Abstract Submission Form

Presentation Title	Branched Libraries and Rapid Relase Single-Source Enviroment Implementation
Suggested Track	Stragegic: to see why Arbortext & Interwoven were a requirement for growth Application: to see why we chose Interwoven and how we made it work
Company	Juniper Networks
Name	Liz Rodgers, Tools Developer Brenda DePaolis, Production Expert
Address	1194 N Mathilda Ave, Sunnyvale, CA 94089
Phone	408-745-8119
Fax	408-936-1801
Email	liz@juniper.net brenda@juniper.net
Abstract	This presentation will highlight the way Arbortext's Epic Editor has been implemented within the technical publications organization at Juniper Networks, focusing on the successful rollout of the editor and support systems. This document also details some of the proposed enhancements for continued Arbortext integration. Finally, it describes some of the challenges in deploying the Arbortext suite of tools and how those challenges were met.
Speaker Bio(s)	Liz Rodgers is the internal tools person for the Technical Publications group at Juniper Networks and project architect. For the last two years, it has been her to plan, architect and implement a new single-sourcing system that will replace traditional publishing applications and processes currently in use in the department. Brenda DePaolis is the publishing and production expert in the Technical Publications group at Juniper Networks. She has been fully responsible for the FOSI implementation and design. She has also become one of the DTD experts (from a functional point of view) to assure output requirements are met. She's currently investigating the move to XSL-FO. Brenda will demonstrate our implementation and workflow while Liz presents the implementation details.

Contact: liz.fraley@single-sourcing.com

Arbortext Implementation Presentation Proposal

Date 18 February 2001

Prepared By
Juniper Networks
Liz Rodgers, presenter
Brenda DePaolis, demonstrator

Purpose of Document This document details the proposed presentation for Juniper Networks at AUGI 2003. It highlights how Arbortext's Epic Editor has been implemented within the technical publications organization at Juniper Networks, focusing on the successful rollout of the editor and support systems. This document also details some of the proposed enhancements for continued Arbortext integration. Finally, it describes some of the challenges in deploying the Arbortext suite of tools and how those challenges were met.

Contents

Project Overview 3
- Conversion Requirements 2
- Authoring Requirements 3
- Production Requirements 3

Project Implementation 4
- Epic Editor 5
- Production 3
- Source Control 3

Successes 5

Project Overview

The Arbortext project at Juniper Networks is a major technical publications initiative designed primarily to convert existing documentation and author new documentation in a single-sourcing environment. By authoring in an XML environment, we are able to make use of vast areas of overlapping information. Through information reuse, we can effectively increase the efficiency of our entire technical publications staff, an ever-increasing goal as demands for documentation increase while staffing and resource challenges persist.

The single-sourcing project had several initial requirements that had to met. These ranged from the successful conversion of Framemaker files to Epic files to the integration of Interwoven to provide a viable source control environment.

Conversion Requirements

Juniper Networks technical publications department handles 71 books with between 100 and 200 pages. Of these books, 13 exist in 4 different physical incarnations to match the 4 active releases (for a total of 52 active documents) and are owned by a total of 10 writers (approximately 5 books per writer). We release software 4 times a year and make updates to active releases for as long as they are available to customers.

Because of the high volume of material, the need to profile material for multiple releases, and the rapid release requirements, the conversion of these books was a primary requirement for the single-sourcing project to succeed.

The staff involved in converting these books includes writers of varying levels of experience. The conversion process had to be such that those writers with less experience could successfully handle the conversion and post-conversion clean-up, while those writers with more experience could easily transition from a known environment to a new authoring environment.

Authoring Requirements

The implementation of the Arbortext authoring environment had several key points that had to be addressed. The new authoring environment had to be very user-friendly and easy to learn to facilitate the migration to a new authoring tool. It had to support the existing style structure that had been implemented in the original FrameMaker authoring environment. Additionally, the look and feel of the published documents had to be comparable to the existing published documentation. Finally, the authoring environment had to be able to scale easily so that we could make enhancements and changes without major infrastructure changes going forward.

Production Requirements

The production requirements were perhaps the most critical to the single-sourcing project. We had to be able to easily publish documentation in PDF, HTML, eBook, and Palm formats. Moreover, this process had to be streamlined such that it could be handled for a growing documentation suite with a reduced production staff.

The project required the integration of a source control tool that could handle branching. The branching requirement exists because our documentation spans many books that are released and maintained for multiple versions of the software they document. Because we maintain multiple concurrent versions of the documentation, we decided that each version had to be maintained within its own branch.

Project Implementation

The implementation of the entire project can be divided into several key areas: Epic Editor, production work, and source control.

Epic Editor

The first task in migrating to Epic Editor was the creation of the DTD and the accompanying FOSI. The DTD and initial FOSI work required analysis of the existing style and formats used with the FrameMaker files. This initial step took four months of work by a part-time contractor. In retrospect, an individual skilled in this area and working full-time on the project very likely could have implemented the DTD and FOSI within a single month. The most difficult task in this initial step was the analysis of book structure to decide which tags would be required for the DTD. The actual creation of the DTD was not a barrier to progress. The FOSI has been more difficult.

The migration to Epic editor focused on the conversion and clean-up of existing documentation. The primary conversion was handled with a simple Interchange mapping file. The development and refinement of this template was fairly easy and straightforward. After refining the conversion template, the most recent conversion efforts required four hours to convert and clean up a single book. We expect this to become more difficult with some of our other books, but the conversion process is simple and straight-forward. Most of the limitations we've experienced here are related to the limitations in the original FrameMaker documents.

Production

The production work to make Epic Editor work within our production requirements primarily included work with the E3 engine to generate PDF from within an Interwoven environment and XSLT and DOM tree work to handle indexing and HTML pagination. The details of this implementation are beyond the scope of this presentation.

Source Control

Our source control environment had to handle multiple versions of multiple books. Even though the documented information is approximately 90% the same from release to release in the FrameMaker environment writers were required to maintain and replicate documentation bug fixes between 4 or 5 sets of binary files.

We investigated creating a library to span books and versions but soon discovered that even the libraries require the branching structure. Because we were most concerned about maintaining the branching integrity, we chose a content management system that specialized in branching capabilities. As such, we chose to integrate Interwoven rather than an Oracle database. Implementing a database would have required additional branching code, which we did not have the resources to implement and complete.

To allow for information reuse, we designed our repository by version. Each version is its own branch, and each branch has its own libraries. The libraries contain the shared information and appear at several locations in the repository tree as dictated by usage requirements. This structure allows us to use file entities to greatly decrease the amount of duplicate information that exists within our documentation suite.

An additional advantage of the Interwoven system is that it provides a source control environment for our support tools. This was simply an added benefit of the integration so that we did not have to use another tool (CVS) to control our tool environment.

Successes

Within a year of beginning full-time development on the single-sourcing project, we were able to publish our first set of books fully within the Arbortext environment. This included the creation of the DTD, identifying information for reuse across those guides and creating the file entities, the creation of the print FOSI and integration of the E3 engine with our Interwoven environment, and the development of some support tools (XSLT, DOM tree, clean-up scripts, and so on). More importantly, we were able to author these guides with literally one-quarter of the work it required prior to the migration to a single-sourcing environment. The re-use of information across multiple books and across multiple versions of the same books was key to being able to deliver these books on time amid staffing changes. Being able to edit only a single chunk of information rather than the same chunk multiple times was key to this success.

Since that time, we also were able to successfully convert a number of other books and push those through production. By the time of the conference, we will have published our first series of 6 books books completely in with Epic Editor.

Our most interesting book is one that documents syslog messages that are generated by the JUNOS software. Prior to the migration to this authoring environment, this book required almost a month of work for each release. Using the single-sourcing environment, this book now takes less than 30 minutes to author the same information and push the entire book through production. We are able to pull the syslog messages directly from the software and import them into our XML environment quickly and effortlessly. The FOSI and XSLT transform the software-produced XML into the correct published format.

With each release (close to 40 a year across all our products), we have to author and publish a set of release notes. Under the new environment, this now takes minutes instead of days. We are able to pull information directly from our bug-tracking system much like with the syslog book. This information is quickly imported into the XML environment, and the release notes are production-ready in minutes. More importantly, because we can do this in minutes, we can now catch last-minute changes that are not always communicated.

Despite these successes, perhaps the most important success we have had is found in the time savings of information reuse. In these tough economic times, staffing and resources are at the forefront of every manager’s mind. The products must still be developed and delivered, but this must be done with fewer people. Quality is sacrificed to meet the volume of work that must be done. Using the Arbortext solution to this problem, we have been able to continue to meet our deliverables with 60% of our previous staffing. More importantly, not only are we meeting deliverables but we are improving the quality of those deliverables.