Understanding the Issues of Data Conversion

Technical Track – Legacy Data Conversion Methodology, Issues, and Best Practices

Mike Cook

S1000D User Forum 2012
June 18-21, 2012

When Do You Need to Do Conversion?

- New project where legacy data must be moved forward
- Mid-stream upgrade where a project may be using an earlier issue of S1000D
- Customer demands a move to S1000D
- *Current process costs too much to continue doing it the old way and S1000D provides cost benefits*
- Move from a different specification or standard (DITA, iSpec 2200, etc) to S1000D
- *Wiring or part data exported from a CAD/CAM system for use in wrngdata.xsd and ipd.xsd based data modules – a form of data conversion which is not part of the mainstream view*

Worst Case Scenarios

- Continual conversion from various source systems that do not use S1000D or any form of structured content (SGML or XML)
  - It's not uncommon for upstream source systems to be in one format and your project must deliver using S1000D in XML. In this scenario, continual conversion of content may be necessary. Many businesses are silo'd such that wiring data, part data, engineering data, and training information use different data formats.
- You do NOT have control of the source data
  - When making a move to S1000D, it is always a good idea to try and leverage the change to S1000D and identify if any of the upstream systems can adopt changes to make improvements to their processes.

What You Can Do to Reduce Source Control Chaos

- Open a dialog with data source owners
  - Sometimes it's as simple as getting everyone to the table at the same time. If they all agree the current content development process is not what it could be, it's easier to implement change.
- Define audit trails and use data source ownership to push for changes
  - Psychological pressure can work too. Pushing the potential failure of a project upstream into the source data supply chain can affect change. Though this isn't a nice thing to do, if it's true and everyone upstream knows it, if the conditions are right, you can affect positive change.
- Renegotiate contracts with customers
  - This may be necessary to offset costs – passing some of the cost of development work off to your customers for increased functionality that helps their ROI – increased funding improves chances of paying for upstream changes

Conversion Options

- The most common options available to project owners are:
  - **Automated conversion**
  - **Manual conversion**
  - **A mix of both automation and manual**
  - ***Contract it out***
- Each of these options are driven by the current conditions of your business. If you have an IT department capable of handling the development of conversion software, they can be a huge benefit. COTS software solutions are capable of managing some data conversion to S1000D.
- Manual conversion has significant benefits too. You can train your authoring and editing staff to create data modules correctly by converting the source data manually. This only works for one time conversion projects (no continual conversion requirement).
- Depending on where your source data is coming from, there is a high probability source data will need to be converted continually. For example, wiring data and part data are commonly in upstream systems that have nothing to do with S1000D data structures.
- Contracting out your data conversion is generally going to fit into the same space as a manual conversion – it's as if you're doing it internally but you've brought someone else on board to do it – someone who already has experience converting content to the new format.

Automated Conversion

- What makes sense to automate
  - Picking your fights is just as important as picking what can be automated. Just because you want to use automation doesn't mean you can (or should). In many cases you are limited by the structure of your source data.
- Figure it out on your own or out source it
  - If your IT department can handle it, start there. If not, examine potential COTS solutions (there are a few). It is very hard to provide an automated solution if the original content is in Microsoft Word, free form text, or a non-structured file format. If you are already using a structured content methodology – like DITA or iSpec 2200, then you stand a much better chance of converting your content.
- Performing QA
  - Just because you think you've successfully converted your content, doesn't mean you did. You must go through each data module and validate the content was properly inserted into the element and attribute structures.
  - In many cases, boilerplate content was used in much of the <identAndStatusSection>. You'll need to find the data modules that need tweaks to the values in the status block and update them as soon as possible, since these elements and attributes tend to be overlooked after conversion.

Manual Conversion

- What makes sense to do manually
  - In most cases, anything that CAN'T be done using automation must be done manually. However, you may choose not to use automation at all. If you have a small project, for example under 2,000 data modules, it might be better to manually convert everything instead of using automation. The reason is, automation commonly does not put all the NEW information in for the new features you want to take advantage of (going from a feature poor to a feature rich environment).
- Do it yourself or out source it
  - Manual conversion is usually done in house. It can serve as both a training paradigm, QA pass, and upgrade all at the same time. Regardless of which method you choose, you must have every data module touched by someone at some point in the process. If you do it as part of the manual upgrade, you've solved three problems all at the same time and possibly saved a considerable amount of time and money (in the long run).
- Performing QA
  - QA is generally less of an issue if a conversion is done manually – since you're more likely to use existing QA (workflow or edit promotion cycle) processes you have in house. Or, as part of the conversion, you add in new editing processes as part of your workflow.

A Mix of Both Automated and Manual

- There is rarely a complete automated solution for conversion
- It is more common to have a mix of the two
- Mixing has the advantage of leveraging the best of each method
- Automation can be used for the “coarse” conformance of data being pushed into a data module. For example, creation of a data module, insertion of boilerplate into the <identAndStatusSection> pushing the bulk of the data into the <content> section of the data module.
- Manual conversion can then take over and massage the content into the appropriate places within the data module that automation could not do. Apply new information into new elements and attributes. Leverage new capabilities by enabling applicability, change markup, reason for update, acronyms, etc so further automation can be used on the backend of the publishing system to generate front matter.

Contract it Out

- Depending on your project, company size, need for advanced capabilities, etc it may be more beneficial or cost effective to contract out the conversion effort.
- Companies with expertise in conversion of content from various file formats to S1000D may be better at getting the job done than trying to do it yourself. Especially if it's the first time you've tried to implement S1000D.
- In this case, you have two alternatives:
  1. Train some of your staff in S1000D.
  2. Go straight to a conversion company and learn as you go.
- The problem with going straight to a conversion company is you don't know if you can save any money doing it yourself. What you don't know can significantly affect the outcome of your project. It is best to have a fundamental understanding of S1000D before you get too far down the road. Therefore, training is highly recommended.

Integration Concerns

- Augmenting the data
  - Generally, a conversion means "I'm adding new capabilities to my content". This means you're going to need to add information to various elements and attributes associated with your data – beyond what you probably already authored in your previous data format. Metadata supports how data can be used, therefore, you have more data to add for how to control the information.
- Dressing it up for the ball
  - In most cases, you need to have a specific plan regarding what new features/capabilities of S1000D you want to leverage for your project. Enabling these features takes extra effort that a conversion isn't going to help you with. Someone usually must touch each data module and add the necessary information to make it more useful. Use business rules to document and communicate this information to the entire team.
- Publication integration
  - After your information is in S1000D data modules, you can now repurpose your information to fit into any publication structure you choose. Because non-S1000D implementations tend to use publication structures that are more closely tied to physical books, you'll potentially need to consider the three way data organization structure allowed by S1000D.
    1. Master TOC (based on the SNS – tends to be used for IETPs).
    2. Publication Modules (manual hand built structure – tends to be used for printed publications).
    3. Graphical Navigation (alternate IETP navigation to content).

Making converted data pass QA using the BREX Data Module

- Now that you've converted your data, how do you find out whether or not you made any gross errors? You can eyeball all of the data modules, but is that feasible for a large project?
- The BREX data module is a great tool for performing some forms of quality assurance. However, it can't find everything – and it tends to be a little slow.
- It is suggested you use a .NET or Java application (not javascript) to validate data module content in a wholesale approach. You can build more complex test relationships using a simple VB .NET application than you can with a BREX DM. In fact, you can create automation to correct predictable errors (similar to a search and replace scenario). You'll also have a more flexible programming environment to do bigger and better things with.
- Creation of Front Matter Data Modules, using a VB.NET or Java application is actually very easy to do and highly recommended – if you've used the appropriate elements and attributes within the data modules. Make sure you know which elements and attributes to leverage automation features.

How Publishing Concerns May Drive Conversion Requirements

- Do your homework
  - Go through the business rules – knowing what you intend to do by the time all the data is converted. Knowing how to leverage the various capabilities of the schema structures goes a long way to meeting your objectives.
- What do you expect all this data to do?
  - The business rules will drive out whether you can or cannot use the capabilities of the specification to facilitate your project needs.
- Project Planning is crucial to success
  - Planning and business rules are part of the same analysis processes. Be sure you know S1000D before you start doing your business rules.
- Experience may be your best friend – call in a specialist
  - Calling in a specialist to help you convert your data is a very good idea if you're too busy working on other issues. HOWEVER, don't expect a specialist to have ESP. You MUST have someone on staff who knows S1000D and how it is meant to be used. ***Without*** someone on staff who knows the terminology and how content is meant to be authored into a data module, you stand a MUCH SMALLER chance of being successful. So get your team educated.

- Convert this!

B

Y

E

!