AUGI 2003

Branched Libraries and Rapid Release Single-Source Environment Implementation

Liz Rodgers
& Brenda DePaolis
Juniper Networks

Juniper™ NETWORKS

Overview

- **Arbortext Epic Editor acts as user front-end**
  - **All content is created and edited at front-end**
  - **Entities (both text and file) are created at front-end**
- **Interwoven Server (IWS) acts as repository for all content and support tools**
- **TeamSite client and Epic interact with IWS**
  - **Epic opens files directly on IWS (authoring)**
  - **TeamSite client invokes Arbortext E3 publishing engine (production)**

Arbortext E3 Publishing Engine

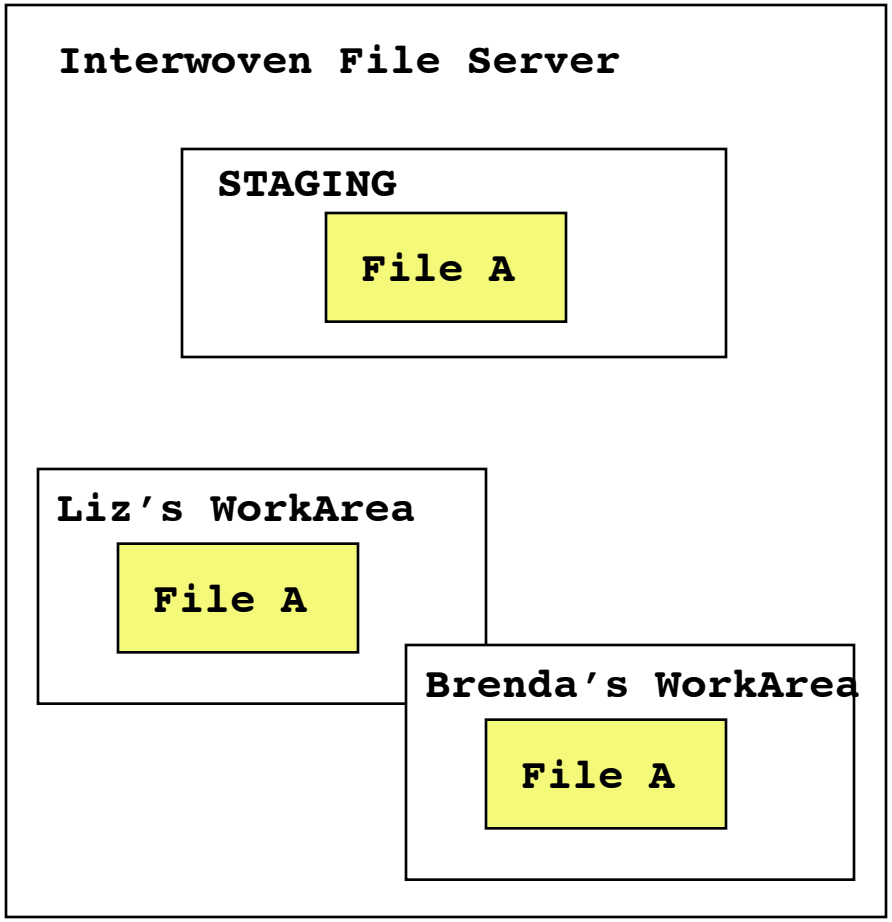
- **E3 engine is invoked by IWS client (Team Site)**
- **To invoke the E3 engine to create PDF or PS:**
  - **From the staging area, use custom preview function (View -> Techpubs Preview)**
  - **Select FOSI and output format**
  - **Team Site invokes a servlet that calls the E3 engine**
  - **Can create PS or PDF on any file (including file entities)**

DOM Tree

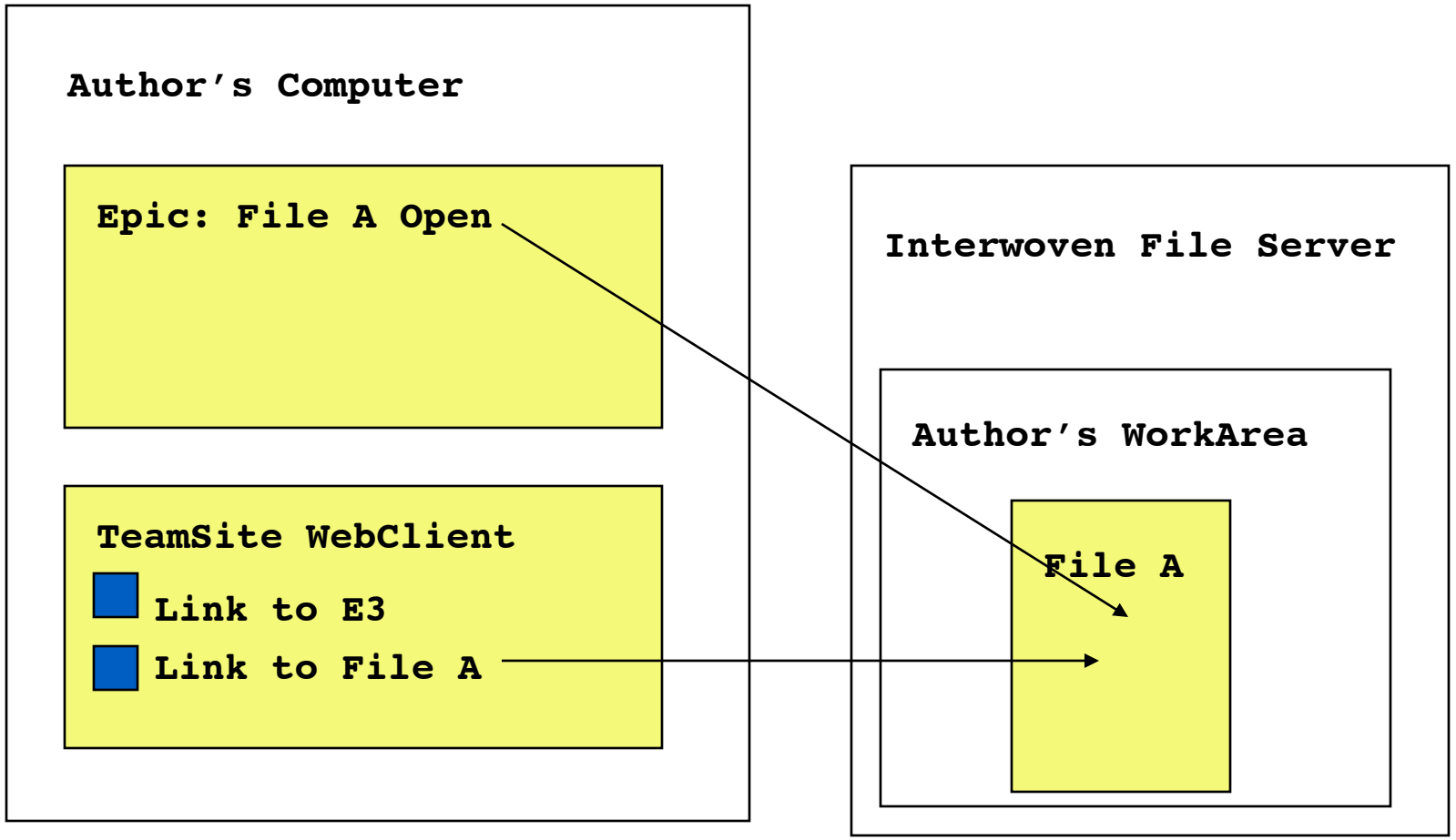
- **DOM tree uses Java to parse the XML structure of the document and save each element as an object in memory**
- **Once the document is stored as an object tree, those objects can be chunked into individual pages based on tag names and element properties**
- **DOM tree executes XSLT to convert pages to HTML**
- **Not yet integrated with E3**

Overview: Understanding IWS

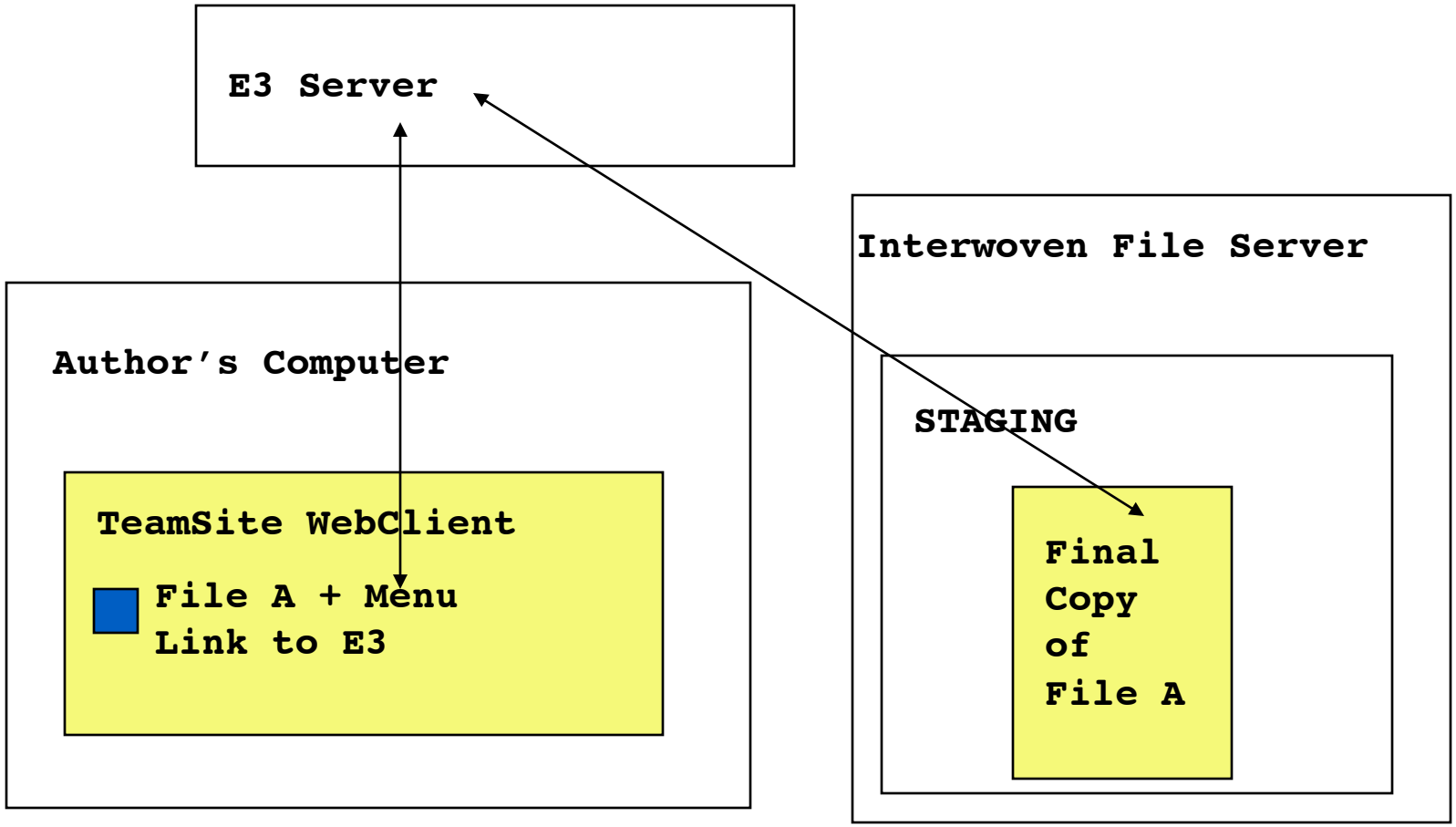
- **STAGING area is repository for all documents**
- **Authors have copies of files they're working on in their workareas**
- **Authors submit changed files to staging for sharing between workareas**
- **IWS manages merging and version control**



Overview:
Authoring in Epic on IWS



Overview: Integrating with E3



Preparing Content for Re-Use

- **Sharing Content across Documents**
  - **Content Re-Use Model**
  - **Content Libraries**
  - **Shared Content Cases**
  - **Shared Identifiers**
  - **Using Scope to Create Conditional Text through Text and File Entities**
- **Sharing Content across Branches**
  - **Branches identify versions points in time**

Content Re-Use Model

- **Information is multi-dimensional**
  - **Information is divided across books at one time: 15 software books, 13 hardware books, and several miscellaneous books**
  - **Information is divided in time: several books exist in multiple versions at the same time**
- **Re-use happens at the chapter, section, and even paragraph level**
- **Text entities are used in conjunction with file entities to create conditional text**
- **Reusable content is stored in branch-specific libraries (including images)**

Content Libraries

- **All reusable content exists in the libraries**
- **Only reusable content exists in the libraries**
- **Image Library is the exception**
  - **All images live in the same library**
  - **Parameter File Entity used to include declarations for all images**

Libraries exist at multiple levels

- **Libraries correspond to level at which content is shared**

Corporate Level	Shared across all books, corporate-wide
Document Type Level	Shared across all Hardware books or all Software books
Book Level	Shared across files in a single book

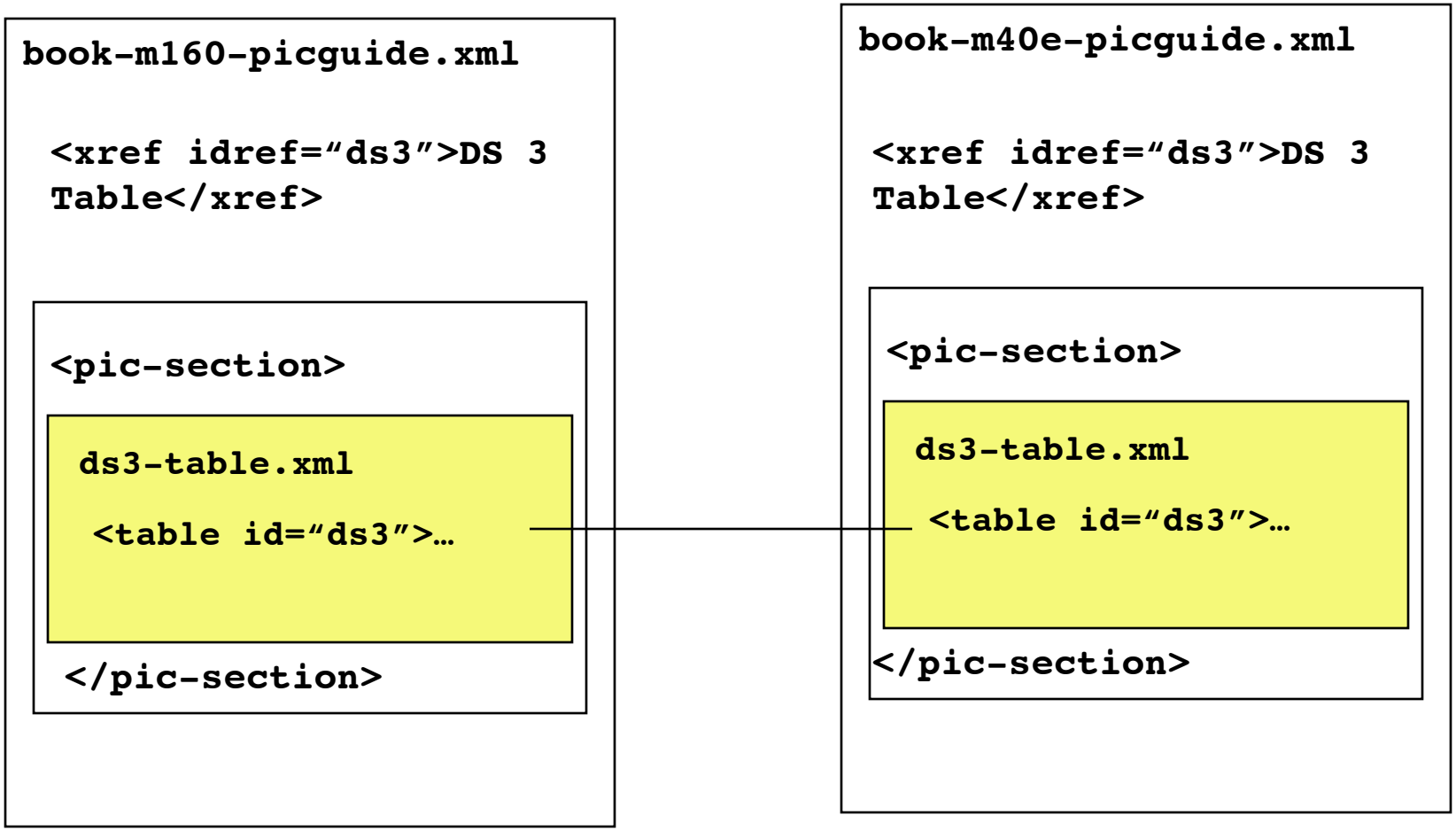
Shared Content Cases

- **Create entities for shared content**
  - **Create text entities for trademarked names and small pieces of boilerplate information that are always shared**
  - **Make entities out of shared content across multiple files if most of the content is not shared**
- **Create entities for unique content**
  - **Make entities out of unshared content in a single file if most of the content in that file is shared**

Shared Identifiers

- **Data Analysis is essential**
- **Identifiers must be unique within a book, not across books**
- **Identifiers in shared material are the same across books**
  - **Need to preserve cross references into shared material**

Example: PIC Books



Using Text and File Entities to Create Conditional Text

AUGI 2003

- **Large chunks of information are reused in multiple books with only a single differing (or omitted) paragraph**
- **Create a file entity for the content**
- **Create individual file entities for the conditional text**
- **Reference the conditional material through a non-specific text entity**
- **Define the text entity as the corresponding file entity for the book-specific scope**

Scope

- **File Entities have multiple scope**
  - **As a stand alone file**
  - **As an included file entity inside another element**
- **Benefits of defining text entities with generic values inside the stand alone file entity**
  - **Values defined at the book level will override values defined inside the file entity**
  - **Users can open up individual files and keep context checking turned on**

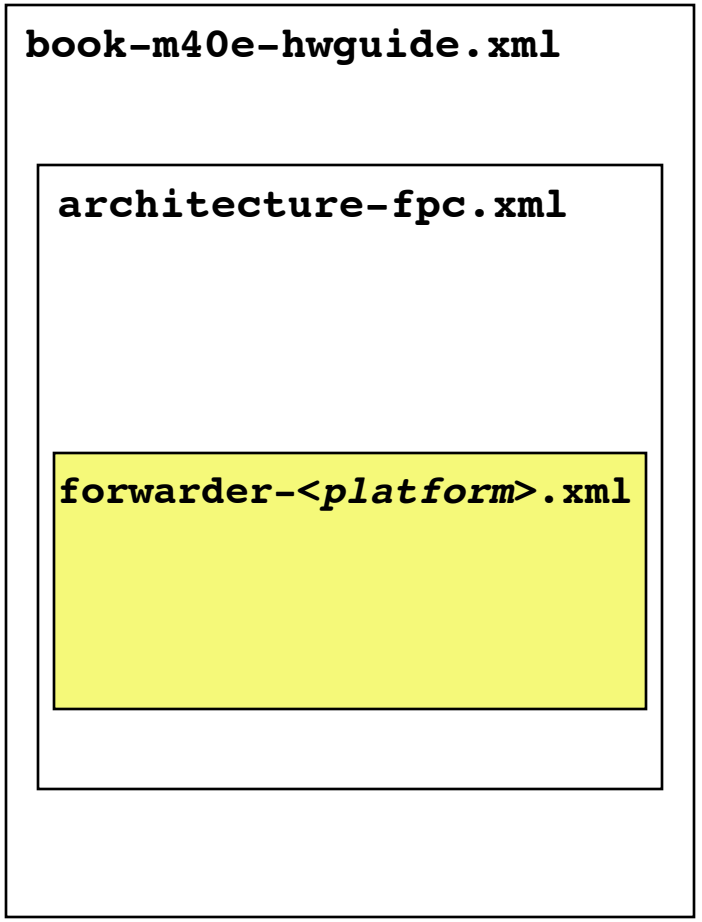
Conditional Text Example

- **Save forwarder text as platform-specific file entities**
- **In architecture-fpc.xml, reference the text with generic text entity**

  &forwarder;

- **In book file, declare the text entity as the platform-specific file entity**

  <!ENTITY forwarder "&forwarder-m40e.xml">



Branching

- Does Content Branch?
- Why Branch and not Copy?
- How are Branches Created?
- Example: Typical Branched Software Structure

Does Content Branch?

- **Software books exist in 5 different versions at the same time**
- **Versions differ by 5%; “shared” material is 95%**
- **Documents branch rather than have a 3rd dimension library (time)**
- **Bug fixes need to be made across branches to the 95% of shared material**
- **Resource restriction lead to choosing branching over database implementation**

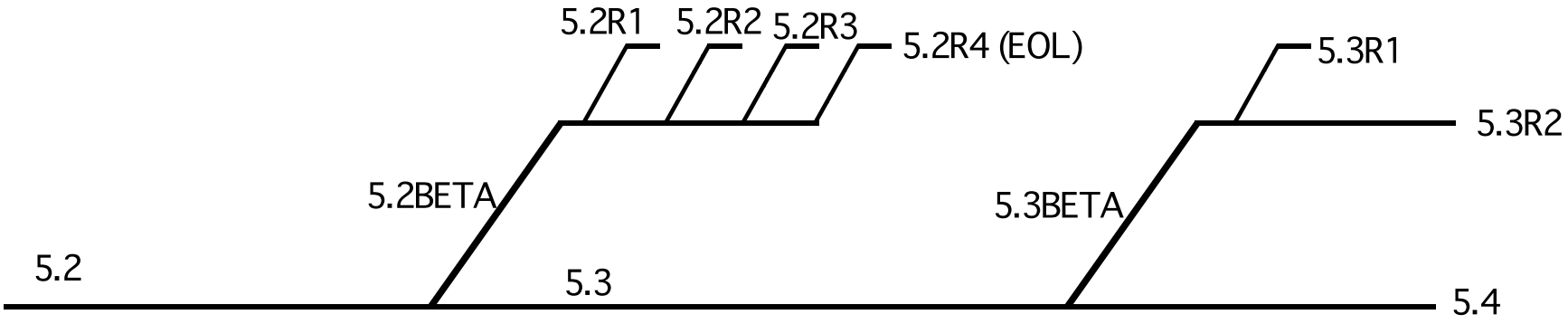
Why Branch and not Copy?

- **Books branch with software releases**
- **Turned out libraries also branched**
- **Information was modified from release to release, and although information was 95% the same, the changes were in feature operation most of the time and not new information**
- **Branches should be made when files contain features with incompatible policy**
  - **When one development group doesn't wish to see another development group's changes, that is also a form of incompatible policy**

How are Branches Created?

- **Repository is set up by version branch**
- **Main trunk is the current version being developed**
- **Branches are created when new versions are released**
- **Editions published in archive branch to freeze versions and create snapshots of final versions**

Example: Branching Software Tree



Implementation Success

- **PIC books delivered to print, web and CD 3/30/2003**
- **Hardware Books delivering 06/15/2003**
- **Syslog Software book, ready but waiting for rest of Software books**
- **Release Notes, pending engineering-side implementation, expected to deliver 07/05/2003**
- **Other Software books in pre-conversion stage, expect completion by 4Q 2003**
- **Cable books in transitional stage, 1Q 2004**
- **Network Operator Guide (NOG) in planning stages 2Q 2004**

Implementation Success

- **PIC books all updated together**
  - **Most information is shared**
  - **One PIC book per platform**
  - **Any (or all) PIC books may need updating each release**
- **Formerly, these books released only four times a year and took a quarter to produce**
- **With XML approach, takes less than 30 minutes to create a new book and only a few days to update them all with any new PICs**

Implementation Success

- **Create release notes directly in XML**
  - **Scripts search the bug tracking system for release note fields**
  - **Information dumped directly into XML**
  - **Editing in Epic takes less than 1 hour**

Implementation Success

- **Create release notes directly in XML**
  - **Scripts search the bug tracking system for release note fields**
  - **Information dumped directly into XML**
  - **Editing in Epic takes less than 1 hour**

Future

- **Software Books**
- **Cable & NOG Books**
  - **Cable books are exactly the same as the Software books but include cable-specific commands and information**
  - **NOG books contain checklists and selected duplicate information found in Software books**
- **.NET XML Diff/Patch to populate bug-fixes across branches**
- **E3 integration with Java/DOM program**

AUGI 2003

Juniper NETWORKS

Liz Rodgers, liz@juniper.net
Brenda DePaolis, brenda@juniper.net

Juniper your Net™