XML](#), R.V.Guha
- [XML with Relational Semantics](#), Sandro Hawke, 05 2001
- [RSS](#) use case -- extensions etc.
- [Using Grammar Annotation to Bring Peace and Understanding to the Semantic Web](#), Sandro Hawke
- [XML Attribute Value Namespace Expansion](#), Eric Prud'hommeaux
- [Schema Adjunct Framework](#), Scott Vorthmann, Lee Buck (Extensibility, Inc.)
- [TIBCO page on Schema Adjunct Framework](#)
- [Meaning Definition Language](#), Charteris. Problem statement: *The problem is XML interoperability - the proliferation of XML languages with overlapping meanings but different structures, and the need to inter-operate between them. If you develop applications for several XML languages, you need to know their structures - DTDs or schemas - and how those structures convey meaning. With even a few different languages, this makes XML applications expensive to develop and maintain.*
- [MDL whitepaper \(pdf\)](#), Robert Worden (May 2001)

B References – Tools and Projects