

Solving Problems with the Resource Description Framework

Practical

RDF



O'REILLY®

Shelley Powers

TWITCH-1048
IPR2025-00308

Practical RDF

Shelley Powers

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

Practical RDF

by Shelley Powers

Copyright © 2003 O'Reilly & Associates, Inc. All rights reserved.
Printed in the United States of America.

Published by O'Reilly & Associates, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly & Associates books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (safari.oreilly.com). For more information, contact our corporate/institutional sales department: (800) 998-9938 or corporate@oreilly.com.

Editor: Simon St.Laurent

Production Editor: Mary Brady

Cover Designer: Ellie Volckhausen

Interior Designer: David Futato

Printing History:

July 2003: First Edition.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly & Associates, Inc. Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Java™, all Java-based trademarks and logos, and JavaScript™ are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries. O'Reilly & Associates, Inc. is independent of Sun Microsystems. Where those designations appear in this book, and O'Reilly & Associates, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps. The association between the image of a secretary bird and the topic of RDF is a trademark of O'Reilly & Associates, Inc.

While every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

ISBN: 978-0-596-00263-3

[LSI]

[2011-04-01]

databases, and OWL would be comparable to the business domain applications such as PeopleSoft and SAP. Both PeopleSoft and SAP make use of existing data storage mechanisms to store the data and the relational data model to ensure that the data is stored and managed consistently and validly; the products then add an extra level of business logic based on patterns that occur and reoccur within traditional business processes. This added business logic could be plugged into a company's existing infrastructure without the company having to build its own functionality to implement the logic directly.

OWL does something similar except that it builds in the ability to define commonly reoccurring inferential rules that facilitate how data is queried within an RDF/XML document or store. Based on this added capability, and returning to the RDF/XML example in the last section, instead of being limited to queries about a specific movement based on a specific resource, we could query on movements that occurred because the document was moved to a new domain, rather than because the document was just moved about within a specific domain. Additional information can then allow us to determine that the document was moved because it was transferred to a different owner, allowing us to infer information about a transaction between two organizations even if this "transactional" information isn't stored directly within elements.

In other words, the rules help us discover new information that isn't necessarily stored directly within the RDF/XML.



Chapter 12 covers ontologies, OWL, and its association with RDF/XML. Read more about the W3C's ontology efforts at <http://www.w3.org/2001/sw/WebOnt/>. The Use Cases and Requirements document can be found at <http://www.w3.org/TR/webont-req/>.

Another very common use of RDF/XML is in a version of RSS called RSS 1.0 or RDF/RSS. The meaning of the RSS abbreviation has changed over the years, but the basic premise behind it is to provide an XML-formatted feed consisting of an abstract of content and a link to a document containing the full content. When Netscape originally created the first implementation of an RSS specification, RSS stood for RDF Site Summary, and the plan was to use RDF/XML. When the company released, instead, a non-RDF XML version of the specification, RSS stood for Rich Site Summary. Recently, there has been increased activity with RSS, and two paths are emerging: one considers RSS to stand for Really Simple Syndication, a simple XML solution (promoted as RSS 2.0 by Dave Winer at Userland), and one returns RSS to its original roots of RDF Site Summary (RSS 1.0 by the RSS 1.0 Development group).

RSS *feeds*, as they are called, are small, brief introductions to recently released news articles or weblog postings (weblogs are frequently updated journals that may include links to other stories, comments, and so on). These feeds are picked up by

aggregators, which format the feeds into human consumable forms (e.g., as web pages or audio notices). RSS files normally contain only the most recent feeds, newer items replacing older ones.

Given the transitory nature of RSS feeds as I just described them, it is difficult to justify the use of RDF for RSS. If RDF's purpose is to record assertions about resources that can be discovered and possibly merged with other assertions to form a more complete picture of the resource, then that implies some form of permanence to this data, that the data hangs around long enough to be discovered. If the data has a life span of only a minute, hour, or day, its use within a larger overall "semantic web" tends to be dubious, at best.

However, the data contained in the RSS feeds—article title, author, date, subject, excerpt, and so on—is a very rich source of information about the resource, be it article or weblog posting, information that isn't easily scraped from the web page or pulled in from the HTML meta tags. Additionally, though the purpose of the RSS feed is transitory in nature, there's no reason tools can't access this data and store it in a more permanent form for mergence with other data. For instance, I've long been amazed that search tools don't use RSS feeds rather than the HTML pages themselves for discovering information.

Based on these latter views of RSS, there is, indeed, a strong justification for building RSS within an RDF framework—to enhance the discovery of the assertions contained within the XML. The original purpose of RSS might be transitory, but there's nothing to stop others from pulling the data into more permanent storage if they so choose or to use the data for other purposes.

I'll cover the issue of RSS in more detail in Chapter 13, but for now the point to focus on is that when to use RDF isn't always obvious. The key to knowing when to make extra effort necessary to overlay an RDF model on the data isn't necessarily based on the original purpose for the data or even the transitory nature of the data—but on the data itself. If the data is of interest, descriptive, and not easily discovered by any other means, little RDF alarms should be ringing in our minds.

As stated earlier, if RDF isn't a replacement for some technologies, it is an opportunity for new ones. In particular, Mozilla, my favorite open source browser, uses RDF extensively within its architecture, for such things as managing table of contents structures. RDF's natural ability to organize XML data into easily accessible data statements made it a natural choice for the Mozilla architects. Chapter 14 explores how RDF/XML is used within the Mozilla architecture, in addition to its use in other open source and noncommercial applications such as MIT's DSpace, a tool and technology to track intellectual property, and FOAF, a toolkit for describing the connections between people.

Chapter 15 follows with a closer look at the commercial use of RDF, taking a look at OSA's Chandler, Plugged In Software's Tucana Knowledge Store, Siderean Software's Seamark, the Intellidimension RDF Gateway, and how Adobe is incorporating RDF data into its products.