

A Bit of XML

(To Solve a Boatload of Problems!)

Russell Ward
NEO-STC 2007 Regional Conference

Speaker information

- # Russell Ward of Carlisle, PA, USA
- # Full-time tech writer for Spirent Communications of Rockville, MD.
www.spirent.com
- # Owner of West Street Consulting, a part-time business for Adobe FrameMaker plugins and general consulting
(www.weststreetconsulting.com)
- # Adobe Certified Expert in FrameMaker
- # Persistent and sometimes annoying evangelist for structured content technologies
- # Contact info:
 - # russ.ward@spirent.com
 - # russ@weststreetconsulting.com
 - # Work phone: 301.548.6922

Problem summary

- ✦ Spirent Communications has a rich history of the following techcomm issues:
 - ✦ **Redundancy** – The same things written (or copy/pasted) over and over again
 - ✦ **Inaccessibility** – Information not readily available to those who need it.
- ✦ Spirent does not have the resources, the will, or quite frankly, the volume of content to bring in a content management system. Therefore, any solution would have to be borne of “Inspired Innovation,” not the corporate checkbook.

Solution summary

- # To address redundancy, the following single-sourcing practices were implemented, all using structured content technologies:
 - # **Conditional text** – Reusing the same content in document order by a process of *filtering*
 - # **Text insets** – Pulling in modules of content at will from a central library of data
 - # **Multichannel publishing** – Rendering the same content in different formats for different audiences, such as HTML vs. PDF, through automated processes
- # Multichannel publishing was also key to addressing inaccessibility.
- # All tools and technologies employed in the solution are effectively FREE.

Challenges

- # No money for outside help
- # No precedent to build upon or learn lessons from
- # A very deep hole to dig out of from decades of poor and inefficient practices
- # A technical publications department that was stoically and pervasively resistant to change even, in the face of imminent process collapse
- # Ongoing corporate downsizing that reduced available manpower
- # An uphill battle to gain support and respect from the rest of the company for new ventures, due primarily to historically substandard TechPubs performance.

The problems - A closer look

- ✦ Spirent produces hardware products which require an interface command manual (How to talk to the device)
- ✦ Many commands are similar or identical between different products
- ✦ External customers obviously need this information
- ✦ Internal personnel need it too for development purposes and they need it *now*, not at product release
- ✦ TechPubs used FrameMaker for these manuals because of classic Word limitations

ACT-USER *Activate User*

[Description](#) | [Input format](#) | [Normal response format](#) | [Error response format](#)

Product: 3577C
Command type: General System
Privilege level: Login

Description

Logs a user into the test system. The specified username and password afterwards.

▲ TOP

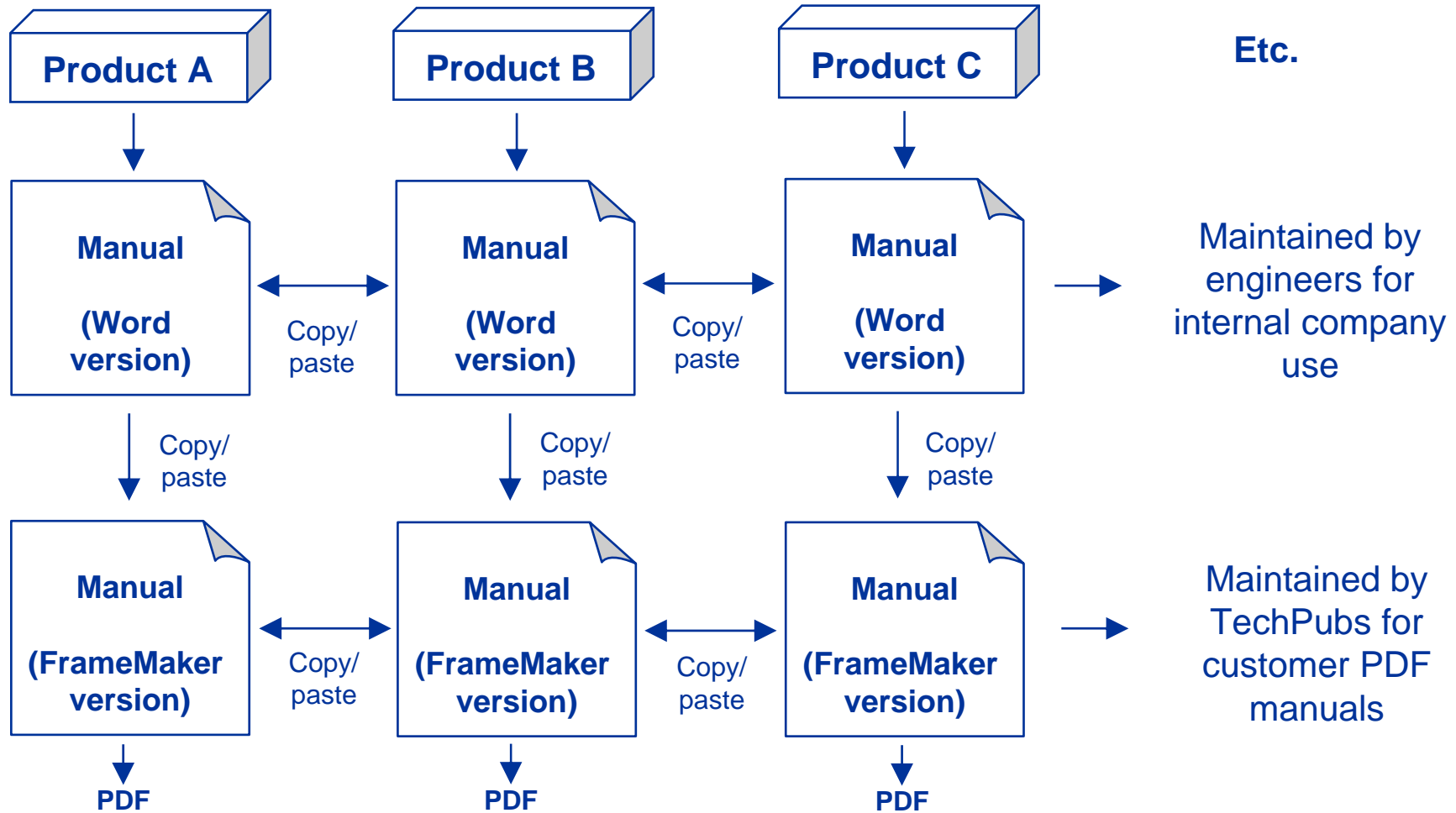
Input format

ACT-USER: [<sid>] :<uid>:<otag>::<pid>;

| Parm | Description |
|-------|---|
| <sid> | Site identifier of the test set to receive the command. |

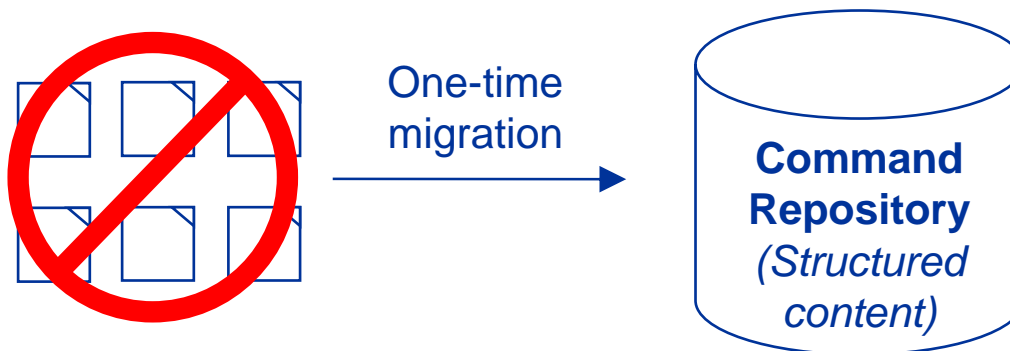
So, here's how these manuals took shape, keeping in mind everything presented so far...

The problems - A closer look (cont'd)



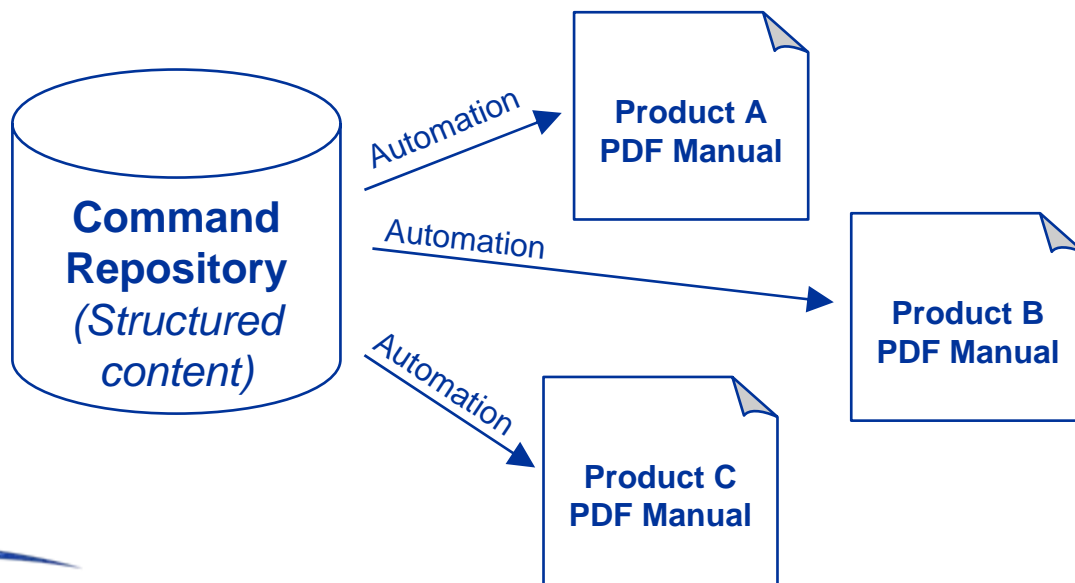
Solution #1 - Conditional processing

- # A central XML repository was created for all commands of all products, much like a giant, composite command manual.
- # Using structural metadata, information was tagged according to the product to which it applied.

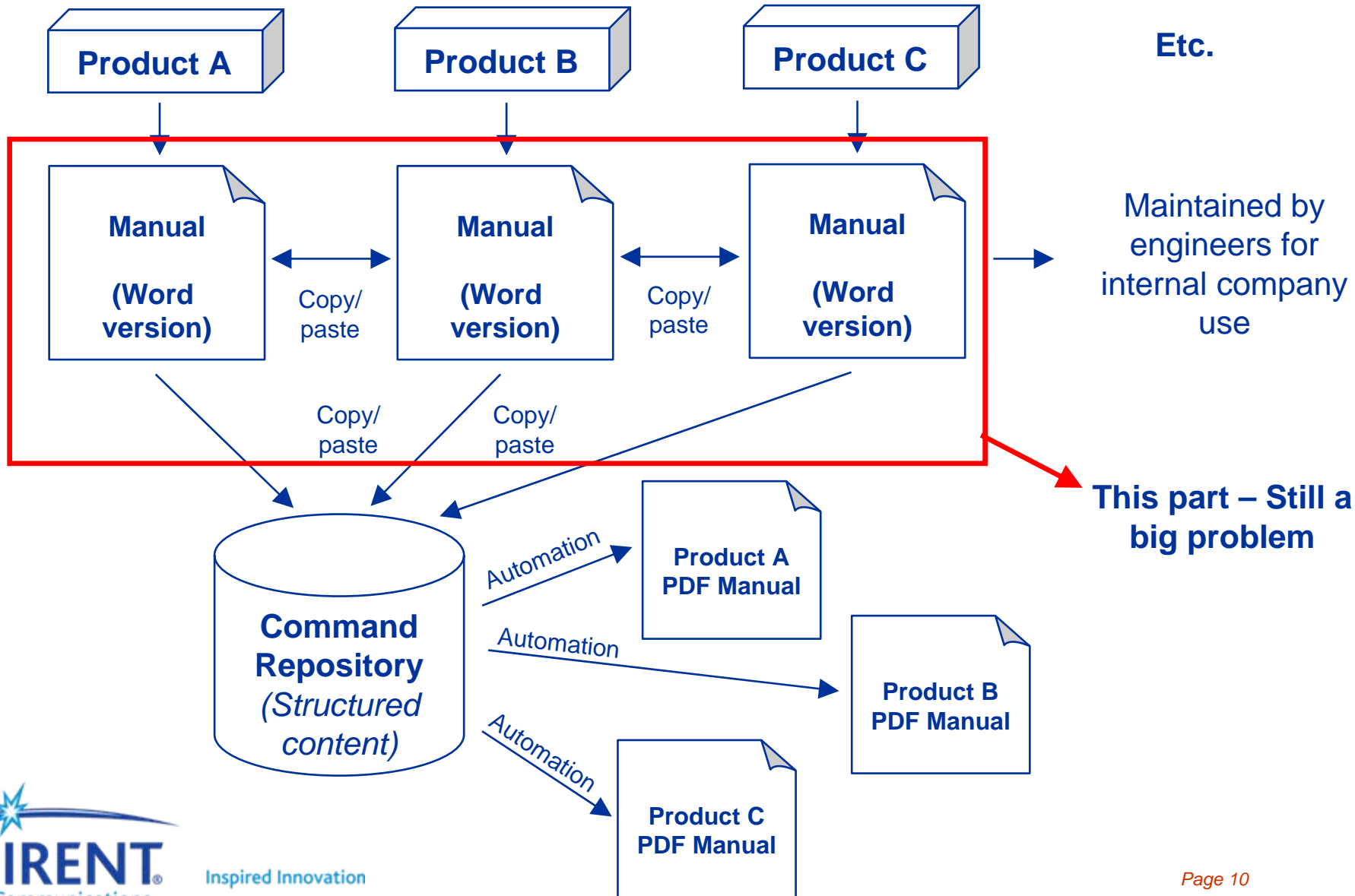


Conditional processing - Cont'd

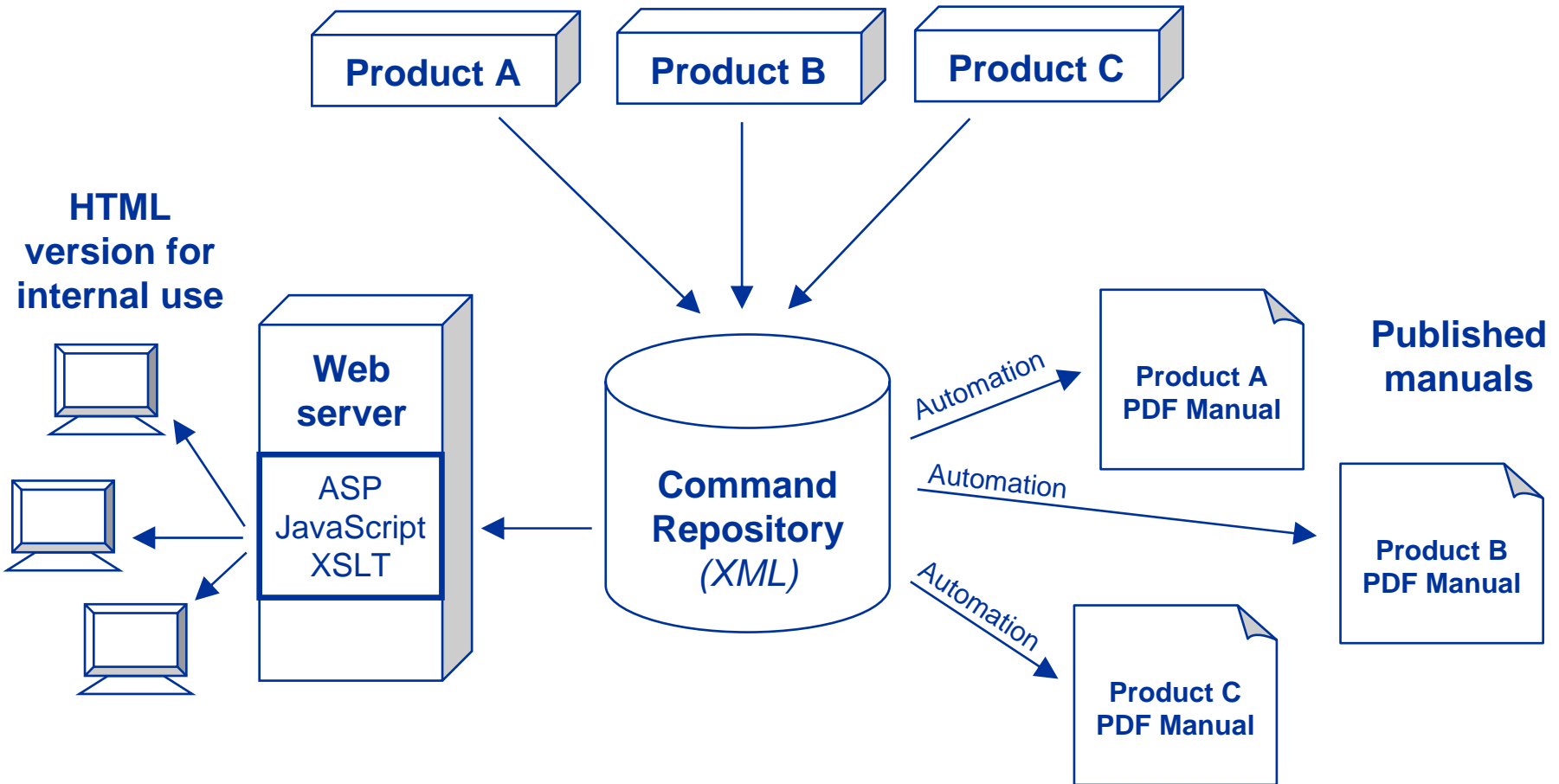
- # FrameMaker, in combination with the ABCM plugin (free! www.weststreetconsulting.com) is used to filter and publish the information upon demand for the required output.
- # FrameMaker is the tool of choice because of the ability to use structure-smart add-ons like ABCM and its ability to publish beautiful PDFs from XML content.



Some improvement, but not done yet...



Solution #2 - Real-time, web-based publishing



The command reference in the web viewer - Sample screenshot

The screenshot shows a web browser window titled "Spirent Online Command Reference - Microsoft Internet Explorer". The address bar contains the URL: `http://inspirespent/go/sabtech/pages/techpubs_docviewer.asp?project=tl1_cmd_ref&product=3577C&audience=Confidential&topic=ACT-USER`. The page title is "TL1 Command Reference".

At the top right, the Spirent Communications logo is displayed with the tagline "Inspired Innovation". Below the logo, there are three dropdown menus for "Product" (set to 3577C), "Version" (set to 03.00), and "Access level" (set to Confidential). A checkbox labeled "Show access level indicators" is present and unchecked.

A red warning message states: "WARNING: This content (or subsections) contains CONFIDENTIAL information." Below this, the command "ACT-USER" is listed with the subcommand "Activate User".

Navigation links include "Description", "Input format", "Normal response format", and "Error response format". The "Description" section contains the text: "Logs a user into the test system. The specified username and password will determine the level of functionality available afterwards." A "TOP" link is provided below the description.

The "Input format" section shows the command syntax: `ACT-USER: [<sid>]:<uid>:<ctag>::<pid>;`

A table provides details for the parameters:

| Parm | Description | Valid values | Default |
|-------|---|---|---------|
| <sid> | Site identifier of the test set to receive the command. | - - - | - - - |
| <uid> | User identifier. | Any valid ASCII string up to 20 characters, noting the following common default values: | - - - |

The left sidebar contains a search box and a list of command categories including ACT-USER, CAL-TACC, CANC-USER, CHG-ACCMD-ICON, CHG-LOGFILTER, CHG-PRTPAR, CHG-PTRST, CHG-SPLTSUPV, CONN-LOCL-IO, CONN-TACC, CPY-MEM, DISC-MEAS, DISC-TACC, ED-ATTR-EQPT, ED-ATTR-HW, ED-ATTR-IP, ED-ATTR-LINK, ED-DAT, ED-DB-IPTST, ED-DB-TAP, ED-DB-TAU, and ED-DB-TSTPREF.

Summary of benefits so far

- ⊕ Consolidation of product data into a single repository, with significant leveraging of reuse through conditional content
- ⊕ An automated path for generating PDF manuals from the repository, rather than maintaining separate, redundant FrameMaker books
- ⊕ The most current data is available in real time to internal personnel via a web browser, which in itself provides the following benefits:
 - ⊕ No more redundant authoring! Engineers never wanted to do documentation anyway... they only did it because Tech Pubs couldn't keep the information accessible.
 - ⊕ Updates to the library are viewable immediately. Everybody loves this.
 - ⊕ It is so much easier to get content reviewed when it is so accessible.
 - ⊕ HTML is simply a superior format for this type of document.

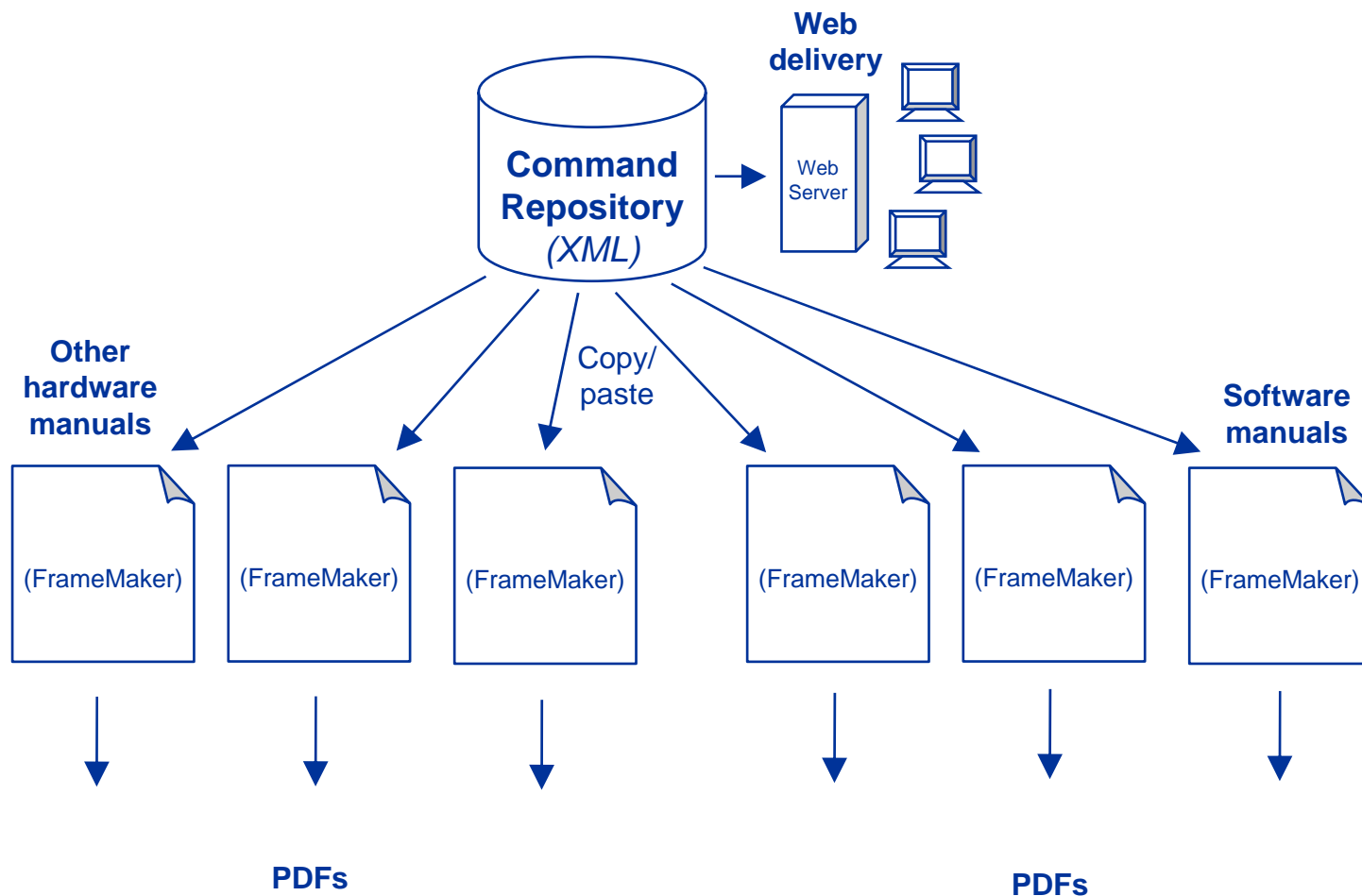
So we've done real good, but...

Once we started to make improvements, the sheer room for more improvement became more evident. In particular, we realized that the level of redundancy extended far beyond the hardware command manuals. For example:

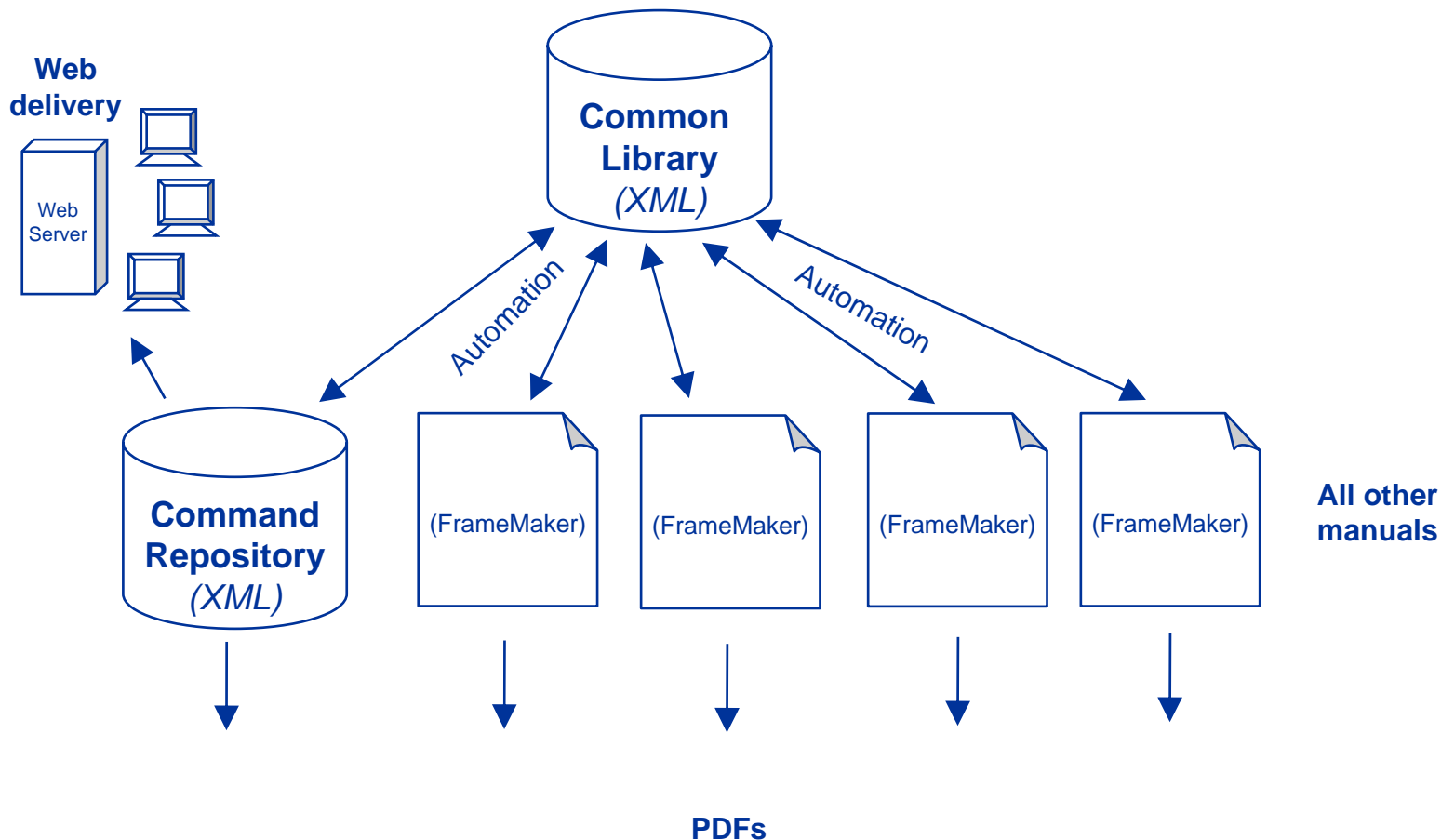
- ⊕ The hardware has other documentation as well which references functional aspects of the equipment, which is inherently redundant with information in the command manual.
- ⊕ Spirent also produces software support systems that provide customers with a means of aggregating all the deployed hardware and operating it from a remote location. The documentation for these products necessarily overlapped the hardware documentation.

Therefore, the scope of the problem now looked much like this...

A higher-level redundancy



Solution #3 - A new architecture using text insets (modular reuse)

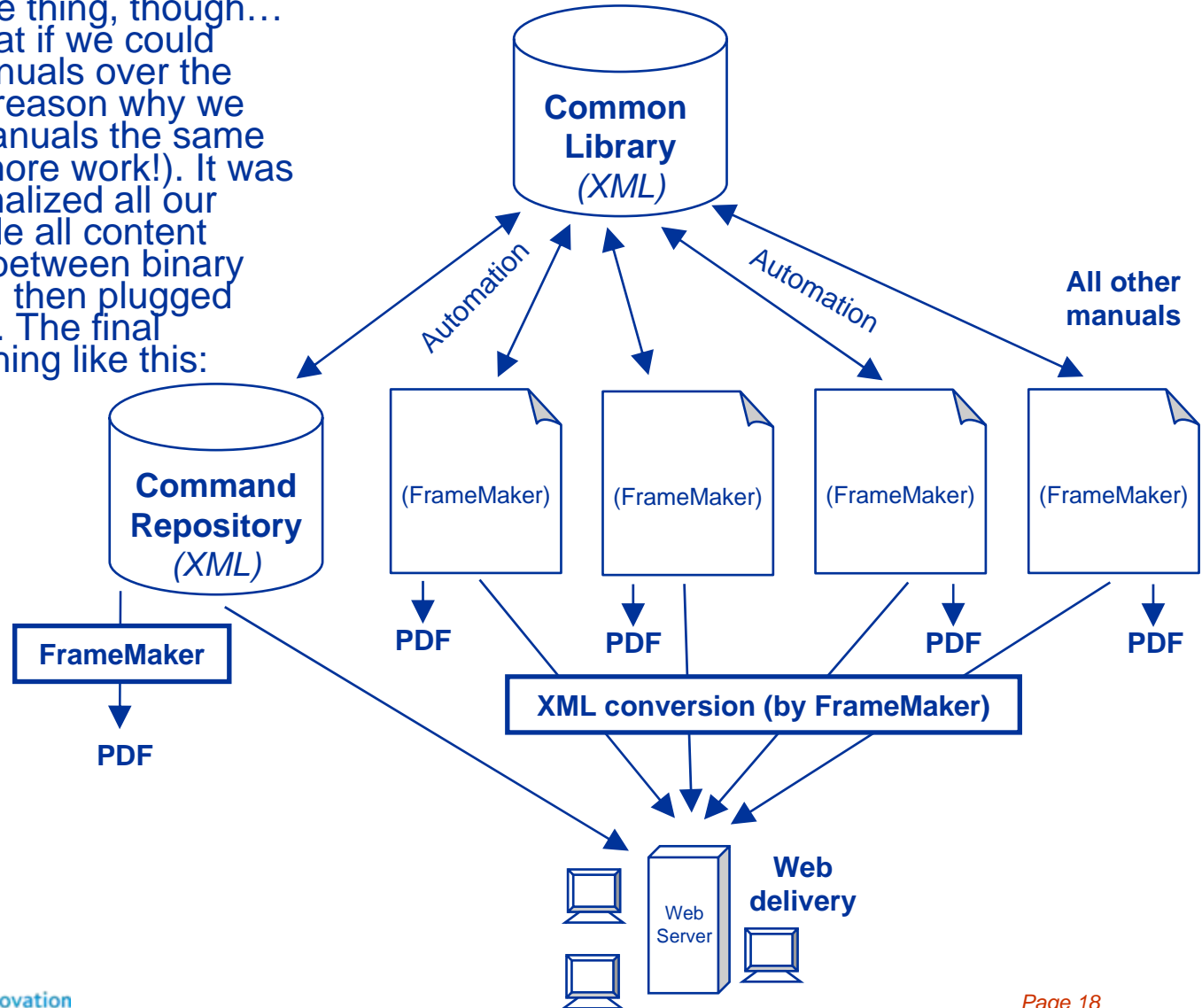


More about the text inset system

- # The amount of reuse is quite enormous. At the time of this writing, Spirent now has over 800 source modules in the library, referenced in total over 3,400 times.
- # The text inset functionality within FrameMaker is provided by the InsetPlus plugin (free! www.weststreetconsulting.com) which provides a host of advanced inset features including:
 - # Reuse of very large or very small, granular components, down to a single character if you want
 - # Source module usage tracking
- # During the JavaScript/XSLT process on the web server, the scripting recognizes these references and resolves them much as you would do manually in FrameMaker.

Solution #4 - Web delivery of ALL content

So far so good. One more thing, though... we eventually realized that if we could deliver the command manuals over the web, there was no good reason why we couldn't deliver all our manuals the same way (except that it was more work!). It was too tempting, so we normalized all our structure definitions, made all content seamlessly transferable between binary FM documents and XML, then plugged them into the web server. The final architecture looks something like this:



A software user guide in the web viewer - Sample screenshot

The screenshot shows a Microsoft Internet Explorer browser window displaying the SSCentral User/Admin Guide (3.000 DRAFT). The browser's address bar shows the URL: http://insidespirent/go/sabtech/pages/techpubs_docviewer.asp?project=sscentral_uag. The page title is "SSCentral User/Admin Guide (3.000 DRAFT)". The Spirent Communications logo is visible in the top right corner.

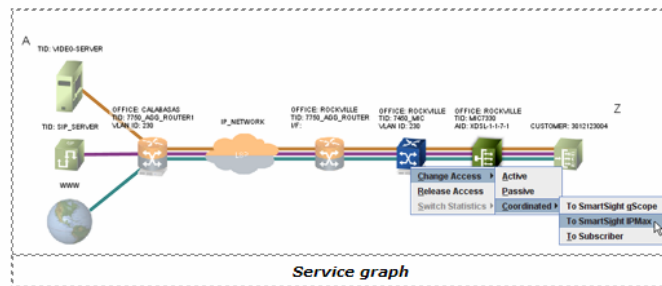
The page content includes a "Filter" dropdown menu set to "SSC_Internal_Version". A left sidebar contains a navigation menu with the following items:

- * Archives (Main page)
- Search
- Metro Troubleshooting Guide
- Introduction
- About Triple-Play
- Configuration and Maintenance
- Getting Started
 - About the system
 - System access and login (web interface)
 - Web GUI overview
 - Overall Web GUI architecture
 - Configuring the web browser and works
 - Service graph**
 - Test results and reporting
 - Stop and rerun buttons
 - Viewing inventory data for the accessed
 - Network resources and service inventories
 - Service and Component Access**
 - Data Testing
 - Voice (VoIP) Testing
 - Video Testing
 - DSL and Copper Loop Testing

Getting Started > Web GUI overview

Service graph

The web interface includes a graphical display of the accessed service. The icons represent NEs, test sets, network "clouds," and other network components. The graph is the portal to all NE access and testing through the web interface, using popup menus to launch nearly all activities.



The graph includes features such as:

- Checkboxes to control which service path(s) are shown, for FTTN services.
- A slider to control the zoom setting.

The service graph appears automatically upon service load, based on the configuration found in the resource server and the service inventory. Note that popup menus are customized

Overall summary of results

- # Redundancy virtually eliminated and quality of output improved dramatically.
- # Content accessibility is 100% better for all consumers.
- # One tech writer is now able to do the work that formerly required 4 or 5 writers (and still find time to goof off at conferences!)
- # Using rough computations based on personnel costs and normal productivity of a tech writer, the savings to Spirent can be easily shown to be in the millions of dollars per year. These numbers look very good on a resume.
- # The professional skills of the remaining members of TechPubs are far more advanced than they were three years ago, in a high-demand area of expertise.
- # The corporate perception of TechPubs has shifted from doggie-doo to the Can-Do Crew.

Technology roundup

Web viewer

- ✦ Uses XML files as the input
- ✦ Uses ASP with JavaScript to drive the process on a Microsoft IIS web server
- ✦ Most processing is server-side and all input files reside on the web server
- ✦ When a page is requested, the JavaScript:
 - ✦ Using the Microsoft XML (MSXML) DOM implementation, opens the proper XML file and locates the desired data
 - ✦ Resolves any text insets and updates with the latest content
 - ✦ Filters the conditional content according to supplied parameters
 - ✦ Does some other cleanup such as cross-reference resolution and image path adjustment
 - ✦ Sends the prepped XML to MSXML engine for transformation to HTML, using an assigned stylesheet
 - ✦ Sends the transformed results to the requesting web browser

Technology roundup

Authoring tool – Adobe FrameMaker

- ✦ Provides a top-notch, friendly editor for structured technical content, whether structured FrameMaker files or genuine XML
- ✦ Allows the use of plugins and scripting for advanced content reuse features that leverage the versatility of structural metadata. Two plugins used by Spirent (both available free at www.weststreetconsulting.com):
 - ✦ **ABCM** (Attribute-based condition management) – Adapts the conditional text model to use structural attributes to denote conditions. Among other benefits, it allows a much more granular and manageable use of multiple, overlapping conditions.
 - ✦ **InsetPlus** – Adapts the text inset model to use structural metadata to mark module sources and references, allowing better management and more advanced features.
- ✦ Provides the concept of a “structure application” to import and format XML content, making it an ideal tool for publishing PDFs from XML.
- ✦ When used to edit XML that is used in some post-process (like the Spirent webviewer, it works great. When used to edit regular files that are never intended for XML, it still works great.

Authoring and publishing

Examples of authoring and publishing tasks at Spirent

- ✦ To insert a text inset, open the XML or structured FrameMaker file and use InsetPlus to locate the module and insert the reference.
- ✦ To create conditional text, edit the appropriate attributes of the applicable element. When necessary, configure the ABCM filtering and coloring settings to produce the desired output.
- ✦ To make an update to the command reference, open the proper XML file, make the change, save the file, then export it to the web server.
- ✦ To make a change to a manual that resides in binary FM format, edit it like any other FM file. If the manual is also configured in the web viewer, save the file as XML too and export the XML file.

Final notes

- ✦ XML, ASP, JavaScript, MSXML, InsetPlus, and ABCM are effectively free! There is no need to buy expensive tools to get remarkable results, if you have the will to make it happen.
- ✦ Don't be afraid to get in there and tinker with these technologies. Tech writers often encounter technical difficulties in their workflow, but then completely forget that they have the ultimate technology tool right in front of them... a computer! Use it!
- ✦ There are a wealth of free resources on the web for learning about new tools and technologies. One of note is the tutorial site www.w3schools.com.
- ✦ Your first name is Technical for a reason... don't forget that. Beautiful prose is a lovely thing, but you shouldn't bet your professional future on it in an increasingly technical world.
- ✦ For an electronic copy of this presentation:
<http://www.weststreetconsulting.com/misc/russ-neostc2007-pres.pdf>
- ✦ For an electronic copy of the original proposal that was presented to management to get this project started:
<http://www.weststreetconsulting.com/misc/russ-xml-project-proposal.pdf>

Thanks so much! Questions?