

S1000D in Chapter-Based Output

Laurie Hagar

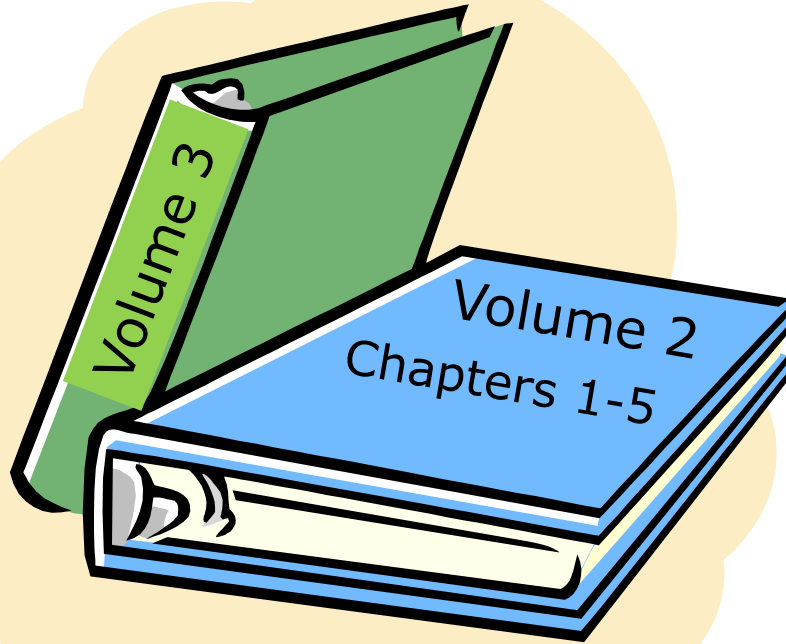
Publishing Project Manager and Consultant

SDL Structured Content Technologies

S1000D chapter-based output

- **Reasons for traditional “book” output.**
- **Options for getting there.**
- **What’s needed to get started.**
- **Problems posed by chapter output.**
- **Key issues to creating chapter-based output**
 - Within the framework of the S1000D spec.
 - Picking up where S1000D leaves off
- **Approach taken in one specific project**
- **Suggestions for success and lessons learned**

S1000D to chapter-based output



← 1995
Source data: ?

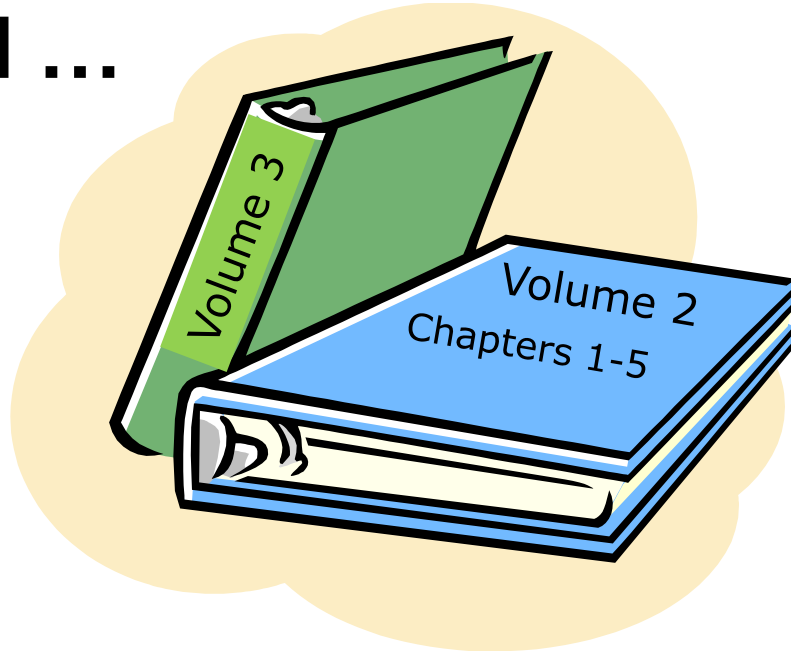
← 2002
Source data: ATA 100



← 2011
Source data: S1000D

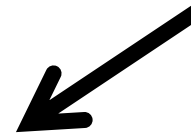
S1000D to chapter-based output

and ...



2011

Source data: S1000D



S1000D chapter output scenarios

- Legacy data produced as chapter-based publication, recently converted to S1000D and updated for IETP output. Chapter-based output remains a requirement.
- New program data authored as S1000D for IETP. Chapter-based output remains a requirement.
- New or legacy program requires S1000D and chapter output. IETP output not yet required.

S1000D chapter output options

- **Parallel data sets: risky, time-consuming, cumbersome.**
- **S1000D converted to prior spec for chapter output: development of conversion, ongoing maintenance of legacy production.**
- **Adapt S1000D to chapter-based production: enables continuance of single-source.**

S1000D chapter output: start

Extent of implementation

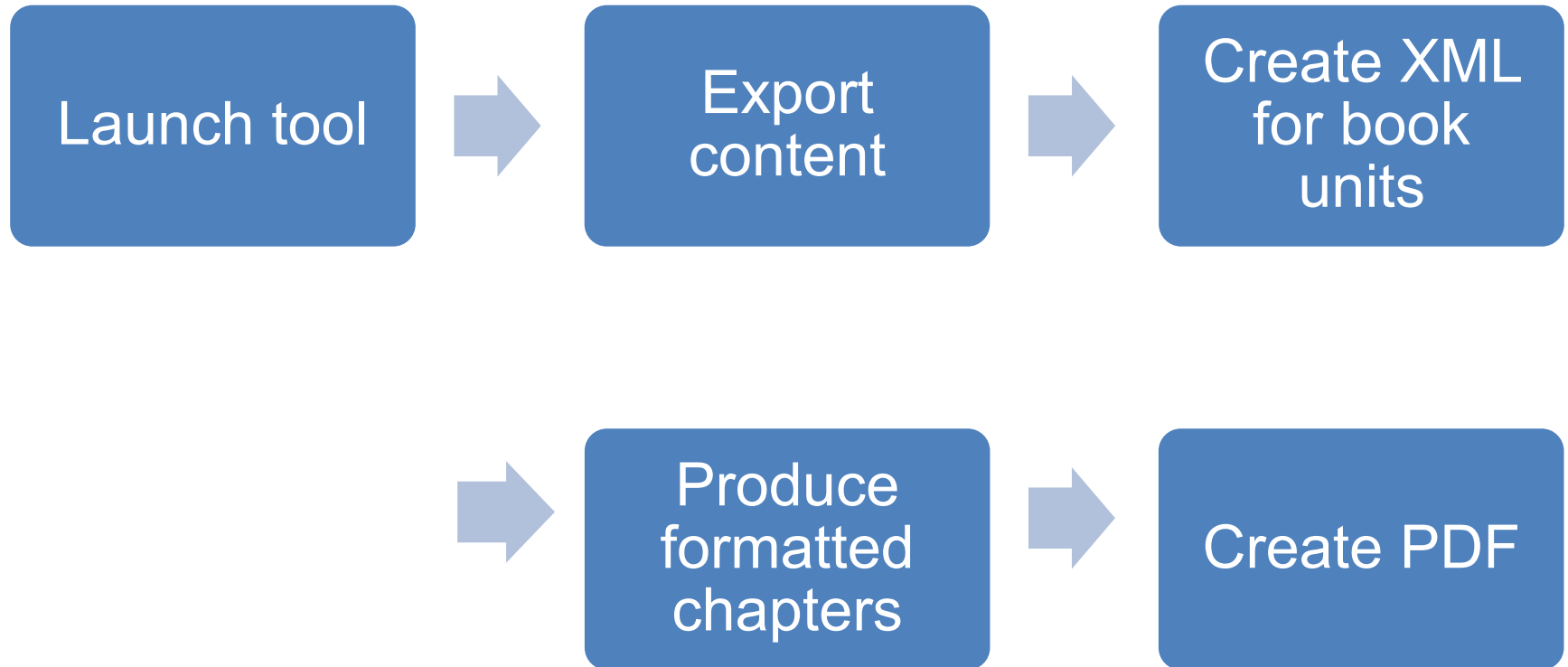
- **Ideal S1000D: followed the spec to a “T”, naming conventions, BREX, database designed for S1000D, etc.**
- **S1000D medium: using markup, S1000D database.**
- **S1000D lite: using markup ... more or less.**

S1000D chapter output: start

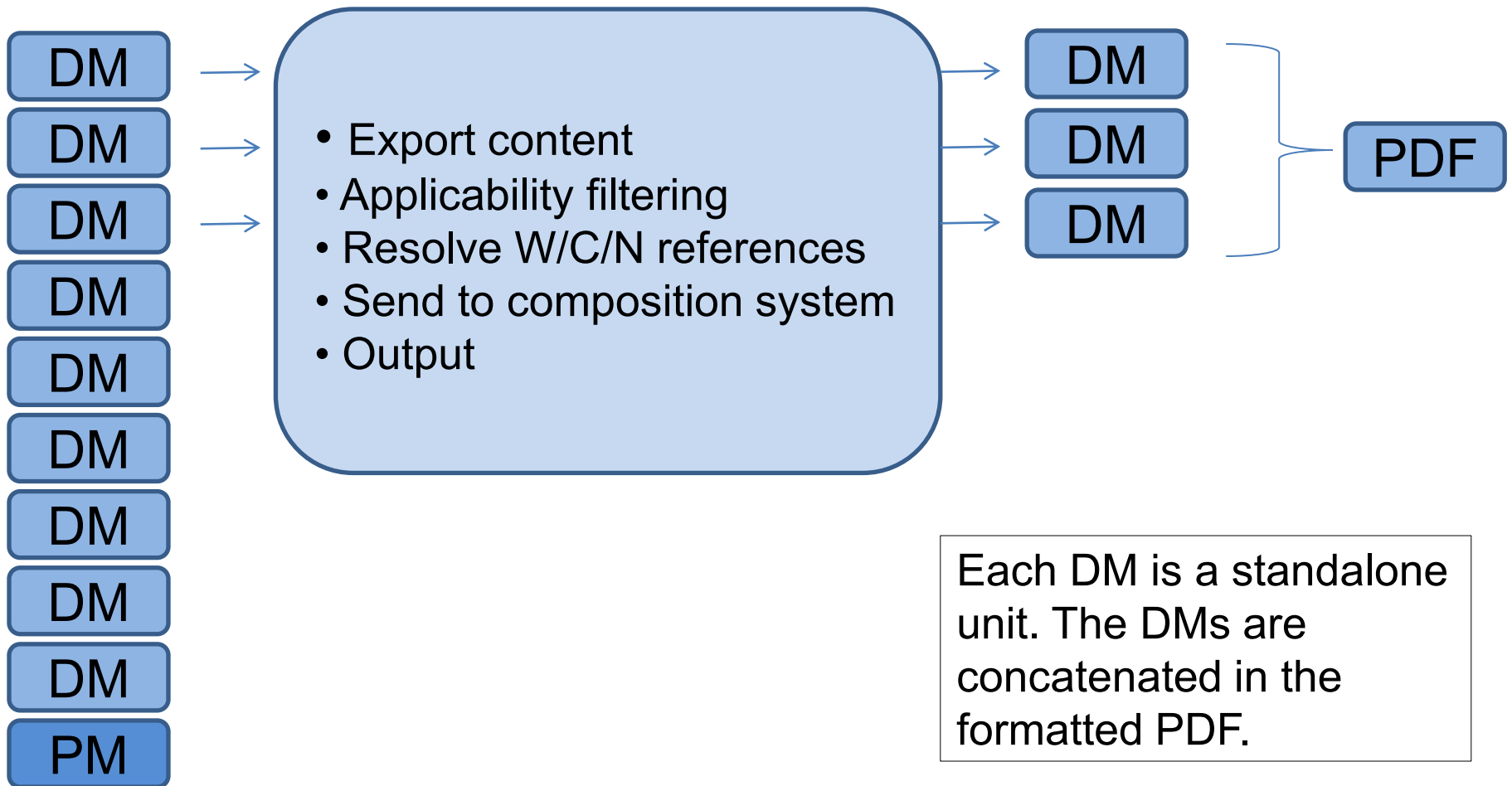
Resource expertise in:

- **S1000D markup**
- **Subject matter**
- **Spec for chapter-based output**
- **XSL**
- **Scripting language**
- **Style/format sheets**
- **Typesetting**

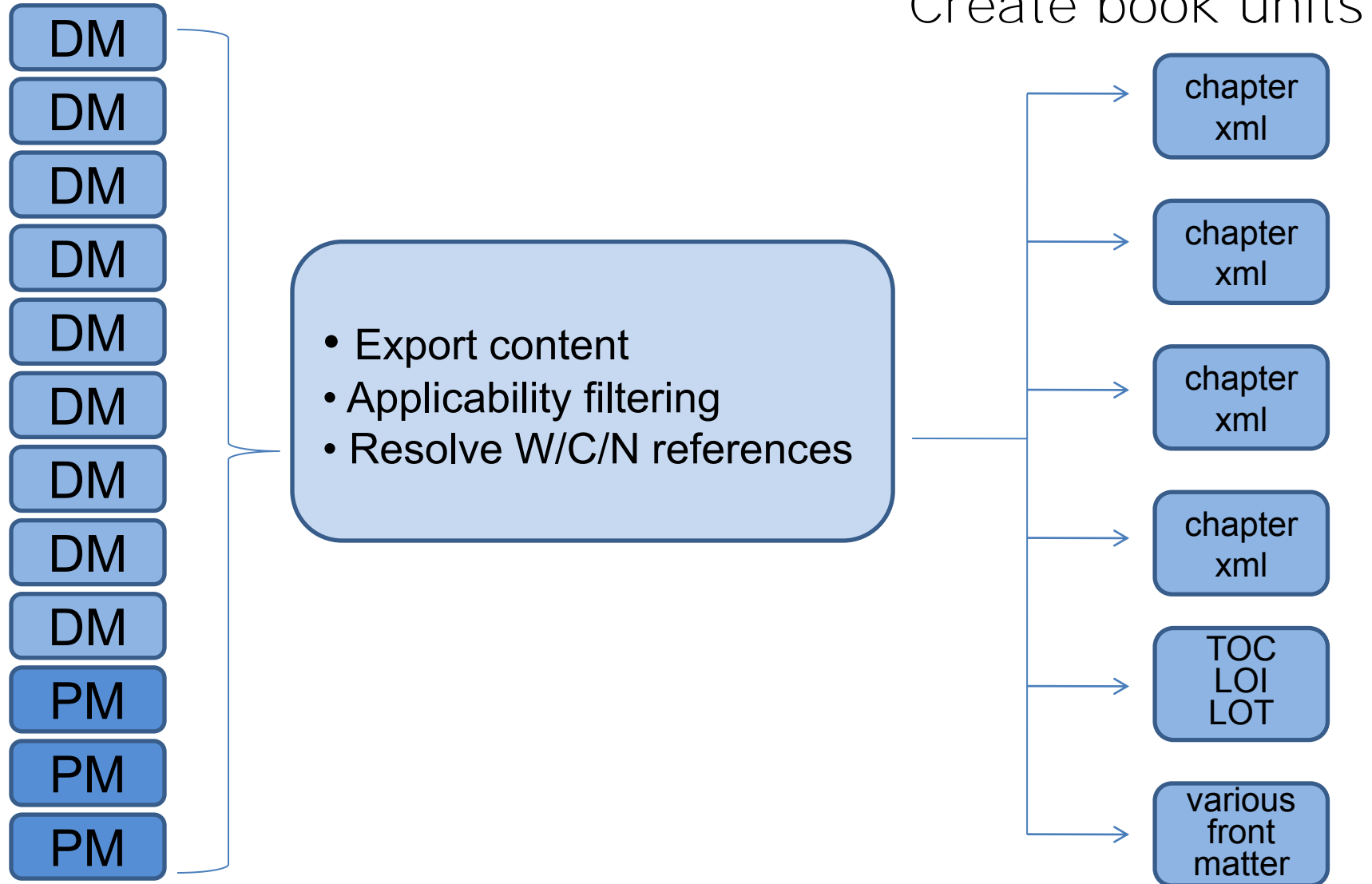
S1000D chapter output: process



S1000D standard output



S1000D „book“ output



Chapter-based requirements

- **Scope of publication (volumes? chapters? sections?)**
- **Front matter components**
- **Order of components**
- **Which components must be generated**

Chapter-based requirements (cont'd)

● Resolving cross references with labels

- “... as shown in Figure 3-4”
- “... are described in Volume 1”

● Numbering

- chapters
- paras
- figures
- tables
- steps
- other

● Change markup and revision level

Chapter-based requirements (cont'd)

● Illustrated Parts Data (IPD)

● Checklists

● Tables

- landscape
- breaking

● Figures

- placement
- sizing
- sheets

Metadata: right info on the right page

- **Dates**
- **Revision numbers**
- **Technical Manual number**
- **Program name**
- **Chapter titles**
- **Chapter/volume numbers**

How do you know what number a chapter should be?

How do you output a TM on a rear cover, if there is no DM for a rear cover?

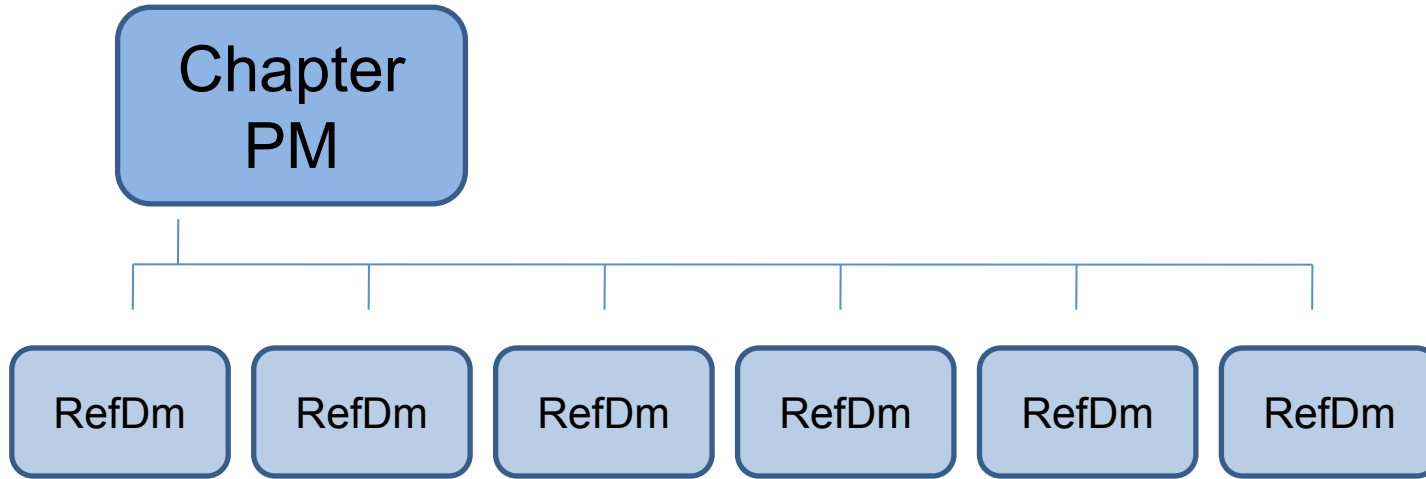
Where will the chapter title come from?

Key 1: Ordering the data

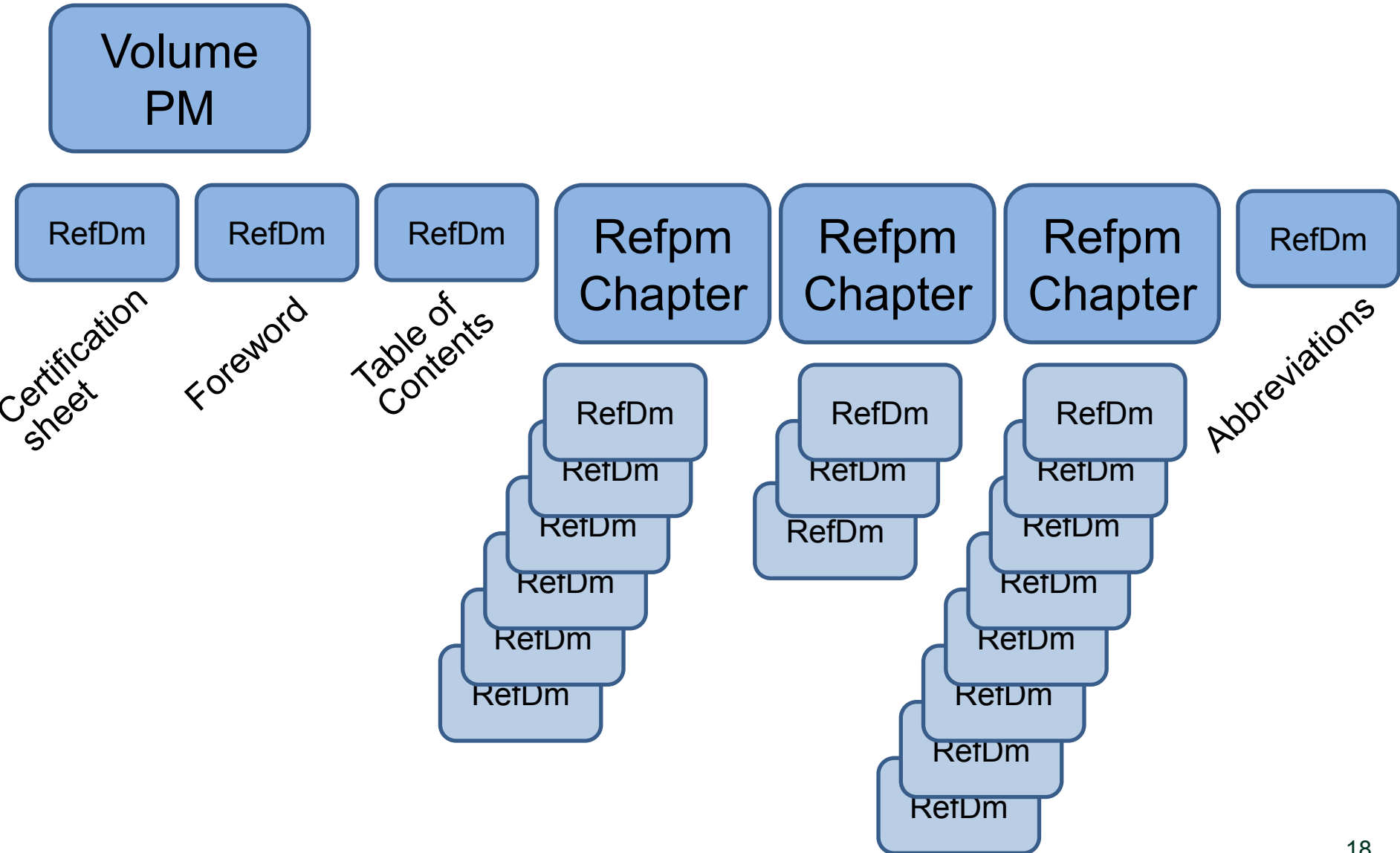
The Publication Module

- A set of PMs specifically for chapter-based output.
- Nested Pmodules may be necessary.

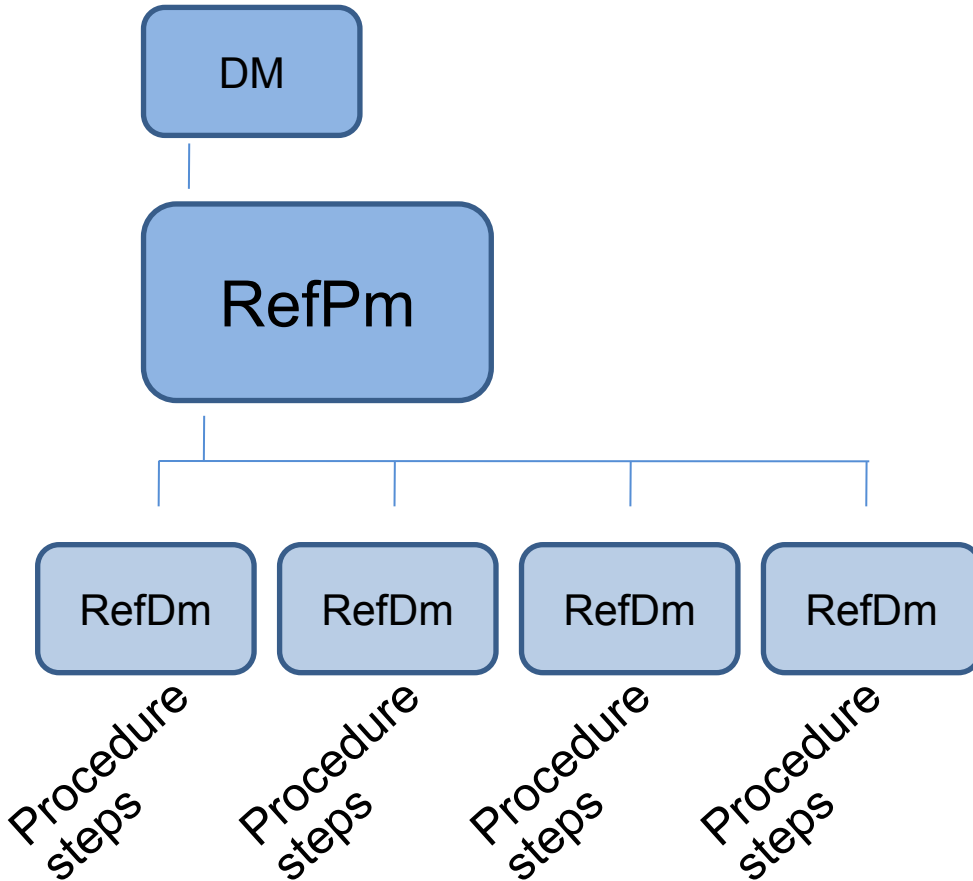
Key 1: Ordering the data (cont'd)



Key 1: Ordering the data (cont'd)

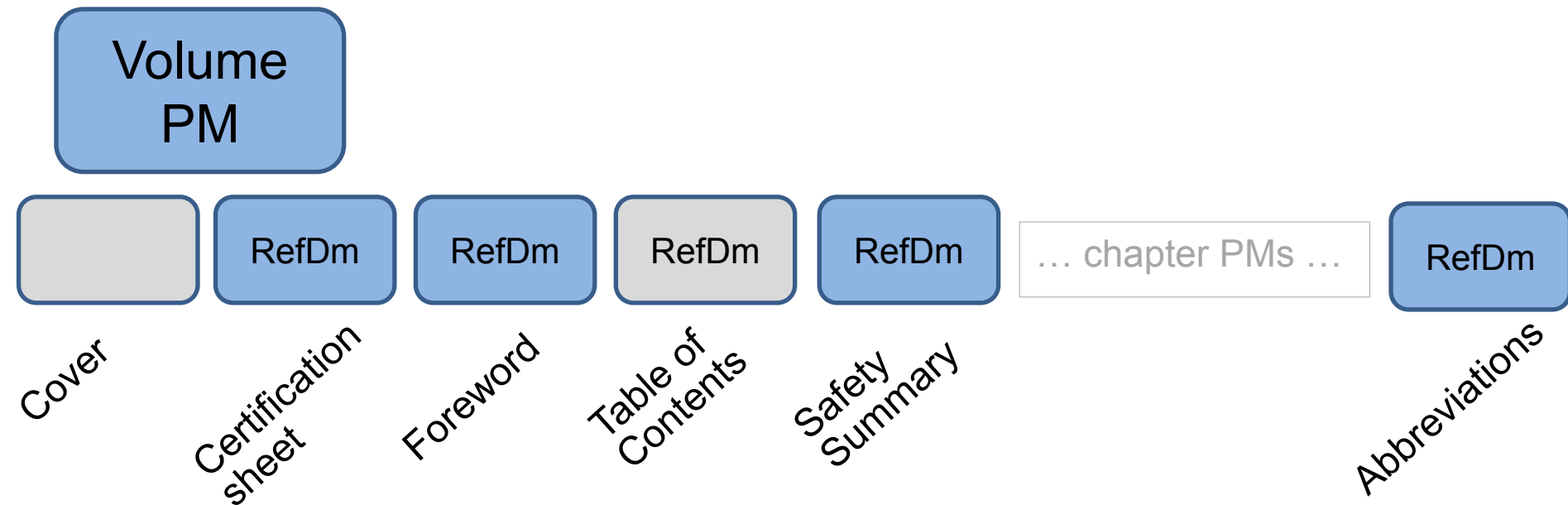


Key 1: Ordering the data (cont'd)



PMs nested in DMs
Publication Modules that are referenced from within a Data Module.

Key 1: Ordering the data – front matter



Front Matter

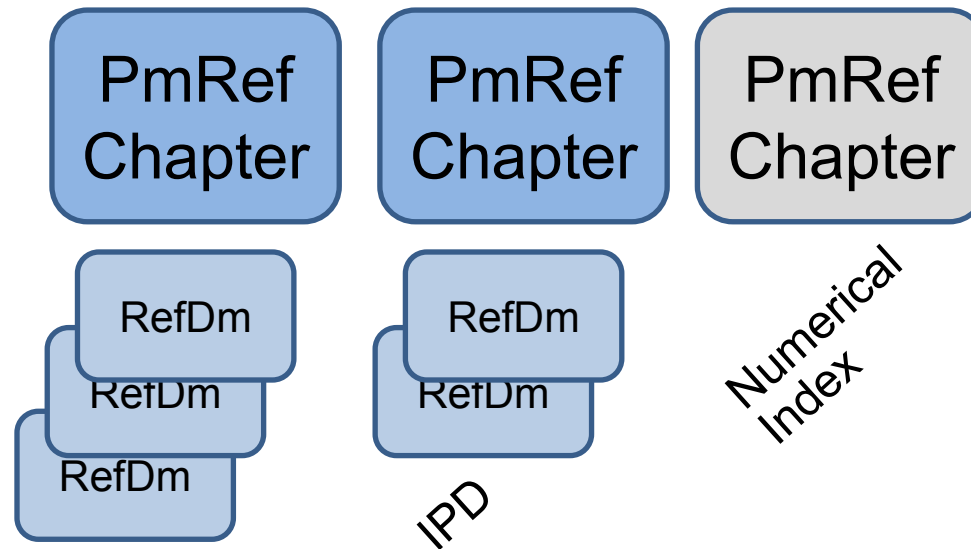
Shaded boxes represent *generated* content. There is no cover DM; its content is generated, but (importantly) is derived from PM metadata. The TOC DM has no content; it's a placeholder. Other front or rear matter may:

- need to be generated
- require placeholders
- contain content derived from other components

Key 1: Ordering the data – IPD

Volume
PM

... usual front matter ...



Illustrated Parts Catalog

Numerical index uses a placeholder, and the content is generated.

Key 2: Mining for Metadata

- **Chapter number came from <pmnumber> (in a chapter PM)**
- **Volume number came from <pmvolume>**
- **Model name came from <Modellc>**
- **Revision number came from <issno>**
 - “001” was Revision 0
 - “002” was Revision 1
 - “003” was Revision 2, etc.
- **Publication title came from <pmtitle>**
- **Technical manual number came from <pm id=“xxx”>**
- **Stock number came from <remarks>**
 - pm id attribute
- **<incode> is useful throughout to determine when you are in a certain component that requires special processing, either for formatting or generating content**

Key 3: References & Cross References

● **Unique IDs across the entire project!**

- Prefix for element type: fig, tab, step, csn, etc.
- DMC
- Sequential number
- Example: sp1-MODELIC-F1-22-00-0000-00000-018E-C-0002

● **xref xidtype populated with destination type: fig, tab, step, csn, etc.**

● **References to volumes or chapters use pubcode of the target volume or chapter PM.**

● **References within a chapter use the ID of the target attribute.**

S1000D chapter output: an approach

● **XML files based on the chapter and/or volume PM that contain:**

- Contents of PM
- Contents of all the referenced DMs (and perhaps nested PMs)

This will provide access to metadata that you may need for formatting or other processing.

S1000D chapter output: an approach

- **An ID registry to keep track of where IDs originate – which chapter, which volume.**
 - Requires two passes at processing; first, to populate the registry, and then to resolve earlier references to targets that came later in the publication.
 - Determine all of the things that can be referenced.
 - Determine all of the „to“ and „from“ locations in the context of the book.

S1000D chapter output: an approach

● Transformation

- Add but don't subtract: keep the basic S1000D structure, but add what you need to facilitate formatted pages:
- Processing Instructions
- Attributes
- New elements

Keep to general rules of well-formed XML, but by the time you reach composition, parsing often is not a requirement.

S1000D chapter output: an approach

● **Formatting (a topic by itself!)**

- Numbering styles
- Making tables fit
- Widow control
- Keeping figures with preceding data
- Marking changes
- Outputting latest revision number occurring in a page
- Indenting of subtopics, steps, lists

● **With planning, it's possible to create chapter-friendly output that can serve as input to different specification formats.**

- Configuration file for numbering styles

Lessons learned

- Expertise in S1000D is not essential, but the more you know, the better it goes.
- Not just expertise, but understanding the reasons and communicating them.
- Know the long view (or pay now, or pay later).
- Document, document, document.
- Test, test, test.
- Robust sampling.
- Review every page of a sample document carefully, with many Post-its handy.
- It's always bigger than you think.
- It's *always* bigger than you think.

Technical project management

● Scope

- Known vs. unknown

● Evolving requirements (style and process)

1. What is the data for this component?
2. How do I get the data for this component?
3. How do I get the data to look like the desired output?

● Collaboration

- Programmers
- Production
- Data experts
- Format experts
- Program managers

Technical project management (cont'd)

● Communication

- Keep in touch with each other
- Don't make assumptions
- Write out specifications clearly and ensure all understand

● Education

- The specs
- The software
- The technology
- The design

Questions?

Laurie Hagar

Publishing Project Manager and Consultant

SDL Structured Content Technologies

