An Innovative Use of Modular DocBook for Wide Scale Documentation in an Open Source Environment

OASIS Adoption Forum - Brussels - 6 October 2004

We will analyze in this presentation a content production system based upon the OASIS DocBook XML Standard. We will analyze the solutions that were implemented to meet traditional business production requirements for a project with strong constraints.

The presentation is made up of four parts:

1. The many constraints that constitute the problem,
2. The technological choices that are the base of the content management system,
3. Implementation details on how to solve specific problems.
4. The social aspect and future of such a system.

1. Initial constraints: Many Big Documents

A full Linux distribution represents:

- about 5 CDs full of installable applications;
- those applications range from text editors to Internet servers;

- and are available in a few dozen different languages from Afrikaans to Vietnamese.

The documentation that accompanies the various Mandrakelinux distributions:

- An Installation Guide
- A User Guide
- A Reference Manual
- System and network services documentation

All these in seven languages totalling more than 4000 pages of documentation.

2. Initial constraints: Many Versions of a Same Document

Note: the numbers below are used to refer to those constraints in Section 16.

- Many publishing formats (1): paper, PDF, HTML, online help;
- Many languages (2): more than 7, it must be easy to add more;
- Quality (3): resulting documents are meant to be sold, documentation represents a good part of a distribution's added value;
- Profile (4):There must be different books for slightly different products.

3. Context Related Constraints

- Time (5): a product may change up to the day its documentation must be printed;
- Release cycle (5b): two new versions of main product a year;
- Versatility (6): some supposedly frozen product features can suddenly be modified;
- Reduced core documentation team (3 people) on different continents (7);
- Many contributors for different tasks (proofreading, translation, etc.) scattered all over the globe (7b);
- Use of Open-Source tools (8): as far as possible;

- Content reusability (9): Other people, either in the company or outside of it, must be able to reuse some of the material in other documents, not necessarily with the same system. Likewise one must be able to easily import content available in an open format.

4. Possible Solutions

- Microsoft Word®: A lot of people don't know there are alternatives.
- Proprietary XML solutions: Documentum, Epic, Trisoft, SiberSafe, etc. Expensive and limited support for internationalization.
- Open XML solutions: Lampadas and others: immature.
- Develop a solution from scratch, based on open sourced building bricks.

5. The Approach Chosen

Specific constraints:

- A lack of finance
- Required to be an Free/Open-sourced system
- A lot of ideas

Led to the adoption of a system slowly built from existing open-source bricks, and to which several specific features were added gradually, as and when needed.

Now on to those bricks and the additional features.

6. Using XML DocBook (A)

- The standard format for the documentation of Open-Source applications;
- Separating content from presentation is crucial for constantly evolving material;
- Allows easy use of single-sourcing techniques for different documents;
- Provides many output formats out of the box, cosmetic changes are easy to perform.

Furthermore a lot of open source tools and libraries allow you to process XML: xsltproc, openjade, jadetex, perl-XML, etc.

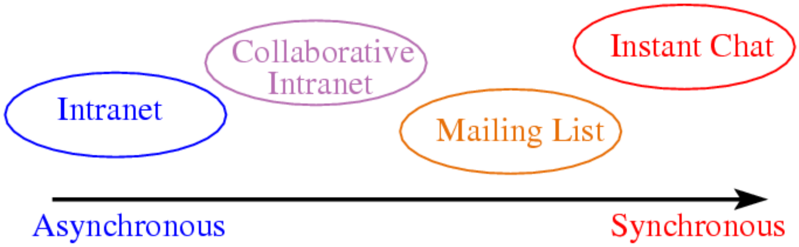
7. CVS: Concurrent Versionning System (B)

- Ideal for remote individuals to share source files;
- Manages situations where two people are modifying the same file at the same time;
- Makes sure everybody always has the latest version of a document at hand.

The use of these technologies which form the base of the system, (XML DocBook and CVS) have been quickly made possible due to the technical skills of the team. Otherwise some further developments to hide the XML and CVS complexity would have been required.

8. Internet Communication Tools (C)

In order to create an effective and friendly working environment, as people in the same physical office enjoy, a wide range of communication tools must be available.



Needless to say that a certain amount of practice and discipline is needed to reach an optimal level of efficiency, using the right tool for the right purpose.

9. Dividing Content Into Topic Modules (E)

This technique consists of isolating elementary topics into separate source files (modules).

- Try to make the modules as independent as possible, to allow for reuse in a different context;
- Allows you to rearrange the sections of a book at a glance;
- The use of little chunks of information makes content management easier and more accurate;
- Allows you to begin translating a document as soon as the first module has been written.

10. Use of Conditional Content (F)

Also called "Profiling", this technique simply relies on the "condition" attribute available in DocBook. It allows you to tag any piece of data to be included or excluded, depending on specific output needs.

- Allows you to maintain versatile modules, containing common content for various applications, but allowing different specialised versions at will for different targets;
- Provides a means to quickly derive a slightly different version from existing material, while sharing common content, thus making maintenance easier.

This is an example of a paragraph with some alternative content in it:

```
<para>Your machine boots up using a little program called
    <phrase condition="IA64">EFI</phrase>
    <phrase condition="IA32">BIOS</phrase> that ensures peripheral
    initialisation.</para>
```

When compiling my manual for a Pentium© processor, I'll direct the system to exclude all content whith `condition="IA64"`. I'll then exclude `IA32` parts for Itanium© processors.

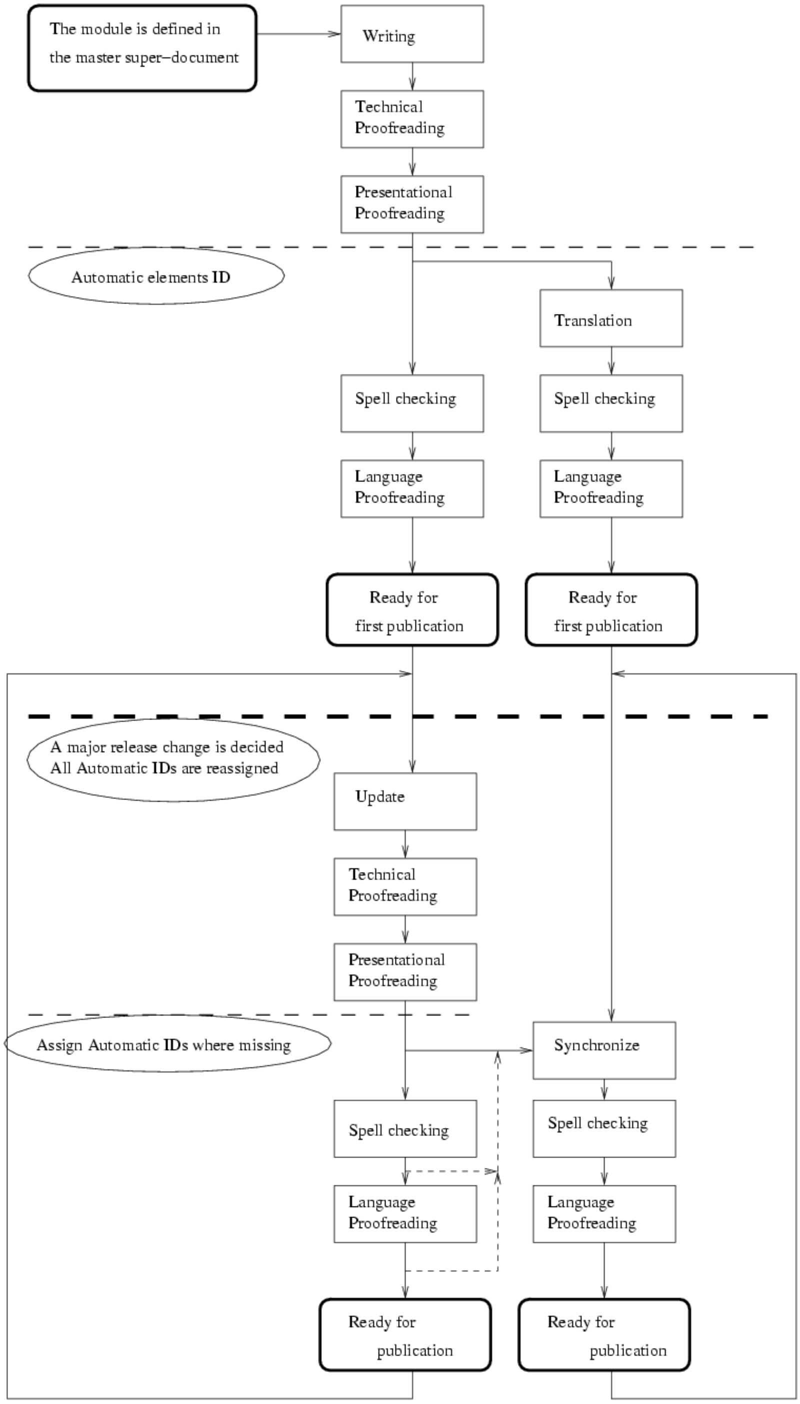
Remark: The standard DocBook stylesheets also provide a simple profiling mechanism based on the use of different attributes, but it is not as scalable as the technique we have developed.

11. Use of a Strict Document Workflow (G)

We call workflow the different states a module must pass through to reach the "ready for publishing state". This is called the third dimention of our system, the first two being modules and languages.

- Allows different people to perform different tasks on the same module in a predetermined sequential way;
- Makes sure no task (especially proofreading) is forgotten;
- Allows you to trigger translation of a newly written module before it passes all states (spell checking for example);

This is a Workflow example (`images/borges-modules-workflow.png`) with six possible states.

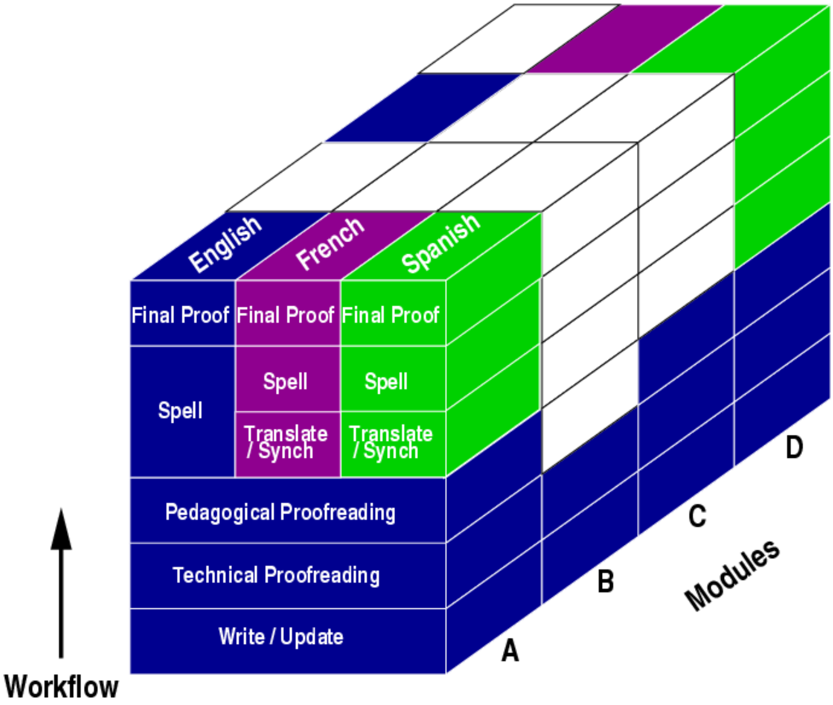

The module is defined in the master super–document
Writing
Technical Proofreading
Presentational Proofreading
Automatic elements ID
Translation
Spell checking
Spell checking
Language Proofreading
Language Proofreading
Ready for first publication
Ready for first publication
A major release change is decided
All Automatic IDs are reassigned
Update
Technical Proofreading
Presentational Proofreading
Assign Automatic IDs where missing
Synchronize
Spell checking
Spell checking
Language Proofreading
Language Proofreading
Ready for publication
Ready for publication

12. A 3-dimentional View of a Project

As we have already mentionned, The whole project can be regarded as a three dimentionnal matrix whose axis are:

- modules
- languages
- module's state

This is a symbolic view of a sample project with two languages and 4 modules.



In this diagram, each colored brick corresponds to a task which has been completed. The whole structure (document) is gradually built by piling bricks. Let's analyze the state of each module:

- A: All bricks are present, this module has been completed in all languages.
- B: The "Pedagogical Proofreading" task is missing for this module, so translations cannot begin yet.
- C: The English version has been completed while other languages have not started yet.

- D: Even though the English original module is not yet ready for publication, the translations have already been completed. This is possible because the necessary English brick ("Pedagogical Proofreading") has already been added.

13. Synchronization System (H)

This system relies on the assignment of project wide unique IDs to every chunk of data, associated to a revision number. The revision number is meant to be increased by authors each time they change the meaning of the associated chunk.

- Allows you to immediately warn translators if they need to update their translations;
- Translators are solicited only when needed;

14. The Semantic Changes Issue

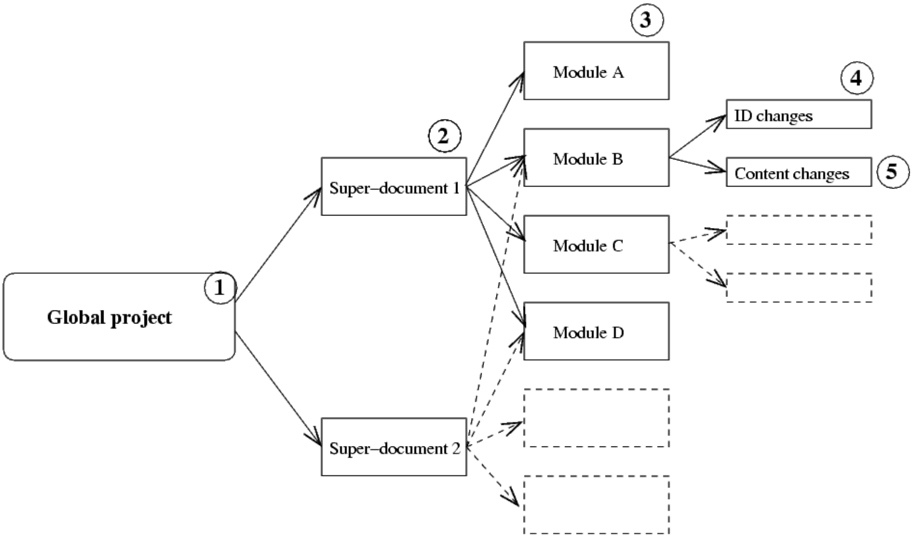
- The main drawback of this system is that authors need to explicitly say whether a change may have an impact on translations or not. Authors must be educated to change the revision number whenever they actually change the meaning of the data, but not for a spelling change for example. No automatic system can detect a semantic change in this case;
- Triangular translation update is not aways the best method;
- The traditional "PO translation system" (gettext) used to localize open source application may prove useful.

15. Automatic reports (I)

1. Global project progression reports;
2. Per manual/language progression reports (another view of the 3D matrix);
3. Module status with respect to workflow;
4. Outdated translation reports;
5. Side by side original - translation reports.

Additionally whenever a task becomes available for a module, the contributor in charge of that task receives a notice by e-mail.

This is the modules hierarchy (`images/borges-reports-diagram.png`).



16. Problems & Solutions

This table shows which choices helped solve which problems.

	1	2	3	4	5	5b	6	7	7b	8	9
A	★			★			★			★	★
B							★		★		
C						★					
E						★					★

F				★							
G			★					★	★		
H		★			★		★				
I		★							★		

17. The Social Aspect

These issues are mostly related to the fact that members of the team basically never meet physically.

- As already stated for communication, tools must be carefully chosen to minimize the absence of physical contact;
- Clear and widely accepted rules, guidelines, charters, must be designed.
- Surprisingly, there's no need to enforce strict access policies. Contributors strongly respect the work of each other;
- Each role being clearly attribued, an auto-regulating system emerges. When a kernel team is in place, new-comers either integrate with it and benefit of all trust from others, or leave.

18. A Practical Example

In April 2003, the documentation team was asked to deliver for May the documentation for the special versions of the operating system for Opteron© and Itanium2© processors. These two versions were based on the previous version (9.0) of the standard operating system which was then at 9.1. Required languages were English and French.

This is how each feature of the system was used:

- A new branch derived from 9.0 is created on CVS to hold Opteron and Itanium specific changes;
- All tasks are assigned to engineers in France and Ukraine, writers in France and Argentina, a proofreader in New-Zealand;
- All needed changes are brought to existing modules in English, using conditional content for one or the other processor or for both at once;
- Changes are mirrored in French;
- After an ultimate proofreading, all documents are generated from source and shipped.

The process lasted one month, resulting in a 450 page document published in 8 versions: for Opteron and Itanium, in English and French in both PDF and HTML.

19. The Future of Borges

- The system we described in this presentation is implemented in the open sourced Borges CMS (`http://www.mandrakelinux.com/en/doc/project/Borges/`) software;
- This system, though still intensively used, has reached its limits in terms of capacity to manage large content volumes;
- A new design (`http://borges-dms.sourceforge.net/`) is being worked out to blow those limitations and bring unprecedented features.

Any help to create such a dazzling open source system will be appreciated.

20. Conclusions

- Available software coupled together has allowed us to build a customized and powerful Content Management System without heavy programming;
- Get more information about XML DocBook consulting the DocBook Wiki (`http://docbook.org/wiki/moin.cgi/FrontPage`);
- Enjoy the end of your OASIS Adoption Forum.

Special thanks to the many people who contributed to the preparation of this presentation.