Resource Oriented Architecture Patterns For Webs Of Data Synthesis Lectures On The Semantic Web Theory And Technology

This book explains how to combine and exploit sensor networks and internet-of-things (IoT) technologies and Web-service design patterns to enrich and integrate Building Information Models (BIMs). It provides approaches and software architectures for facilitating the interaction with (and between) BIMs through Web services, and for enabling and facilitating the fusion of the information residing in such models or of information acquired from IoT technologies. The proposed software architectures are presented in the form of design patterns. This information fusion will facilitate many novel application fields ranging from emergency response, to urban monitoring and surveillance, and to smart buildings. The book consists of 8 chapters. The first 2 chapters focus on the basics of BIMs, while chapter 3 presents fundamental service-oriented architecture patterns for complex information models. Subsequently, chapters 4 and 5 elaborate on the hardware and software side of IoT, with a special focus on their use for BIMs. Chapter 6 provides advanced SOA patterns for BIMs, while chapter 7 details patterns for IoT, and

for BIM and IoT information fusion. Lastly, chapter 8 summarizes the work and provides an outlook on promising future developments. Overall, the book will be beneficial for researchers and developers in the fields of building information models, IoT applications, and systems integration. Linked Data (LD) is a well-established standard for publishing and managing structured information on the Web, gathering and bridging together knowledge from different scientific and commercial domains. The development of Linked Data Visualization techniques and tools has been followed as the primary means for the analysis of this vast amount of information by data scientists, domain experts, business users, and citizens. This book covers a wide spectrum of visualization issues, providing an overview of the recent advances in this area, focusing on techniques, tools, and use cases of visualization and visual analysis of LD. It presents the basic concepts related to data visualization and the LD technologies, the techniques employed for data visualization based on the characteristics of data techniques for Big Data visualization, use tools and use cases in the LD context, and finally a thorough assessment of the usability of these tools under different scenarios. The purpose of this book is to offer a complete guide to the evolution of LD visualization for interested readers from any background and to empower them to get started

with the visual analysis of such data. This book can serve as a course textbook or a primer for all those interested in LD and data visualization.

The practice of enterprise application development has benefited from the emergence of many new enabling technologies. Multi-tiered object-oriented platforms, such as Java and .NET, have become commonplace. These new tools and technologies are capable of building powerful applications, but they are not easily implemented. Common failures in enterprise applications often occur because their developers do not understand the architectural lessons that experienced object developers have learned. Patterns of Enterprise Application Architecture is written in direct response to the stiff challenges that face enterprise application developers. The author, noted object-oriented designer Martin Fowler, noticed that despite changes in technology--from Smalltalk to CORBA to Java to .NET--the same basic design ideas can be adapted and applied to solve common problems. With the help of an expert group of contributors, Martin distills over forty recurring solutions into patterns. The result is an indispensable handbook of solutions that are applicable to any enterprise application platform. This book is actually two books in one. The first section is a short tutorial on developing enterprise applications, which you can read from start to finish to understand the scope of the

book's lessons. The next section, the bulk of the book, is a detailed reference to the patterns themselves. Each pattern provides usage and implementation information, as well as detailed code examples in Java or C#. The entire book is also richly illustrated with UML diagrams to further explain the concepts. Armed with this book, you will have the knowledge necessary to make important architectural decisions about building an enterprise application and the proven patterns for use when building them. The topics covered include · Dividing an enterprise application into layers · The major approaches to organizing business logic · An in-depth treatment of mapping between objects and relational databases · Using Model-View-Controller to organize a Web presentation · Handling concurrency for data that spans multiple transactions · Designing distributed object interfaces Business and IT organizations are currently embracing new strategically sound concepts in order to be more customer-centric, competitive, and cognitive in their daily operations. While useful, the various software tools, pioneering technologies, as well as their unique contributions largely go unused due to the lack of information provided on their special characteristics. Novel Practices and Trends in Grid and Cloud Computing is a collection of innovative research on the key concerns of cloud computing and how they are being addressed,

as well as the various technologies and tools empowering cloud theory to be participative, penetrative, pervasive, and persuasive. While highlighting topics including cyber security, smart technology, and artificial intelligence, this book is ideally designed for students, researchers, and business managers on the lookout for innovative IT solutions for all the business automation software and improvisations of computational technologies.
Enterprise Software Architecture and Design
Validating RDF Data
Fowler
Patterns for Resource Management
Architectural Patterns
97 Things Every Software Architect Should Know

Learn the importance of architectural and design patterns in producing and sustaining next-generation IT and business-critical applications with this guide. Key Features Use patterns to tackle communication, integration, application structure, and more Implement modern design patterns such as microservices to build resilient and highly available applications Choose between the MVP, MVC, and MVVM patterns depending on the application being built Book Description Enterprise Architecture (EA) is typically an aggregate of the business, application, data,

and infrastructure architectures of any forward-looking enterprise. Due to constant changes and rising complexities in the business and technology landscapes, producing sophisticated architectures is on the rise. Architectural patterns are gaining a lot of attention these days. The book is divided in three modules. You'll learn about the patterns associated with object-oriented, component-based, client-server, and cloud architectures. The second module covers Enterprise Application Integration (EAI) patterns and how they are architected using various tools and patterns. You will come across patterns for Service-Oriented Architecture (SOA), Event-Driven Architecture (EDA), Resource-Oriented Architecture (ROA), big data analytics architecture, and Microservices Architecture (MSA). The final module talks about advanced topics such as Docker containers, high performance, and reliable application architectures. The key takeaways include understanding what architectures are, why they're used, and how and where architecture, design, and integration patterns are being leveraged to build better and bigger systems. What you will learn Understand how several architectural and design patterns work to

systematically develop multitier web, mobile, embedded, and cloud applications Learn object-oriented and component-based software engineering principles and patterns Explore the frameworks corresponding to various architectural patterns Implement domain-driven, test-driven, and behavior-driven methodologies Deploy key platforms and tools effectively to enable EA design and solutioning Implement various patterns designed for the cloud paradigm Who this book is for This book will empower and enrich IT architects (such as enterprise architects, software product architects, and solution and system architects), technical consultants, evangelists, and experts.

This proceedings volume contains selected revised and extended research articles written by researchers who participated in the World Congress on Engineering and Computer Science 2015, held in San Francisco, USA, 21-23 October 2015. Topics covered include engineering mathematics, electrical engineering, circuits, communications systems, computer science, chemical engineering, systems engineering, manufacturing engineering, and industrial applications. The book offers the reader an overview of the state of the art in engineering

technologies, computer science, systems engineering and applications, and will serve as an excellent reference work for researchers and graduate students working in these fields.

Many of the initial developments towards the Internet of Things have focused on the combination of Auto-ID and networked infrastructures in business-to-business logistics and product lifecycle applications. However, the Internet of Things is more than a business tool for managing business processes more efficiently and more effectively – it will also enable a more convenient way of life. Since the term Internet of Things first came to attention when the Auto-ID Center launched their initial vision for the EPC network for automatically identifying and tracing the flow of goods within supply-chains, increasing numbers of researchers and practitioners have further developed this vision. The authors in this book provide a research perspective on current and future developments in the Internet of Things. The different chapters cover a broad range of topics from system design aspects and core architectural approaches to end-user participation, business perspectives and

applications.

The dramatic progress of smartphone technologies has ushered in a new era of mobile sensing, where traditional wearable on-body sensors are being rapidly superseded by various embedded sensors in our smartphones. For example, a typical smartphone today, has at the very least a GPS, WiFi, Bluetooth, triaxial accelerometer, and gyroscope. Alongside, new accessories are emerging such as proximity, magnetometer, barometer, temperature, and pressure sensors. Even the default microphone can act as an acoustic sensor to track noise exposure for example. These sensors act as a ""lens"" to understand the user's context along different dimensions. Data can be passively collected from these sensors without interrupting the user. As a result, this new era of mobile sensing has fueled significant interest in understanding what can be extracted from such sensor data both instantaneously as well as considering volumes of time series from these sensors. For example, GPS logs can be used to determine automatically the significant places associated to a user's life (e.g., home, office, shopping areas). The logs may also reveal travel patterns, and how a user moves

from one place to another (e.g., driving or using public transport). These may be used to proactively inform the user about delays, relevant promotions from shops, in his ""regular"" route. Similarly, accelerometer logs can be used to measure a user's average walking speed, compute step counts, gait identification, and estimate calories burnt per day. The key objective is to provide better services to end users. The objective of this book is to inform the reader of the methodologies and techniques for extracting meaningful information (called ""semantics"") from sensors on our smartphones. These techniques form the cornerstone of several application areas utilizing smartphone sensor data. We discuss technical challenges and algorithmic solutions for modeling and mining knowledge from smartphone-resident sensor data streams. This book devotes two chapters to dive deep into a set of highly available, commoditized sensors---the positioning sensor (GPS) and motion sensor (accelerometer). Furthermore, this book has a chapter devoted to energy-efficient computation of semantics, as battery life is a major concern on user experience. Demystifying OWL for the Enterprise

Entity Resolution in the Web of Data
RESTful Web Services
Semantics in Mobile Sensing
World Congress on Engineering and Computer Science 2015
Entities, Services, and Resources

Designing application software to run in distributed and concurrent environments is a challenge facing software developers. These patterns form the basis of a pattern language that address issues of distribution, concurrency and networking.

WebAssembly: The Definitive Guide provides a thorough and accessible introduction to one of the most transformative technologies hitting our industry. What started as a way to use languages other than just JavaScript in the browser has evolved into a comprehensive path toward portability, performance, increased security and greater code reuse across an impressive collection of deployment targets. The goals may sound familiar, but in practice, we're finally getting our safe, fast, portable, and secure software development environment offering the potential for reuse. This practical book introduces the elements of this technology incrementally while building to several concrete, code-driven examples of practical but cutting edge WebAssembly uses.

This book fills a gap between high–level overview texts that are often too general and low–level detail oriented technical handbooks that lose sight the "big picture". This book discusses SOA from the low–level perspective of middleware, various XML–based technologies, and basic service design. It also examines broader implications of SOA, particularly where it intersects with

business process management and process modeling. Concrete overviews will be provided of the methodologies in those fields, so that students will have a hands–on grasp of how they may be used in the context of SOA.

The surge of interest in the REpresentational State Transfer (REST) architectural style, the Semantic Web, and Linked Data has resulted in the development of innovative, flexible, and powerful systems that embrace one or more of these compatible technologies. However, most developers, architects, Information Technology managers, and platform owners have only been exposed to the basics of resource-oriented architectures. This book is an attempt to catalog and elucidate several reusable solutions that have been seen in the wild in the now increasingly familiar "patterns book" style. These are not turn key implementations, but rather, useful strategies for solving certain problems in the development of modern, resource-oriented systems, both on the public Web and within an organization's firewalls.
Provenance
Techniques, Tools, and Big Data
Collective Wisdom from the Experts
Introduction to Middleware
Pattern-Oriented Software Architecture, Patterns for Resource Management
Uncover essential patterns in the most indispensable realm of enterprise architecture

Web browsing would not be what it is today without the use of Service-Oriented Architecture (SOA). Although much has been written about SOA methodology, this emerging platform is continuously under development. Exploring Enterprise Service Bus in the Service-Oriented

Architecture Paradigm is a detailed reference source that examines current aspects and research methodologies that enable enterprise service bus to unify and connect services efficiently on a common platform. Featuring relevant topics such as SOA reference architecture, grid computing applications, complex event computing, and java business integration, this is an ideal resource for all practitioners, academicians, graduate students, and researchers interested in the discoveries on the relationship that Service-Oriented architecture and enterprise service bus share.

Learn the importance of architectural and design patterns in producing and sustaining next-generation IT and business-critical applications with this guide. About This Book Use patterns to tackle communication, integration, application structure, and more Implement modern design patterns such as microservices to build resilient and highly available applications Choose between the MVP, MVC, and MVVM patterns depending on the application being built Who This Book Is For This book will empower and enrich IT architects (such as enterprise architects, software product architects, and solution and system architects), technical consultants, evangelists, and experts. What You Will Learn Understand how several architectural and design patterns work to systematically develop multitier web, mobile, embedded, and cloud applications Learn object-oriented and component-based software engineering principles and patterns Explore the frameworks corresponding to various architectural patterns Implement domain-driven, test-driven, and behavior-

driven methodologies Deploy key platforms and tools effectively to enable EA design and solutioning Implement various patterns designed for the cloud paradigm In Detail Enterprise Architecture (EA) is typically an aggregate of the business, application, data, and infrastructure architectures of any forward-looking enterprise. Due to constant changes and rising complexities in the business and technology landscapes, producing sophisticated architectures is on the rise. Architectural patterns are gaining a lot of attention these days. The book is divided in three modules. You'll learn about the patterns associated with object-oriented, component-based, client-server, and cloud architectures. The second module covers Enterprise Application Integration (EAI) patterns and how they are architected using various tools and patterns. You will come across patterns for Service-Oriented Architecture (SOA), Event-Driven Architecture (EDA), Resource-Oriented Architecture (ROA), big data analytics architecture, and Microservices Architecture (MSA). The final module talks about advanced topics such as Docker containers, high performance, and reliable application architectures. The key takeaways include understanding what architectures are, why they're used, and how and where architecture, design, and integration patterns are being leveraged to build better and bigger systems. Style and Approach This book adopts a hands-on approach with real-world examples and use cases.

The current drug development paradigm---sometimes expressed as, ``One disease, one target, one drug''---is under question, as relatively few drugs have reached the

market in the last two decades. Meanwhile, the research focus of drug discovery is being placed on the study of drug action on biological systems as a whole, rather than on individual components of such systems. The vast amount of biological information about genes and proteins and their modulation by small molecules is pushing drug discovery to its next critical steps, involving the integration of chemical knowledge with these biological databases. Systematic integration of these heterogeneous datasets and the provision of algorithms to mine the integrated datasets would enable investigation of the complex mechanisms of drug action; however, traditional approaches face challenges in the representation and integration of multi-scale datasets, and in the discovery of underlying knowledge in the integrated datasets. The Semantic Web, envisioned to enable machines to understand and respond to complex human requests and to retrieve relevant, yet distributed, data, has the potential to trigger system-level chemical-biological innovations. Chem2Bio2RDF is presented as an example of utilizing Semantic Web technologies to enable intelligent analyses for drug discovery.Table of Contents: Introduction /Data Representation and Integration Using RDF /Data Representation and Integration Using OWL /Finding Complex Biological Relationships in PubMed Articles using Bio-LDA / Integrated Semantic Approach for Systems Chemical Biology Knowledge Discovery /Semantic Link Association Prediction /Conclusions /References / Authors' Biographies

Build straightforward and maintainable APIs to create

services that are usable and maintainable. Although this book focuses on distributed services, it also emphasizes how the core principles apply even to pure OOD and OOP constructs. The overall context of Creating Maintainable APIs is to classify the topics into four main areas: classes and interfaces, HTTP REST APIs, messaging APIs, and message payloads (XML, JSON and JSON API as well as Apache Avro). What You Will Learn Use object-oriented design constructs and their APIs Create and manage HTTP REST APIs Build and manage maintainable messaging APIs, including the use of Apache Kafka as a principal messaging hub Handle message payloads via JSON Who This Book Is For Any level software engineers and very experienced programmers.

RESTful .NET

Using IoT Services and Integration Patterns

Semantic Breakthrough in Drug Discovery

The Epistemology of Intelligent Semantic Web Systems

Migrating Legacy Applications: Challenges in Service Oriented Architecture and Cloud Computing Environments

Data Management at Scale

The Semantic Web is a young discipline, even if only in comparison to other areas of computer science. Nonetheless, it already exhibits an interesting history and evolution. This book is a reflection on this evolution, aiming to take a snapshot of where we are at this specific point in time, and also showing what might be the focus of future research. This book provides both a

**conceptual and practical view of this evolution, especially targeted at readers who are starting research in this area and as support material for their supervisors. From a conceptual point of view, it highlights and discusses key questions that have animated the research community: what does it mean to be a Semantic Web system and how is it different from other types of systems, such as knowledge systems or web-based information systems? From a more practical point of view, the core of the book introduces a simple conceptual framework which characterizes Intelligent Semantic Web Systems. We describe this framework, the components it includes, and give pointers to some of the approaches and technologies that might be used to implement them. We also look in detail at concrete systems falling under the category of Intelligent Semantic Web Systems, according to the proposed framework, allowing us to compare them, analyze their strengths and weaknesses, and identify the key fundamental challenges still open for researchers to tackle.
RESTful .NET is the first book that teaches Windows developers to build RESTful web services using the latest Microsoft tools. Written by Windows Communication Foundation (WFC) expert Jon Flanders, this hands-on tutorial demonstrates how you can use WCF and other components of the .NET 3.5 Framework to build, deploy and use REST-based web services in a**

variety of application scenarios. RESTful architecture offers a simpler approach to building web services than SOAP, SOA, and the cumbersome WS- stack. And WCF has proven to be a flexible technology for building distributed systems not necessarily tied to WS- standards. RESTful .NET provides you with a complete guide to the WCF REST programming model for building web services consumed either by machines or humans. You'll learn how to: Program Read-Only (GET) services Program READ/WRITE services Host REST services Program REST feeds Program AJAX REST clients Secure REST endpoints Use workflow to deliver REST services Consume RESTful XML services using WCF Work with HTTP Work with ADO.NET Data Services (Astoria) RESTful .NET introduces you to the ideas of REST and RESTful architecture, and includes a detailed discussion of how the Web/REST model plugs into the WCF architecture. If you develop with .NET, it's time to jump on the RESTful bandwagon. This book explains how. "While REST is simple, WCF is not. To really understand and exploit this part of WCF requires a knowledgeable and experienced guide. I don't know anybody who's better suited for this role than Jon Flanders. ...Jon is first-rate at explaining complicated things. This book is the best introduction I've seen to creating and using these services with WCF."--David Chappell, Chappell & Associates

This book provides a comprehensive and accessible introduction to knowledge graphs, which have recently garnered notable attention from both industry and academia. Knowledge graphs are founded on the principle of applying a graph-based abstraction to data, and are now broadly deployed in scenarios that require integrating and extracting value from multiple, diverse sources of data at large scale. The book defines knowledge graphs and provides a high-level overview of how they are used. It presents and contrasts popular graph models that are commonly used to represent data as graphs, and the languages by which they can be queried before describing how the resulting data graph can be enhanced with notions of schema, identity, and context. The book discusses how ontologies and rules can be used to encode knowledge as well as how inductive techniques—based on statistics, graph analytics, machine learning, etc.—can be used to encode and extract knowledge. It covers techniques for the creation, enrichment, assessment, and refinement of knowledge graphs and surveys recent open and enterprise knowledge graphs and the industries or applications within which they have been most widely adopted. The book closes by discussing the current limitations and future directions along which knowledge graphs are likely to evolve. This book is aimed at students, researchers, and practitioners who

wish to learn more about knowledge graphs and how they facilitate extracting value from diverse data at large scale. To make the book accessible for newcomers, running examples and graphical notation are used throughout. Formal definitions and extensive references are also provided for those who opt to delve more deeply into specific topics.

As data management and integration continue to evolve rapidly, storing all your data in one place, such as a data warehouse, is no longer scalable. In the very near future, data will need to be distributed and available for several technological solutions. With this practical book, you'll learnhow to migrate your enterprise from a complex and tightly coupled data landscape to a more flexible architecture ready for the modern world of data consumption. Executives, data architects, analytics teams, and compliance and governance staff will learn how to build a modern scalable data landscape using the Scaled Architecture, which you can introduce incrementally without a large upfront investment. Author Piethein Strengholt provides blueprints, principles, observations, best practices, and patterns to get you up to speed. Examine data management trends, including technological developments, regulatory requirements, and privacy concerns Go deep into the Scaled Architecture and learn how the pieces fit together Explore data governance and data

security, master data management, self-service data marketplaces, and the importance of metadata
Pattern Enterpr Applica Arch
A Practical, Case-Study Approach
Build and Consume RESTful Web Services with .NET 3.5
Knowledge Graphs
Pattern-Oriented Software Architecture, Patterns for Concurrent and Networked Objects
Creating Maintainable APIs

"Every developer working with the Web needs to read this book." -- David Heinemeier Hansson, creator of the Rails framework "RESTful Web Services finally provides a practical roadmap for constructing services that embrace the Web, instead of trying to route around it." -- Adam Trachtenberg, PHP author and EBay Web Services Evangelist You've built web sites that can be used by humans. But can you also build web sites that are usable by machines? That's where the future lies, and that's what RESTful Web Services shows you how to do. The World Wide Web is the most popular distributed application in history, and Web services and mashups have turned it

into a powerful distributed computing platform. But today's web service technologies have lost sight of the simplicity that made the Web successful. They don't work like the Web, and they're missing out on its advantages. This book puts the "Web" back into web services. It shows how you can connect to the programmable web with the technologies you already use every day. The key is REST, the architectural style that drives the Web. This book: Emphasizes the power of basic Web technologies -- the HTTP application protocol, the URI naming standard, and the XML markup language Introduces the Resource-Oriented Architecture (ROA), a common-sense set of rules for designing RESTful web services Shows how a RESTful design is simpler, more versatile, and more scalable than a design based on Remote Procedure Calls (RPC) Includes real-world examples of RESTful web services, like Amazon's Simple Storage Service and the Atom Publishing Protocol Discusses web service clients for popular programming languages Shows how to implement RESTful services in three

popular frameworks -- Ruby on Rails, Restlet (for Java), and Django (for Python) Focuses on practical issues: how to design and implement RESTful web services and clients This is the first book that applies the REST design philosophy to real web services. It sets down the best practices you need to make your design a success, and the techniques you need to turn your design into working code. You can harness the power of the Web for programmable applications: you just have to work with the Web instead of against it. This book shows you how.

This book describes a set of methods, architectures, and tools to extend the data pipeline at the disposal of developers when they need to publish and consume data from Knowledge Graphs (graph-structured knowledge bases that describe the entities and relations within a domain in a semantically meaningful way) using SPARQL, Web APIs, and JSON. To do so, it focuses on the paradigmatic cases of two middleware software packages, grlc and SPARQL Transformer, which automatically build and run SPARQL-based REST APIs and

allow the specification of JSON schema results, respectively. The authors highlight the underlying principles behind these technologies—query management, declarative languages, new levels of indirection, abstraction layers, and separation of concerns—, explain their practical usage, and describe their penetration in research projects and industry. The book, therefore, serves a double purpose: to provide a sound and technical description of tools and methods at the disposal of publishers and developers to quickly deploy and consume Web Data APIs on top of Knowledge Graphs; and to propose an extensible and heterogeneous Knowledge Graph access infrastructure that accommodates a growing ecosystem of querying paradigms.

"This book presents a closer look at the partnership between service oriented architecture and cloud computing environments while analyzing potential solutions to challenges related to the migration of legacy applications"--Provided by publisher. The Definitive Guide to Building Web-Centric SOA with REST The World Wide

Web is based on the most successful technology architecture in history. It has changed how we view, access, and exchange information and, with the advent of REST, it has also provided us with compelling ways to build and improve automation solutions. REST provides a great deal of guidance to ensure that an architecture and its automation logic are technically sound, though it is still your responsibility to build services that actually add value to your business. SOA with REST is the first comprehensive tutorial and reference for designing and building RESTful services as part of service-oriented solutions and in conjunction with service-oriented architecture (SOA). This book demonstrates that REST is not only a suitable medium for building truly service-oriented solutions, but also that the service-oriented architectural model is a necessary foundation for REST technology architectures to realize their full business potential. The authors provide thorough mapping of REST constraints and architectural goals with service-orientation

/Online Library Resource Oriented Architecture

principles and SOA characteristics.
Using real-world examples, they show
how to leverage REST's simplicity,
flexibility, and low overhead without
compromising the power or manageability
of service-oriented solutions and
architectures. This ebook will be
valuable to IT architects, developers,
and any practitioner seeking to use SOA
and REST together.
Natural Language Processing for the
Semantic Web
WebAssembly: The Definitive Guide
Design, develop, and deploy highly
adaptable, scalable, and secure RESTful
web APIs
SOA with REST
Pattern-Oriented Software Architecture,
A System of Patterns
Exploring Enterprise Service Bus in the
Service-Oriented Architecture Paradigm
*The first volume of the POSA pattern series introduced
a broad-spectrum of general-purpose patterns in
software design and architecture. The second narrowed
the focus to fundamental patterns for building
sophisticated concurrent and networked software
systems and applications. This volume uses design
patterns to present techniques for implementing
effective resource management in a system. The*

patterns are covered in detail making use of several examples providing directions to the readers on how to implement the presented patterns. Additionally, the volume presents a thorough introduction into resource management and a case study where the patterns are applied to the domain of mobile radio networks. The patterns are grouped by different areas of resource management and hence address the complete lifecycle of resources: resource acquisition, coordination and release.

In cooperation with experts and practitioners throughout the SOA community, best-selling author Thomas Erl brings together the de facto catalog of design patterns for SOA and service-orientation. More than three years in development and subjected to numerous industry reviews, the 85 patterns in this full-color book provide the most successful and proven design techniques to overcoming the most common and critical problems to achieving modern-day SOA. Through numerous examples, individually documented pattern profiles, and over 400 color illustrations, this book provides in-depth coverage of: • Patterns for the design, implementation, and governance of service inventories–collections of services representing individual service portfolios that can be independently modeled, designed, and evolved. • Patterns specific to service-level architecture which pertain to a wide range of design areas, including contract design, security, legacy encapsulation, reliability, scalability, and a variety of implementation and governance issues. • Service composition patterns that address the many

aspects associated with combining services into aggregate distributed solutions, including topics such as runtime messaging and message design, inter-service security controls, and transformation. • Compound patterns (such as Enterprise Service Bus and Orchestration) and recommended pattern application sequences that establish foundational processes. The book begins by establishing SOA types that are referenced throughout the patterns and then form the basis of a final chapter that discusses the architectural impact of service-oriented computing in general. These chapters bookend the pattern catalog to provide a clear link between SOA design patterns, the strategic goals of service-oriented computing, different SOA types, and the service-orientation design paradigm. This book series is further supported by a series of resources sites, including soabooks.com, soaspecs.com, soapatterns.org, soamag.com, and soaposters.com.

The surge of interest in the REpresentational State Transfer (REST) architectural style, the Semantic Web, and Linked Data has resulted in the development of innovative, flexible, and powerful systems that embrace one or more of these compatible technologies. However, most developers, architects, Information Technology managers, and platform owners have only been exposed to the basics of resource-oriented architectures. This book is an attempt to catalog and elucidate several reusable solutions that have been seen in the wild in the now increasingly familiar "patterns book" style. These are

not turn key implementations, but rather, useful strategies for solving certain problems in the development of modern, resource-oriented systems, both on the public Web and within an organization's firewalls. Table of Contents: List of Figures / Informational Patterns / Applicative Patterns / Procedural Patterns

Designing application and middleware software to run in concurrent and networked environments is a significant challenge to software developers. The patterns catalogued in this second volume of Pattern-Oriented Software Architectures (POSA) form the basis of a pattern language that addresses issues associated with concurrency and networking. The book presents 17 interrelated patterns ranging from idioms through architectural designs. They cover core elements of building concurrent and network systems: service access and configuration, event handling, synchronization, and concurrency. All patterns present extensive examples and known uses in multiple programming languages, including C++, C, and Java. The book can be used to tackle specific software development problems or read from cover to cover to provide a fundamental understanding of the best practices for constructing concurrent and networked applications and middleware. About the Authors This book has been written by the award winning team responsible for the first POSA volume "A System of Patterns", joined in this volume by Douglas C. Schmidt from University of California, Irvine (UCI), USA. Visit our Web Page

Linked Data Visualization
Uncover Essential Patterns in the Most Indispensable Realm of Enterprise Architecture
Resource-Oriented Architecture Patterns for Webs of Data
Pattern-oriented Software Architecture
Web Services, Object Components, and Cloud Computing
Principles, Patterns & Constraints for Building Enterprise Solutions with REST

Pattern-oriented software architecture is a new approach to software development. This book represents the progression and evolution of the pattern approach into a system of patterns capable of describing and documenting large-scale applications. A pattern system provides, on one level, a pool of proven solutions to many recurring design problems. On another it shows how to combine individual patterns into heterogeneous structures and as such it can be used to facilitate a constructive development of software systems. Uniquely, the patterns that are presented in this book span several levels of abstraction, from high-level architectural patterns and medium-level design patterns to low-level idioms. The intention of, and motivation for, this book is to support both novices and experts in software development. Novices will gain from the experience inherent in pattern descriptions and experts will hopefully make use of, add to, extend and modify patterns to tailor them to their own needs. None of the pattern descriptions are cast in stone and, just as they are borne from experience, it is expected that further use will feed

in and refine individual patterns and produce an evolving system of patterns. Visit our Web Page http://www.wiley.com/compbooks/

"A stereotype of computer science textbooks is that they are dry, boring, and sometimes even intimidating. As a result, they turn students' interests off from the subject matter instead of enticing them into it. This textbook is the opposite of such a stereotype. The author presents the subject matter in a refreshing story-telling style and aims to bring the Internet-generation of students closer to her stories." --Yingcai Xiao, The University of Akron Introduction to Middleware: Web Services, Object Components, and Cloud Computing provides a comparison of different middleware technologies and the overarching middleware concepts they are based on. The various major paradigms of middleware are introduced and their pros and cons are discussed. This includes modern cloud interfaces, including the utility of Service Oriented Architectures. The text discusses pros and cons of RESTful vs. non-RESTful web services, and also compares these to older but still heavily used distributed object/component middleware. The text guides readers to select an appropriate middleware technology to use for any given task, and to learn new middleware technologies as they appear over time without being greatly overwhelmed by any new concept. The book begins with an introduction to different distributed computing paradigms, and a review of the different kinds of architectures, architectural styles/patterns, and properties that various researchers have used in the past to examine distributed applications

and determine the quality of distributed applications. Then it includes appropriate background material in networking and the web, security, and encoding necessary to understand detailed discussion in this area. The major middleware paradigms are compared, and a comparison methodology is developed. Readers will learn how to select a paradigm and technology for a particular task, after reading this text. Detailed middleware technology review sections allow students or industry practitioners working to expand their knowledge to achieve practical skills based on real projects so as to become well-functional in that technology in industry. Major technologies examined include: RESTful web services (RESTful cloud interfaces such as OpenStack, AWS EC2 interface, CloudStack; AJAX, JAX-RS, ASP.NET MVC and ASP.NET Core), non-RESTful (SOAP and WSDL-based) web services (JAX-WS, Windows Communication Foundation), distributed objects/ components (Enterprise Java Beans, .NET Remoting, CORBA). The book presents two projects that can be used to illustrate the practical use of middleware, and provides implementations of these projects over different technologies. This versatile and class-tested textbook is suitable (depending on chapters selected) for undergraduate or first-year graduate courses on client server architectures, middleware, and cloud computing, web services, and web programming.

In recent years, several knowledge bases have been built to enable large-scale knowledge sharing, but also an entity-centric Web search, mixing both structured data and text querying. These knowledge bases offer

e.g., persons, places, published on the Web as Linked Data. However, due to the different information extraction tools and curation policies employed by knowledge bases, multiple, complementary and sometimes conflicting descriptions of the same real-world entities may be provided. Entity resolution aims to identify different descriptions that refer to the same entity appearing either within or across knowledge bases. The objective of this book is to present the new entity resolution challenges stemming from the openness of the Web of data in describing entities by an unbounded number of knowledge bases, the semantic and structural diversity of the descriptions provided across domains even for the same real-world entities, as well as the autonomy of knowledge bases in terms of adopted processes for creating and curating entity descriptions. The scale, diversity, and graph structuring of entity descriptions in the Web of data essentially challenge how two descriptions can be effectively compared for similarity, but also how resolution algorithms can efficiently avoid examining pairwise all descriptions. The book covers a wide spectrum of entity resolution issues at the Web scale, including basic concepts and data structures, main resolution tasks and workflows, as well as state-of-the-art algorithmic techniques and experimental trade-offs.

RDF and Linked Data have broad applicability across many fields, from aircraft manufacturing to zoology. Requirements for detecting bad data differ across communities, fields, and tasks, but nearly all involve

some form of data validation. This book introduces data validation and describes its practical use in day-to-day data exchange. The Semantic Web offers a bold, new take on how to organize, distribute, index, and share data. Using Web addresses (URIs) as identifiers for data elements enables the construction of distributed databases on a global scale. Like the Web, the Semantic Web is heralded as an information revolution, and also like the Web, it is encumbered by data quality issues. The quality of Semantic Web data is compromised by the lack of resources for data curation, for maintenance, and for developing globally applicable data models. At the enterprise scale, these problems have conventional solutions. Master data management provides an enterprise-wide vocabulary, while constraint languages capture and enforce data structures. Filling a need long recognized by Semantic Web users, shapes languages provide models and vocabularies for expressing such structural constraints. This book describes two technologies for RDF validation: Shape Expressions (ShEx) and Shapes Constraint Language (SHACL), the rationales for their designs, a comparison of the two, and some example applications.

SOA Design Patterns

Challenges in Service Oriented Architecture and Cloud Computing Environments

Hands-On RESTful API Design Patterns and Best Practices

An Introduction to PROV

Easing Access to Semantic Data for Application Developers

Enhanced Building Information Models

After a slow incubation period of nearly 15 years, a large and growing number of organizations now have one or more projects using the Semantic Web stack of technologies. The Web Ontology Language (OWL) is an essential ingredient in this stack, and the need for ontologists is increasing faster than the number and variety of available resources for learning OWL. This is especially true for the primary target audience for this book: modelers who want to build OWL ontologies for practical use in enterprise and government settings. The purpose of this book is to speed up the process of learning and mastering OWL. To that end, the focus is on the 30% of OWL that gets used 90% of the time. Others who may benefit from this book include technically oriented managers, semantic technology developers, undergraduate and post-graduate students, and finally, instructors looking for new ways to explain OWL. The book unfolds in a spiral manner, starting with the core ideas. Each subsequent cycle reinforces and expands on what has been learned in prior cycles and introduces new related ideas. Part 1 is a cook's tour of ontology and OWL, giving an informal overview of what things need to be said to build an ontology, followed by a detailed look at how to say them in OWL. This is illustrated using a healthcare example. Part 1 concludes with an

explanation of some foundational ideas about meaning and semantics to prepare the reader for subsequent chapters. Part 2 goes into depth on properties and classes, which are the core of OWL. There are detailed descriptions of the main constructs that you are likely to need in every day modeling, including what inferences are sanctioned. Each is illustrated with real-world examples. Part 3 explains and illustrates how to put OWL into practice, using examples in healthcare, collateral, and financial transactions. A small ontology is described for each, along with some key inferences. Key limitations of OWL are identified, along with possible workarounds. The final chapter gives a variety of practical tips and guidelines to send the reader on their way.

This book introduces core natural language processing (NLP) technologies to non-experts in an easily accessible way, as a series of building blocks that lead the user to understand key technologies, why they are required, and how to integrate them into Semantic Web applications. Natural language processing and Semantic Web technologies have different, but complementary roles in data management. Combining these two technologies enables structured and unstructured data to merge seamlessly. Semantic Web technologies aim to convert unstructured data to meaningful representations, which benefit enormously from the use of NLP technologies, thereby enabling applications such as

connecting text to Linked Open Data, connecting texts to each other, semantic searching, information visualization, and modeling of user behavior in online networks. The first half of this book describes the basic NLP processing tools: tokenization, part-of-speech tagging, and morphological analysis, in addition to the main tools required for an information extraction system (named entity recognition and relation extraction) which build on these components. The second half of the book explains how Semantic Web and NLP technologies can enhance each other, for example via semantic annotation, ontology linking, and population. These chapters also discuss sentiment analysis, a key component in making sense of textual data, and the difficulties of performing NLP on social media, as well as some proposed solutions. The book finishes by investigating some applications of these tools, focusing on semantic search and visualization, modeling user behavior, and an outlook on the future.

In this truly unique technical book, today's leading software architects present valuable principles on key development issues that go way beyond technology. More than four dozen architects -- including Neal Ford, Michael Nygard, and Bill de hOra -- offer advice for communicating with stakeholders, eliminating complexity, empowering developers, and many more practical lessons they've learned from years of experience. Among the 97 principles

in this book, you'll find useful advice such as: Don't Put Your Resume Ahead of the Requirements (Nitin Borwankar) Chances Are, Your Biggest Problem Isn't Technical (Mark Ramm) Communication Is King; Clarity and Leadership, Its Humble Servants (Mark Richards) Simplicity Before Generality, Use Before Reuse (Kevlin Henney) For the End User, the Interface Is the System (Vinayak Hegde) It's Never Too Early to Think About Performance (Rebecca Parsons) To be successful as a software architect, you need to master both business and technology. This book tells you what top software architects think is important and how they approach a project. If you want to enhance your career, 97 Things Every Software Architect Should Know is essential reading.
REST architecture (style) is a pivot of distributed systems, simplify data integration amongst modern and legacy applications leverages through the RESTful paradigm. This book is fully loaded with many RESTful API patterns, samples, hands-on implementations and also discuss the capabilities of many REST API frameworks for Java, Scala, Python and Go
Web Data APIs for Knowledge Graphs
With examples in Java
Novel Practices and Trends in Grid and Cloud Computing
Transactions on Engineering Technologies
Enabling Technologies, Platforms, and Use Cases

Architecting the Internet of Things

"A comprehensive overview of the challenges teams face when moving to microservices, with industry-tested solutions to these problems." - Tim Moore, Lightbend 44 reusable patterns to develop and deploy reliable production-quality microservices-based applications, with worked examples in Java Key Features 44 design patterns for building and deploying microservices applications Drawing on decades of unique experience from author and microservice architecture pioneer Chris Richardson A pragmatic approach to the benefits and the drawbacks of microservices architecture Solve service decomposition, transaction management, and inter-service communication Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About The Book Microservices Patterns teaches you 44 reusable patterns to reliably develop and deploy production-quality microservices-based applications. This invaluable set of design patterns builds on decades of distributed system experience, adding new patterns for composing services into systems that scale and perform under real-world conditions. More than just a patterns catalog, this practical guide with worked examples offers industry-tested advice to help you design, implement, test, and deploy your microservices-based application. What You Will Learn How (and why!) to use microservices architecture Service decomposition strategies Transaction management and querying patterns Effective testing strategies Deployment patterns This Book Is Written For

Written for enterprise developers familiar with standard enterprise application architecture. Examples are in Java. About The Author Chris Richardson is a Java Champion, a JavaOne rock star, author of Manning's POJOs in Action, and creator of the original CloudFoundry.com. Table of Contents Escaping monolithic hell Decomposition strategies Interprocess communication in a microservice architecture Managing transactions with sagas Designing business logic in a microservice architecture Developing business logic with event sourcing Implementing queries in a microservice architecture External API patterns Testing microservices: part 1 Testing microservices: part 2 Developing production-ready services Deploying microservices Refactoring to microservices

As more and more devices become interconnected through the Internet of Things (IoT), there is an even greater need for this book,which explains the technology, the internetworking, and applications that are making IoT an everyday reality. The book begins with a discussion of IoT "ecosystems" and the technology that enables them, which includes: Wireless Infrastructure and Service Discovery Protocols Integration Technologies and Tools Application and Analytics Enablement Platforms A chapter on next-generation cloud infrastructure explains hosting IoT platforms and applications. A chapter on data analytics throws light on IoT data collection, storage, translation, real-time processing, mining, and analysis, all of which can yield actionable insights from the data

collected by IoT applications. There is also a chapter on edge/fog computing. The second half of the book presents various IoT ecosystem use cases. One chapter discusses smart airports and highlights the role of IoT integration. It explains how mobile devices, mobile technology, wearables, RFID sensors, and beacons work together as the core technologies of a smart airport. Integrating these components into the airport ecosystem is examined in detail, and use cases and real-life examples illustrate this IoT ecosystem in operation. Another in-depth look is on envisioning smart healthcare systems in a connected world. This chapter focuses on the requirements, promising applications, and roles of cloud computing and data analytics. The book also examines smart homes, smart cities, and smart governments. The book concludes with a chapter on IoT security and privacy. This chapter examines the emerging security and privacy requirements of IoT environments. The security issues and an assortment of surmounting techniques and best practices are also discussed in this chapter.

The World Wide Web is now deeply intertwined with our lives, and has become a catalyst for a data deluge, making vast amounts of data available online, at a click of a button. With Web 2.0, users are no longer passive consumers, but active publishers and curators of data. Hence, from science to food manufacturing, from data journalism to personal well-being, from social media to art, there is a strong interest in provenance, a description of what influenced an artifact, a data set, a document, a

blog, or any resource on the Web and beyond. Provenance is a crucial piece of information that can help a consumer make a judgment as to whether something can be trusted. Provenance is no longer seen as a curiosity in art circles, but it is regarded as pragmatically, ethically, and methodologically crucial for our day-to-day data manipulation and curation activities on the Web. Following the recent publication of the PROV standard for provenance on the Web, which the two authors actively help shape in the Provenance Working Group at the World Wide Web Consortium, this Synthesis lecture is a hands-on introduction to PROV aimed at Web and linked data professionals. By means of recipes, illustrations, a website at www.provbook.org, and tools, it guides practitioners through a variety of issues related to provenance: how to generate provenance, publish it on the Web, make it discoverable, and how to utilize it. Equipped with this knowledge, practictioners will be in a position to develop novel applications that can bring openness, trust, and accountability. Table of Contents: Preface / Acknowledgments / Introduction / A Data Journalism Scenario / The PROV Ontology / Provenance Recipes / Validation, Compliance, Quality, Replay / Provenance Management / Conclusion / Bibliography / Authors' Biographies / Index

Microservices Patterns

The Internet of Things