Telecommunications Electronic Reviews (TER) Volume 10, Number 2, June 2003

ter
Technology Electronic Reviews
Formerly *Telecommunications Electronic Reviews*

Volume 10, Issue 2, June 1, 2003

Technology Electronic Reviews (TER) is a publication of the Library and Information Technology Association.

REVIEW OF: Kurt A. Gabrick and David B. Weiss (2002). Java 2EE (J2EE) and XML Development. Greenwich, CT: Manning.

by Michael B. Spring

Building distributed application frameworks, enterprise applications, and information marketplaces are difficult tasks. The initial learning curve can be high, even for experienced developers. Gabrick and Weiss have put together a very readable and comprehensive treatment of building enterprise applications using J2EE (Java 2 Enterprise Edition) and XML (eXtensible Markup Language). They have a clear bias, which this reviewer shares, for open standards-based approaches, but also maintain a healthy perspective toward proprietary solutions, suggesting that there are places and situations in which proprietary systems may be required or necessary. One of the most difficult tasks in writing a book at this level is finding a way to stay at the right level, neither omitting information that really is needed nor including information that really should be the responsibility of an appropriately prepared reader. The authors do a magnificent job of staying on task and level. The choices for Web references and preparatory reading are well taken. The inclusion and length of the appendices are also well reasoned.

The book focuses on the use of XML tools in Java applications - primarily Web-based applications. They cover the issues in six chapters, with the first and the last providing a synthesis. The substantive chapters include XML technology (chapter 2), XML as data (chapter 3), application integration using XML (chapter 4), and XML for client control (chapter 5). They include three appendices, the first of which is on design patterns. Throughout the book, the authors seek to raise the reader's view of the process to a high architectural level and they succeed. Appendix A with its six design patterns is worth the price of admission. Indeed, if there is something that might be provided in a second edition, it would be an extended appendix saying more about these design patterns. What is included is elegant, to the point, and clear. At the same time, one is left admiring the brevity and clarity of the authors' insights, at least this reader would have enjoyed them saying more.

Chapter 1 is devoted to providing an overview and the scope is well set out. In a mere 20 pages, they cover most of the basic concepts of distributed systems. Where they set themselves apart from others is how much detail they bring to bear. That error handling is an important challenge in distributed systems is easy to say. To go on in an overview at this level to then classify error types as process failures, omission failures, and arbitrary failures, with clear and concise definitions is unusual. They also cover both fault tolerance and fault masking in a brief one-page description. The format of the book is also well conceived. They highlight important definitions throughout, and this is valuable in all the chapters for those who are unfamiliar with one or another of the topics being discussed as most every reader who picks up this book will be. Chapter 1 goes on to talk about the J2EE development process. With the same simplicity and clarity of the first part of the chapter, they cover topics from analysis and design processes to problem tracking. The wealth of their experience begins to become clear as well, with lines like "many development projects use a bug tracking database, usually written by a college intern with limited skills." While the tools they discuss are in rapid flux given the evolution of Java and the Java tools sets, this chapter provides a valuable starting point for thinking about what enterprise applications will require, both conceptually and pragmatically.

In the introduction to XML, the authors endear themselves to this reviewer early on. They clearly situate the "markup languages" that tend to confuse users. Unlike many who cloud the water, they correctly class WML, ebXML, and XHTML as schema developed in accord with the XML standard. As a bonus, they go on to provide a unique explanation for the frequent confusion, suggesting that a schema defines legitimate markup and therefore constitutes a kind of markup language while XML itself is less a markup language and more a meta language. This is the kind of clarity of thinking and explanation that typifies the work. They quickly move beyond XML and schema to the parsing technologies available for XML. Notable in the treatment is the emphasis on the evolving state of the technologies and the search for efficient and scalable solutions. The section on alternative XML storage technologies, flat files, relational databases, parsed binary files, and XML databases is well introduced and later expanded in chapter 3. If there is a minor nit to pick with the authors, it is the process of using XML when there are collections of documents to work with. While one can anticipate this from what they say about both relational and XML databases, it would have been nice to see one of the examples address the process of manipulations of collections in XML documents in databases such as xindice or Tamino.

The current state of the Java APIs for XML processing is hard to stay on top of. The various factories, parsers, and filters get confusing for all but the best prepared object-oriented designers. The authors are very clear about the structure of these APIs, their relationship one to the other, and the reasons for the architecture. As they do throughout the book, they introduce at every appropriate point the pros and cons of one approach over another. As they so clearly state at the beginning of chapter 3, "this book is about using XML in your J2EE applications with discretion." The first part of chapter 3 is devoted to XML for component interfaces, and while of interest, particularly related to cross-platform applications, it is an area in which much has been written by others already. The second part of the chapter is devoted to XML as persistent data. The authors are clear about the fact that reliable, scalable, and functional XML databases are still in the future, and will play an important role as they mature. Their summary on XML repository solutions, the issues to consider when deciding whether or not to use XML for data storage, and their advice about wrappers and mediators are well worth reading. Again, the one minor weakness is some examples of XML database queries and the implications for managing collections. Everything that they do say about XPath and XQuery is accurate and clean. At the same time, seeing the results of a query on a native XML database provides one with significant insights, and would add to the many clear images and insights already provided by the authors.

Chapter 4 is devoted to integrating J2EE applications. The real value of the chapter comes in two parts. First, they posit integration as possible at four levels: data, message, procedure, and object, with each of the levels being more complex and more sophisticated. By "complex," they mean to infer that there are more pieces that need to be put together. By "sophistication," they mean to suggest a measure of functionality provided by the underlying framework. As a simple example, object-level integration most often includes a level of indirection for finding objects of interest and some form of distributed registry that allows appropriate objects to be found. This freedom from having to know the location of a given service requires the additional complexity of having registries, knowing how to register services, or objects, and knowing how to look them up. They discuss the advantages and disadvantages of integration at each of these levels and the analysis is clear, concise, and to the point. They also introduce constraint-based modeling as the last part of this first section. As they lay it out, it is simple and clear, but it leaves this reader feeling that the wealth of experience they bring to bear in constraint-based modeling is far more important than the simple concepts they lay out. The remainder of the chapter is devoted to an example of a Web services-based application using SOAP (Simple Object Access Protocol) and UDDI (Universal Discovery, Description and Integration). It does a nice job of showing how the upper three levels of integration build on each other.

Chapter 5 will be of great interest to some readers and of little interest to others. It addresses the issues of various levels of encapsulation and indirection in the use of XML to meet the demands of differentiated output to Web browsers or thin clients. Most notable in the treatment, which like the other sections of the book moves from simple brute force approaches to more sophisticated framework approaches, is the inclusion of sections on formatting objects and publishing frameworks. XSLT Formatting Objects (XSLTFO) provides a device-independent presentation markup that is currently only implemented in a prototype PDF maker. Ultimately, with the exception of bandwidth requirements, XSLTFO might be used to move XML documents to a variety of different output devices. As the authors see it, its primary functionality is to prepare binary formats for presentation. In this reviewer's opinion, again ignoring bandwidth requirements, XSLTFO may have wider application. The authors choose Web publishing frameworks, such as Cocoon, as the ultimate level of sophistication in preparing presentations.

Finally, Chapter 6 ties everything together with an extended example that shows all the stages in the development of an application, from analysis and design through testing. The authors do an excellent job of making the complex clear and the book, while written in a period of rapid development of new applications and tools, will likely have a longer shelf life than most books in this category, given the ability of the authors to abstract and clarify critically important principles and patterns that will survive the particular tools.

Michael B. Spring is an Associate Professor of Information Science and Telecommunications at the University of Pittsburgh.

REVIEW OF: Stuart D. Lee (2002). Electronic Collection Development: A Practical Guide. Neal-Schuman.

by Cindy Schofield-Bodt

For all the libraries that have begun to collect electronic resources outside of the guidelines of their collection development policies, this book is just in time.

Stuart D. Lee writes with authority about electronic collection development from his position as the Head of the Learning Technologies Group at Oxford University Computing Services. His involvement in digitization projects, work groups on digital images and resources, and previous writing projects make him an expert worth hearing from.

Electronic Collection Development: a Practical Guide is a short introduction to the basic issues facing librarians as the world of electronic publishing expands. The book consists of five chapters, a glossary, and a bibliography that includes helpful Internet sites. A comprehensive index provides instant access to the work when inquiring about specific concepts. The reader does have to keep in mind that Lee is writing from his office in the UK and some of the examples he uses are more familiar to overseas colleagues. None of the concepts or suggested guidelines are limited by continent -- the subject matter, after all, is about resources that are available via the truly international Internet.

Chapters 1-3 provide the reader with an introduction to the various electronic resources that are available. The author is convinced that the beginner in the field must have a basic understanding of the products and the issues that emerge as digital material is acquired for a library collection. Lee argues throughout the book that digital resources and traditional library materials need not be considered on separate planes. One collection development policy with a focus on the scope of the collection, the needs of the users, and the mission of the library can be developed to address various information formats that meet the outlined needs. "The single most important message of this book is that electronic resources should be considered alongside printed resources (as indeed in some cases, such as e-journals, they must be) and that libraries should formulate an overall 'coherent' collection development policy covering all materials" (p.7). Lee goes on to say that the policy should consist of an assessment portion and be available to everyone who is involved in purchasing decisions.

While the author advocates the use of one collection development policy to guide acquisitions across formats, he recognizes that one policy increasingly means one pot of money from which purchases can be made. This can be a problem since the one major area where print and electronic resources diverge is in the predictability of cost. It is therefore very important that it is clear who in an organization is responsible for budgeting decisions surrounding electronic resources. There must be an awareness of the generally higher prices and unpredictable price fluctuations of comparable publications in e-format.

Additional introductory concepts presented include the need to evaluate publications, prioritize requests, establish delivery protocols, and monitor usage. These combined form the introductory analysis necessary to consider electronic resource acquisition.

In his attempt to describe the electronic resources landscape, Lee addresses the major issues one needs to understand to build an electronic collection. He sees the clearest division between print and electronic as the separation between the data itself and the delivery mechanism. The fact that the same data (be it textual, numerical, pictorial, etc.) can be repackaged and offered by various suppliers, means that it is possible to buy the same information from multiple sources. A collection developer must decide what information is needed and what delivery platform is desired. Those decisions lead to the need to understand such concepts as remote and local access, authentication issues, bundled deals, push/pull technology, and linkage services. An explanation of each of these concepts is followed by relevant examples.

The overview of the electronic landscape includes a discussion of archiving and long-term access. In his discussion, Lee points out the reality that one payout for print materials that remain intact for decades or even centuries is not at this time duplicated in the electronic landscape. Payment for access to current issues can mean that if a budget cut requires cancellation of an electronic subscription, access to the issues already paid for is also denied. Preservation of electronic material and ongoing access to purchased information is a thorny issue. Lee briefly explores the world of electronic archiving and describes a number of scenarios that may become the future reality.

Chapter 4 is developed as a "hands on" guide to the details of getting involved with on-line resources. In reference to the collection development policy, Lee again advises that one policy address all formats of information. He believes that the policy needs to be made "as public as possible." It needs to address clearly "where the institution currently is in terms of its holdings, and where it wishes to be" (p.64). He argues that only after the current state of the collection has been established can librarians consider more strategic-level decisions. A library should be asking itself: "...how far down the road of digital collection development do you want to travel? This is perhaps the fundamental problem, and there are many questions and issues which arise from this one simple query." If there is no clear picture of where the digital resource collection is heading and how it will interface with more traditional collections, experience has shown that resources will start to appear in an idiosyncratic fashion, without any cohesive policy in terms of targeting areas or matching priorities (p.65).

The discussion continues first around budget issues and then awareness of the dataset. Institutions must have clear guidelines regarding how funding is allocated and Lee offers advice on establishing a budget with electronic resources in mind. Determining what to buy through librarian research and user request is also a complicated process, but one that is enhanced through established assessment, trial, and evaluation procedures. A valuable evaluation checklist is provided in chapter 4 to be consulted as each new electronic product is experienced on trial. The suggestion is made that a core evaluation group be formed with certain members of the group responsible for certain sections of the checklist. Also in this chapter, elements of a product license are reviewed in detail with lists of advantages and disadvantages of various aspects of licensing and license-use issues. Finally, Lee walks the reader through the actual purchase of an electronic resource. If a librarian could read only one chapter of this book, chapter 4 is the one to choose.

Lee's final chapter deals with all that must happen behind the scenes after the dataset is purchased. He writes about cataloging and archiving the data, advertising, maintaining usage statistics, renewing subscriptions, and general troubleshooting. A model step-by-step approach to stepping into the world of electronic publishing is extremely helpful -- both comprehensive and concise. *Electronic Collection Development: A Practical Guide* is an excellent tool for getting on board with electronic collection development. Stuart Lee presents a process along with his philosophy that is so straight forward that one can only assume that as more copies of this book are read and the processes presented adopted, the daunting issues facing librarians hesitating at the door of the 21st century will become routine procedures.

Cindy Schofield-Bodt (Schofieldbc1@southernct.edu (mailto:Schofieldbc1@southernct.edu)), Librarian, Southern Connecticut State University, is currently the Buley Library Technical Services Division Head. She has taught cataloging and acquisitions courses as an adjunct at the SCSU Graduate School of Library And Information Science.

REVIEW OF: Tony Mancill (2002). Linux Routers: A Primer for Network Administrators. Upper Saddle River, NJ: Prentice-Hall.

by Ray Olszewski

Most readers of this review are likely to have at least a casual familiarity with Linux. Briefly, Linux is an operating system for computers. It has the unusual characteristic that all the source code for its components is freely available to anyone to review, modify, and improve, subject only to the requirement that the changes be made available on the same terms. Typically, Linux itself -- a "kernel" plus a small number of closely-related programs -- is combined with a larger suite of applications into a "distribution", which is itself either sold or given away (or both). The best known distributions are Red Hat, Mandrake, and Debian, and there are many others.

In this form, as what Linux aficionados call a "full-strength" distribution, Linux is in widespread use as an operating system for servers, and it plays a small role as an operating system for desktop and laptop workstations. But there is another way to use Linux: as the operating system for a small, specialized system, one that provides only a single service, or a set of closely relateed services. Almost an embedded system, this sort of server is an increasingly important part of Linux development. One familiar system of this sort is the TiVo Personal Video Recorder.

Tony Mancill's book is about a different sort of specialized use of Linux: turning old, off-the-shelf computers into specialized routers. A router is a device that connects two or more networks. Most readers here have likely encountered routers as the devices that connect small, home and office Local Area Networks (LANs) to an Internet Service Provider (ISP) and then to the Internet. Such "edge routers" are the best known use of small routers, and many commercial products exist, including inexpensive ones from companies such as Linksys, Netgear, and D-Link, and expensive ones from Cisco and others.

Through a series of case studies, Mancill tells you how to use older PCs to perform the principal functions of commercial edge routers, as well as how to provide additional, related capabilities, such as firewalling, NAT'ing (Network Address Translation, a method of letting several computers on a LAN "share" a single IP address assigned by an ISP), traffic monitoring, and some elements of LAN management. He goes well beyond the simplest edge-router applications, though, including cases in which he used Linux-based routers to interconnect multiple LANs, to interconnect older types of LANs to modern ones (one case involves connecting a Token Ring LAN to an Ethernet), and to provide Virtual Private Network (VPN) connections among a company's scattered offices.

Mancill uses the case-study approach to introduce and explain just about every major routing-related capability of the Linux kernel and its common application programs. Each case study focuses on solving an actual routing problem Mancill faced in his work, an approach that keeps all the explanation grounded in real, practical solutions to actual problems. And Mancill knows his stuff -- he is familiar with the details of Linux from the ground up, he writes both informatively and entertainingly, and he is realistic about the kinds of decisions and trade-offs real systems administrators need to make in real situations.

The book mainly consists of seven case studies, which he names after elements from the Periodic Table:

Silicon -- This is a LAN-to-LAN router that he uses as the framework for introducing core routing concepts and explaining the use of several traffic-analysis utilities.

Erbium -- This is a LAN-to-Internet router that uses a dial-up (modem) connection to an ISP. He uses it to explain the use of Network Address Translation (NAT) to "share" a single public IP address and to explain the process of maintaining a persistent connection using dial-up service.

Zinc -- This is a LAN-to-Internet router that uses a Frame Relay (FR) connection to an ISP. He uses it to explain the use of Linux with the specialized hardware needed to access FR service and other services that use DS-1 links.

Cesium and Xenon -- They are a pair of routers that linked two Token-Ring LANs over an intervening Ethernet. He uses them to explain the procdeure for setting up and maintaining VPNs with the IPSec (IP Secure) capabilities of Linux.

Oxygen -- Another LAN-to-Internet router, this one using synchronous PPP over a dedicated line to the ISP. He uses it to develop concepts of network security, including firewalling, intrusion detection, and vulnerability testing.

Californium -- One more LAN-to-Internet router, this time one designed for use by satellite offices of a company, so particularly optimized for remote management. He uses it to discuss tools for traffic "shaping" -- in essence, giving some sorts of uses priority over others -- and to explore the limits of how much additional work a typical router can do.

Hafnium -- The last LAN-to-Internet router. Here he introduces the concept of a proxy server and continues the discussion of using a router's excess capacity to provide other services, such as DNS resolution and e-mail forwarding.

In the end, it is hard to say enough good things about this book. Anyone who wants to develop a good understanding of the intricacies of routing, and of how Linux can be used to address tricky routing problems in cost-effective ways, will benefit from reading it. But one big caution is in order -- despite the practical focus of the case studies, this is in no sense a cookbook. It offers no "10 quick steps to a router" sorts of answers; it is very much not a book for "dummies". The case studies provide a great basis for making the discussion of concepts concrete, but does not provide simple recipes for solving typical problems. In part, this limitation derives from Mancill's having drawn his examples from many years of his own experience. As a result, they cover older forms of connectivity well. But there is no case study covering the most common type of broadband connection being installed today -- connection to an ISP through a DSL or cable modem that connects to the router via Ethernet or USB. Linux can easily handle this sort of connection too -- I am one of many Linux users who do it -- but Mancill's cases do not include this sort of setup.

Other hesitations about the book are really quibbles. The case study approach is great for cover-to-cover reading of the book, but it is a challenge when using it as a reference. In particular, it asks more of the book's index than this one delivers. And the book says too little about Linux distributions that are customized to specialized uses like routing; it refers from time to time to the all-but-defunct, obsolete Linux Router Project but fails to mention more active successor sites like LEAF (http://leaf.sourceforge.net/ (http://leaf.sourceforge.net/)). But as I said, these are quibbles, at least in comparison to the overall high quality of the book's coverage of almost everything you need to know about routing. Beginners will learn a lot from it. Experienced network managers will find it a handy tool for refreshing the memory. Me, I'll be sure to shelve my copy in easy reach.

Ray Olszewski (ray@comarre.com (mailto:ray@comarre.com)) is a freelance computer programmer, focused on embedded systems design using Windows- and Linux-based development tools. He has also worked as a statistician and he moonlights as an Open Source software developer. His past work includes development of custom Web-based software to support online research.

REVIEW OF: Theo Peterson (2002). Web Development with Apache and Perl. Greenwich, CT: Manning.

by Brian K. Yost

Web Development with Apache and Perl provides a survey of available Open Source tools for creating advanced Web sites. This is not a book for absolute beginners. Although the examples in each chapter start at a basic level and progress in complexity, neophytes would soon become overwhelmed. Peterson states in the Preface: "Teaching the basics is beyond the scope of this book; and besides, I couldn't do a better job than many texts already available to help you learn" (p. xiv). This said, with some Unix and Perl knowledge, users will be able to learn a great deal from the examples used in this title.

Peterson begins the book by arguing the case for using Open Source software. He claims Open Source software is superior in terms of support, quality, security, and innovation. After making the case for Open Source in general, Peterson discusses Apache and CGI scripting with Perl in-depth.

In Part 2 Peterson surveys available tools for Web applications, covering databases, scripting tools, security, and combining Perl with HTML. Part 3 covers the various functions of Web sites by exploring sample virtual communities, intranet applications, and e-commerce. Peterson concludes the book with a discussion of site management issues covering content and performance management.

Throughout the book, Peterson provides "buyer's guides," comparisons of products available for various functions. For example, there are comparisons of Web servers, scripting languages, and databases. Although Peterson clearly prefers Open Source tools, he provides well informed and balanced comparisons that readers making software decisions for their Web projects will find very useful.

An author of a survey book such as this must make many decisions about what to include and exclude. This is an area in which *Web Development with Apache and Perl* excels. Peterson and his editors have done a great job of including the most relevant information and excluding extraneous details. For a title that is intended as an "overview," there is an amazing amount of very practical information. For example, in chapter 2, Peterson provides Apache httpd.conf options for various scenarios. One of these includes allowing users to run CGI scripts from their own home directories. Peterson's examples provide solutions for common scenarios that are often missing from the supposedly "complete" Apache books.

I enjoyed the author's writing and presentation style. It is clear that he is knowledgeable and an expert on the topics covered. Peterson takes a conversational approach to writing that makes reading the book similar to having a conversation with an expert. It is also clear that a great deal of effort went into the organization of the book. The chapters flow very well from one topic into the next.

The publisher of this title, Manning, provides a Web forum in which the author participates. Although there are not a lot of questions posted in the forum, the author seems to respond promptly and thoroughly. This is an interesting addition that allows for interaction between readers and the author of a book.

So who would benefit from this book? It is not a comprehensive tutorial on Perl and Apache. On the other hand, it goes beyond being an overview of Open Source Web development tools. I would recommend this book to anyone who is interested in learning about the tools available to build Web sites that go beyond simple static pages.

Brian K. Yost (yostb@hope.edu (mailto:yostb@hope.edu)) is Systems Librarian at Hope College in Holland, Michigan.

REVIEW OF: Erik Wilde and David Lowe (2003). XPath, XLink, XPointer, and XML: A Practical Guide to Web Hyperlinking and Transclusion. Boston: Addison-Wesley.

by Brad Eden

A wide range of exciting possibilities in the realm of Internet content management is now available with the emergence of the eXtensible Markup Language (XML) and its related interlinking standards. The authors have written a book that contains practical guidelines for application of these standards, as well as an advanced reference work documenting historical and potential use of these standards within the worlds of Web site management and Web publishing.

The book is divided into a preface, introduction, and four parts. The preface provides the authors' purpose for the book, as well as its audience, which includes Web users, students, Web authors, Web developers, and Web project managers. The introduction provides a short historical account of the development of the Web, HTML, and XML.

Part I focuses on the conceptual framework of the Web, considering current technology and the limitations of that technology. It then moves into an account of types of changes that will occur on the Web in the future, including the possibilities that linking issues might provide, as well as a scenario of what this might look like. The authors then move into a discussion of XPath, XPointer, and XLink from a conceptual viewpoint, examining how these standards will provide the types of sophisticated linking and content management needed on the Web in the future.

Part II describes specific details of a number of the new technologies that will assist in the realization of this future. XML Infoset, XHTML, XSLT, XSL, RDF, XML Namespaces, XML Base, XInclude, XSL-FO are all given space in this section. The focus is then moved to the power of the three standards provided in the title, and the authors make sure that the reader understands the current status of these standards, which version they are describing, and where to go to maintain currency of information.

Part III examines how these technologies will help to move the vision forward as described in Part I. Authoring applications using these standards are discussed, issues that still need to be addressed are debated, and then final comments are given.

The word "transclusion" is defined and identified in the authors' preface, and the reader should make sure to understand this term before beginning the book, as it is a relatively new concept. An extensive index is provided with the book as well. Topics covered in depth, for those wishing a short list, include: hypermedia concepts and alternatives to the Web; XML Namespaces, XML Base, XInclude, XML Information Set, XHTML, and XSLT; XPath, XLink, and XPointer concepts, strengths, and limitations; emerging tools, applications, and environments; migration strategies, from conventional models to more sophisticated linking techniques; and future perspectives on the XPath, XLink, and XPointer standards.

I found the book an excellent current guide to XML and its linking standards. Its explanations are geared toward the beginning user, while its practical applications section is sophisticated enough for the medium to advanced developer. A very useful and overall well-written book.

Brad Eden, Ph.D., Head, Bibliographic and Metadata Services, University of Nevada, Las Vegas, beden@ccmail.nevada.edu (mailto:beden@ccmail.nevada.edu)..