SHEDCAT - A WATERSHED CATALOG

TOM VAN BUREN, DOUG HENDERSON

TWENTY-THIRD ANNUAL ESRI INTERNATIONAL USER CONFERENCE

9 JULY 2003

ABSTRACT

Information at the Watershed Management Division is scattered throughout the enterprise - in basements, closets, and file cabinets; on disk drives; or in our heads. Typically the information is in a proprietary format, unstructured, undocumented, and difficult to use in decision-making, analysis, or presentations. Like a good library, there is a wealth of information; unlike a library, we have neither a catalog nor a mechanism to retrieve it and integrate it into ArcGIS. ShedCat will allow for the contribution, storage, retrieval, and analysis of data and by utilizing Web Services it will serve as a foundation for our enterprise application integration architecture.

1. INTRODUCTION

1.1 ShedCat at a glance

We envision a watershed catalog system, ShedCat, which will allow the contribution, storage, retrieval, and analysis of current, historic, and future data collected and managed within the Seattle Public Utilities Watershed Management Division. ShedCat will serve as the foundation for the division's long term data management architecture and will help the division meet the data reporting requirements of several legal mandates including the Cedar River Habitat Conservation Plan.

1.2 Context

1.2.1 Watershed Mangement Division

Our customers are staff members of the City of Seattle, Seattle Public Utilities, Watershed Management Division responsible for managing two municipal watersheds in King County Washington. The Cedar River Municipal Watershed, the largest of the two, is located 35 miles southeast of Seattle and supplies more than two-thirds of the region's drinking water. Comprising two-thirds of the entire Cedar River Basin, this 91,465 acre natural area supports a wide diversity of plants and animals and several major ecosystems. The Cedar River Habitat Conservation Plan (HCP), a 50-year, ecosystem-based plan ensures the City of Seattle's drinking water supply and protects and restores habitats of 83 species of fish and wildlife that may be affected by water supply and hydroelectric operations on the Cedar River. The plan includes land and forest management in the municipal watershed, mitigation for the blockage to anadromous salmon and trout at the City's drinking water intake, regulation of stream flows in the Cedar River, and research and monitoring to support conservation and mitigation measures.

1.2.2 Seattle Public Utilities Asset Management

Asset Management for Seattle Public Utilities is defined as meeting agreed customer and environmental service levels while minimizing whole life-cycle costs. The culture now in place at Seattle Public Utilities is where we routinely **consider the full life-cycle cost** of a capital or programmatic investment; where our major work activities are driven by established **service levels and comprehensive planning activities;** where we have **"specifiers**" – or work units primarily responsible for directing capital or programmatic expenditures based on established plans and service levels; **and "providers**" – work units whose primary responsibilities are to carry out the work to achieve planning objectives and desired service levels. Chuck Clarke, the Director of Seattle Public Utilities is demanding results - he wants more of our time and resources spent on the hard costs - like building working software - and less time on soft costs, like design and analysis.

1.2.3 Enterprise IT Architecture - Building a more cohesive whole

Customers typically do not think about the enterprise when they need to solve immediate problems. The resultant enterprise IT architecture often resembles multiple stovepipes, a hodgepodge of vertical applications that meet a particular need yet do not necessarily integrate horizontally with each other. One solution, recognized by Seattle Public Utilities in response to asset management, are data warehouses that enable multiple viewpoints and combinations of data not typically analyzed together (Kimball 2002). Such views can be likened to an individualized cockpit or dashboard, similar to that found in an automobile, that displays or *publishes* new and relevant information to the customer and may help make informed decisions.

ShedCat is in concept similar to a data warehouse, creating a more cohesive whole and providing a view to data typically stored in multiple applications across the enterprise. Our enterprise IT architecture is "the collection of strategic and architectural disciplines that encompass the Information, Business System, and Technical Architectures" (Perks 2003, see also Cook 1996)

The most influential paper award, received by Bashar Nuseibeh with two of his colleagues at the 2003 International Conference on Software Engineering, was for a framework they developed ten years ago for organizing multiple views - "development of complex systems invariably involves many stakeholders who have different perspectives on the problem they are addressing, the system being developed, and the process by which it is being developed.. The framework acknowledges the inevitability of multiple inconsistent views, promotes separations of concerns." (Nuseibeh 2003) This framework, and a model Nuseibeh refers to as the Twin Peaks, where requirements and architecture are developed concurrently, emphasizes the difficulties faced in multi-perspective software development - certainly the case for enterprise solutions like a data warehouse and ShedCat.

1.2.4 Agile Software Development

ShedCat is strongly influenced by agile software development methodologies. Practices such as Dynamic Systems Development Methods in Europe, Feature Driven Development in Australia, and Extreme Programming, SCRUM, Crystal, and Adaptive Software Development in USA embrace change, feedback and iterative and incremental development. In 2001 17 practitioners of these practices met and wrote the "Manifesto for Agile Software Development" **(www.agilemanifesto.com).**

The Agile Manifesto
We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Extreme Programming is a discipline of software development with values of simplicity, communication, feedback, and courage. The following XP Bill of rights has guided the development of ShedCat (Jeffries 2001):

Managers and Customers you have the right to:

- ✓ An overall plan, to know what can be accomplished, when, and at what cost
- ✓ Get the most possible value out of every programming week
- ✓ See progress in a running system, proven to work by passing repeatable tests that you specify
- ✓ Change your mind, to substitute functionality, and to change priorities without paying exorbitant costs.
- ✓ You can cancel at any time and be left with a useful working system

Programmers you have the right to:

- ✓ Know what is needed, with clear declarations of priority
- ✓ Produce quality work at all times
- ✓ Ask for and receive help from peers, superiors, and customers
- ✓ Make and update your own estimates
- ✓ Accept your responsibilities instead of having them assigned to you

We've found useful the XP concept of a system metaphor - a story that everyone, customers, programmers, managers, can tell about how the system works. Also influencing our thinking is Alistair Cockburn's metaphor that software development is a cooperative game of invention and communication. The primary goal is to deliver useful, working software. Cockburn sees "managing the incompleteness of communications" as core to mastering agile software development. (Cockburn, 2002, Lakoff 1993, Beck 2000)

Agile development refines the system as it evolves. McMenamin (1992) wrote:

Iterative refinement accommodates two widespread human traits:
Misconception *we get things wrong before we get them right.*
Improvement *we make things badly before we make them well.*

Agile development also accommodates the positive act of discovery - we discover new information, improve the design and grow the software. The real reason we've adopted agile software development methods are to reduce the risk of failure and get to the heart of what the stakeholders want (Cockburn 2002, Cockburn 1993, Larman 2002). For a great overview of agile methods and extreme programming see special feature editions of IEEE Software May/June 2003 and IEEE Computer June 2003.

2. POSITIONING

2.1 Business Opportunity

Currently information on the Cedar River and Tolt River Watersheds at the Watershed Management Division is stored in basements, closets, file cabinets, PC disk drives, or in our heads. Typically the information is in a proprietary format, unstructured, undocumented, and difficult to use in decision-making, analysis, or presentations. Like a good library, there is a wealth of information; unlike a library, we have neither a catalog to understand its content nor a mechanism to retrieve it.

2.2 Problem Statement

Our customers spend too much time and resources locating, retrieving, and disseminating documents and data. The current systems and methods for storing documents and data make it difficult for this information to be integrated across systems, exchanged and accessed. Frequently work is redone because documents and data are lost.

Information is being generated at an astonishingly fast pace. If we don't capture information in a meaningful structure it will be lost forever. Sharing information is vital in learning from others, adapting management, tracking of our work, and incorporating the results in future decisions. Dr Harold Varmus, a Nobel laureate in medicine is quoted in the New York Times "The written record is the lifeblood of science. Our ability to build on the old to discover the new is all based on the way we disseminate the results" (Harmon, 2002). The Knowledge Network of BioComplexity places even more emphasis of access - "The science of ecology urgently requires a concerted effort to develop informatics and collaboratory framework to provide access to information from the many disciplines contributing to our understanding of biocomplexity in the context of global ecosystems" (KNB, 1998).

Information is only valuable to the extent that it is structured. Because of a lack of structure in the creation, distribution, and reception of information, the information often does not arrive where it is needed and, therefore, is useless (Koniger & Janowitz 1995).

2.3 Product Position Statement

ShedCat is an information management service that systematically gathers, analyzes and retrieves resources to help watershed management practitioners at all levels design, implement, monitor, learn, and adjust their essential work processes (Franklin, 1997). It preserves digital information, builds metadata databases and portal access technologies, and allows practitioners to publish their information on the web.

ShedCat effectively meets a real business need; ShedCat analysts find out what workers and managers really do by going to where the work is actually done and carefully observing what actually happens (Goguen, 1997). ShedCat affects more than 10 critical business processes in the Watershed Management Division: watershed protection, cultural resource, public education programs, hydrology, forest ecology, fish and wildlife, watershed planning, field operations, forest road engineering, asset management, and communications. For a nice

definition of business process management see (Smith 2003 and Cook 1996). ShedCat is focused on strategy, targets essential work activities and makes it easier for us to get the job done. ShedCat creates value and aligns enterprise IT architecture with business goals.

ShedCat will provide **context appropriate information**. People need usable, useful information to guide their learning and decisions. Each unit will implement approaches that select the right subset of information from available sources, and publish it in multiple views - the way most effective for a given problem and person. ShedCat will make knowledge so readily accessible that it can't be avoided.

ShedCat is an **open standards-based** information management system. Effective integration of data, information and knowledge requires common syntax and semantics. Adopting and extending open standards and best computing practices will help us build component based architecture to efficiently and cost effectively share resources among disparate data and software applications and to provide seamless and rapid access to information across the enterprise. Open standards include XML, SOAP, and other web services technologies

ShedCat is a **digital library** that identifies resources and collects and represents metadata about those resources. Identifying resources will enable us to discover network accessible things, like electronic documents, images and services; non network accessible things, like employees, trees, and arrowheads; and abstract things that don't physically exist, like "creator". Each work unit will implement approaches that facilitate the use of information acquired in one arena of action, such as field operations, by people in another arena, such as policy decisions.

ShedCat will **integrate and interoperate** with ArcGIS and other installed software systems**.** ShedCat will tightly integrate spatial and non-spatial data.

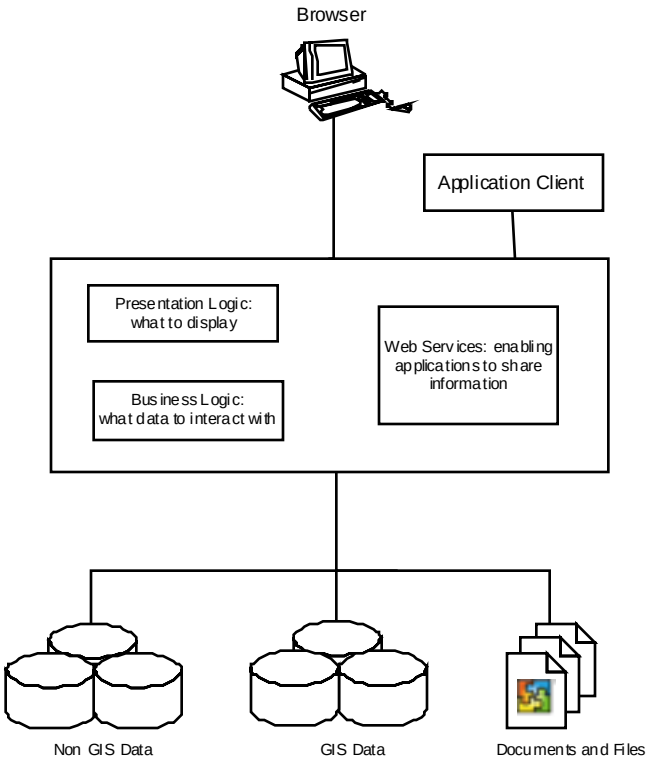
2.4 Benefits

We identified several reasons to build ShedCat:

- Write once use everywhere - reusing software objects and components with other divisions, branches, and departments
- Enabling all staff to contribute and publish content
- One stop shopping - multiple views of the same data for multiple users
- Just-in-time delivery of knowledge - data mining, information discovery, knowledge management so that informed decisions are possible
- Improved Records Management – document the organization, functions, policies, decisions, procedures, and essential transactions of the enterprise
- Improved communication - readily accessible records of action taken, decision points, telling our story, public outreach and education
- Improved project management, time, cost, scope, risk, and procurement management
- Preserving our digital information forever, archived, cataloged, and accessible resources
- Enabling machines to read and process information
- Proof of concept - piloting an n-tier application server, XML, and web services

3. PRODUCT OVERVIEW

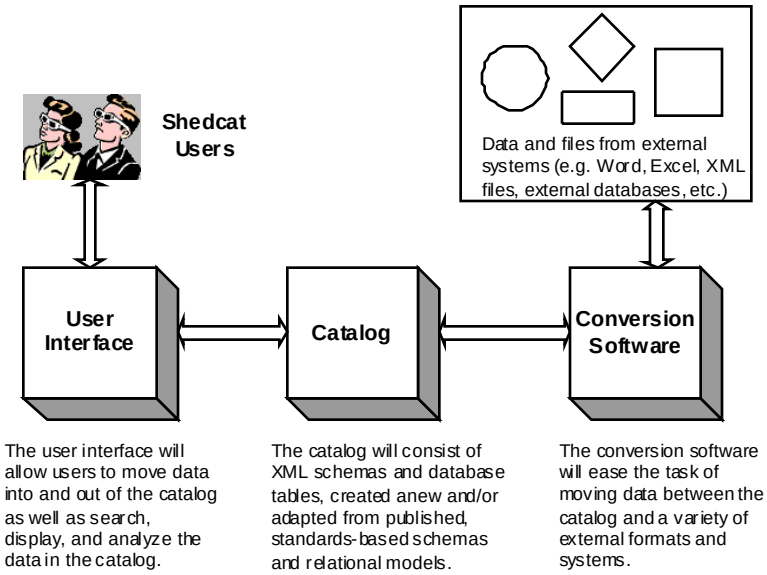
3.1 Architecture



ShedCat is an n-tier service-based architecture - designed to share services and operate within established standards. We apply principles and patterns to create better object designs and follow a set of common activities in analysis and design. ShedCat follows a Model View Control Pattern where the data is separated from how the information is presented and its architecture is divided by functional tiers (Subrahmanyam, 2000):

- User interface - typically a web browser
- Presentation logic - defines what the user interface displays
- Business logic - controls the flow of information
- Infrastructure services - provides support for information flows
- Data tier - stores data - in RDBMS, documents, or directories.

ShedCat consists of three main components: a Catalog, a User interface, and Conversion Software. These components will function as shown in the diagram below:



3.2 Development Methodologies

ShedCat is developed using the *Unified Process*, highlights of which include:

1. Iterative development (frequent releases with heavy customer feedback)
2. Use Case Driven development
3. UML (Unified Modeling Language) as a standard notation for all requirements and design diagrams
4. Object Oriented development

For more information about the Unified Process, see (Larman 2002, Eeles 2003, Krutchen 2000). ShedCat's project web site is structured in large part on (McConnell 1998).

3.3 Risk Management

Risk is the basic problem in software development (Beck, 2000). DeMarco and Lister (2003) advise you to run towards the risk - "A risk is a problem that has yet to occur, and a problem is a risk that has already materialized." We manage risks to ShedCat with XP practices (Beck 2000):

Schedule Slips - Build short release cycles, scope of any slip is limited. Frequent iterations of customer requested features for fine grained feedback of progress.

Project Canceled - Chose the smallest release that makes the most business sense so there is less to go wrong before going into production.

Business misunderstood - Customer integral part of the team. Specification of project is continually refined so learning by the customer and the team can be reflected in the software.

Business changes - Shorten release cycle so there is less change during the development of a single release. The customer is welcome to substitute new functionality for functionality not yet complete.

False feature rich - Insist that only the highest priority tasks are addressed.

Competing requirements - Elicit, organize, and document required functionality and constraints. Track and document trade-offs and decisions. Write effective use cases and prioritize.

Risk Management sets us up for success, helps us bound the uncertainty, maximizes opportunities for personal growth, and focus our attention where it is needed (DeMarco 2003).

3.4 Use Cases

The functionality of ShedCat is divided into *use cases*, which are specific tasks the user can perform with the system. A use case (or *usage case*) may also be considered as a contract between the user and the behavior of the system. Use cases provide an easy way to plan and prioritize the development of the system while improving the chances that the system truly meets the needs. Generally use cases developed for ShedCat focus on elementary business process -as defined in (Larman 2002) - "A task performed by one person in one place at one time, in response to a business event, which adds measurable business value and leaves the data in a consistent state." The definitive resource on writing use cases is (Cockburn 2001).

Our customers declared priority use cases for ShedCat and three subsystems are now in production - Photo Archive, Documents Catalog and a Forest Road Inventory System.

Some use case names for ShedCat follow:

PA-1 Retrieve a photograph from the Photo Archive
PA-2 Enter a photograph into the Photo Archive
PA-3 Update a photograph in the Photo Archive
PA-4 Delete a photograph from the Photo Archive
DC-1 Retrieve a document in the Documents Catalog
DC-2 Enter a document in the Documents Catalog
DC-3 Update a document in the Documents Catalog
DC-4 Delete a document in the Documents Catalog
RI-1 Query Roads Inventory
RI-2 Update Road Inventory
RI-3 Add field(s) to the Road Inventory
RI-4 View all data for a given road system
RI-5 Record observed data for a given road system
RI-6 Incorporate road data into RMAPS reports
RI-7 Locate culvert(s) by station, GPS, or culvert ID
RI-8 Integrate new prescriptions into existing road data
RI-9 Generate a custom roads report
RI-10 Link or overlay roads, streams, and wetlands
RI-11 Incorporate road data into HCP report
RI-12 View road data for Tolt watershed
RI-13 Use road data to prioritize maintenance, decommissioning, and improvements

RI-14 Link road data to ecosystems layers Road Inventory
RI-15 Link road data to noxious vegetation locations
RI-16 Review maintenance records by road # and cost

3.5 Photo Archive

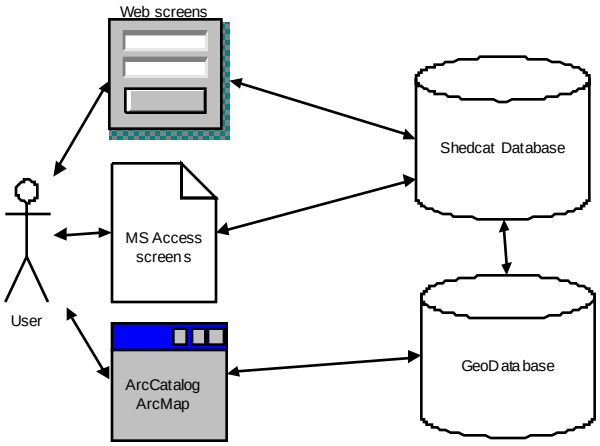
The photo archive allows users to quickly and easily find photographs taken in and around the Cedar River Watershed using key words, locations, and other search criteria. Users can also add photographs to the archive, delete photographs from the archive, and/or update search data for photos in the archive.

3.6 Documents Catalog

Similar to the Photo Archive, the Documents Catalog allows users to quickly and easily find documents about the Cedar River Watershed. The user can publish documents to the catalog, delete documents, or update search data. In addition the Documents Catalog has a navigation tree to help users explore documents in a way they are used to doing in Microsoft Windows.

3.7 Road Inventory Management System

The Road Inventory Management system is being built to meet the needs of the Watershed Management Division's Operations Section in storing, retrieving, and analyzing information related to the roads, bridges, and culverts within the Cedar and Tolt Watersheds. The Road Inventory is the first subsystem of ShedCat that integrates with ArcGIS. The architecture which has emerged for the Road Inventory is shown below:



All the data ultimately resides in the Geodatabase (e.g. roads, culverts, bridges) as map features or attribute tables related to these features. This data is accessible through ArcCatalog and ArcMap, including custom input screens which link feature classes with attribute tables and prompt the user for all such data in one edit session. There are links from the ShedCat database to the data in the Geodatabase, along with MS Access and web screens for editing this data. The idea here is that if a graphical representation is not important for a given editing session, the user will not have to start up ArcCatalog or ArcMap to edit road or culvert data, or generate reports. For example, if a 12" pipe is replaced with an 18" pipe for

a given culvert, the user should just be able to bring up a simple screen where she can make this simple edit to one piece of data, without having to go through one of the Arc tools. The links from the ShedCat database to the Geodatabase, along with the MS Access and web input screens, make this possible.

The architecture shown above is the foundation for the Road Inventory system. The remaining use cases will be implemented by just adding details to this architecture (e.g. more tables, feature classes, input screens, etc.). The hardest part of the project, i.e., working out this architecture, is mostly behind us. The remaining work should go faster with more use cases being implemented in shorter time periods.

4. EVALUATION

The Watershed Management Division has put in production working software that responds to our customer's declared priorities. While information is still scattered throughout the organization we now have in place a foundation to build additional modules, to integrate with ArcGIS and to make it easy for our customers to publish, retrieve, and analyze information. The basic architecture is in place and has met with the customer's general approval.

Some of the details behind opening up the data in the Geodatabase for editing by multiple people using both the Arc tools and non-Arc tools are currently being worked out. There are technical issues involving database roles and privileges, the versioning nature of the Geodatabase, and creation of new data (as opposed to merely editing existing data) from outside the Arc tools, which are being solved.

There are many use cases remaining but these details will fit into the existing framework and will not require extensive redesign of the basic architecture. Essentially, these remaining use cases will be met by adding more data and tables to the existing architecture.

Publishing content to the catalog is a new process for our customers and requires effort not traditionally expended. As more content is published, and providing information about the content becomes easier and routine, we are convinced usage will pick up.

Since ShedCat was released to production on September 19, 2002 there have been over 70,000 individual hits to the system - averaging 410 hits per business day.

5. REFERENCES

Beck, K., 2000, eXtreme Programming explained: Embrace Change, Boston: Addison Wesley

Beck, K., Fowler, M., 2000, Planning Extreme Programming, Boston: Addison Wesley

Cockburn, A., 1993, "The Imapact of Object Orientation on Application Development," *IBM Systems Journal*, Vol 38, Nos 2&3.

Cockburn, A., 2001, Writing Effective Use Cases, Boston: Addison Wesley

Cockburn, A., 2002, Agile Software Development, Boston: Addison Wesley

Cook, M., 1996, Building Enterprise Information Architectures, Upper Saddle River, NJ: Prentice Hall

DeMarco, T., Lister, T., 2003, Waltzing with Bears: Managing Risk on Software Projects, New York, NY: Dorset House Publishing

Eeles, P., et al., 2003, Building J2EE Applications with the Rational Unified Process, Boston: Addison Wesley

Franklin, J, 1997, "Ecosystem Management: An Overview", in Boyce, M, Haney A eds, Ecosystem Management, New Haven: Yale University Press

Goguen, J., 1997, "Towards a Social, Ethical Theory of Information." In Social Science, Technical Systems, and Cooperative Work: Beyond the Great Divide, edited by G. C. Bowker, W. Turner, S. L. Star and L. Gasser. Mahwah, NJ.: Lawrence Erlbaum Associates. pp.27-56.

Harmon, A. 2002, "*New Premise in Science: Get the Word out Quickly, Online*", in New York Times, December 17, 2002

Jeffries, R., Anderson, A., Hendrickson, C., 2001, Extreme Programming Installed, Boston: Addison Wesley

Kimball R, Ross M., 2002, The Data Warehouse ToolKit, New York, NY: Wiley

KNB, 1998, Knowledge Network for Biocomplexity: http://knb.ecoinformatics.org/

Koniger, P., and Janowitz, K., 1995 "Drowning in Information, But Thirsty for Knowledge." *International Journal of Information Management* 15, no. 1 (February 1995): 5-16.

Kruchten, P., 2000, The Rational Unified Process An Introduction, Boston: Addison-Wesley

Lakoff, G., Johnson, M., 1983, Metaphors We Live By, Chicago, IL: University of Chicago Press

Larman, C., 2002, Applying UML and Patterns, Upper Saddle River, NJ: Prentice Hall

McConnell, S., 1998, Software Project Survival Guide, Redmond, WA: Microsoft Press

McMenamin, S., 1992, "The New Economics of Requirements Modeling and Prototyping,: The First National Conference on Software Methods, Orlando FL, presented by the Technology Transfer Institute, 741 Tenth Street, Santa Monica, CA 90402 (March 30, 1992)

Nuseibeh, B, 2001 Weaving the Software Development Process Between Requirements and Architecture Proceedings of ICSE-2001 International Workshop: From Software Requirements to Architectures (STRAW-01) Toronto, Canada

Perks, C, Beveridge, T., 2003, Guide to Enterprise IT Architecture, New York, NY: Springer-Verlag

Rowley, J., and Farrow, J., 2000, 3rd ed., Organizing Knowledge, Cambridge, UK: Gower

Smith, H. and Fingar P., 2003, Business Process Management: The Third Wave, Tampa, FL: Meghan-Kiffer Press

Subrahmanyam, A., et al, 2000, Professional Java Server Programming, Birmingham, UK: Wrox Press

Author Information

Tom Van Buren

Watershed Management Division, Resource Management Branch
Seattle Public Utilities, City of Seattle
19901 Cedar Falls Road
North Bend, WA 98045
Phone (206) 386 4212
Fax (206) 615 1923
Email vanburt@seattle.gov

Doug Henderson

Watershed Management Division, Resource Management Branch
Seattle Public Utilities, City of Seattle
19901 Cedar Falls Road
North Bend, WA 98045
Phone (206) 615 0203
Fax (206) 615 1923
Email henderdl@seattle.gov