WHITE PAPER

MEDIUM NEUTRAL CONTENT PRODUCTION

A PROBLEM ANALYSIS

ERIK SIEGEL

V0.69 - 23-FEB-2004 11:22

0. Table of contents

0.1. Usage and copyrights

Usage and circulation of this document and of the ideas described is not restricted. You can do with it whatever you like, provided that you mention the author (Erik Siegel) and the source URL (`www.siegel-ict.nl`).

I would however appreciate it if you told me about using this document or the ideas described in it. You can do this by simply sending me an e-mail about it (`erik@siegel-ict.nl`). Also, if you have any comments (good or bad), other ideas or complementary information, please send me an e-mail. I am looking forward to hear from you.

1. INTRODUCTION

It isn't easy, producing educational content with XML technologies. At the beginning of the lifecycle are authors with hardly any IT affinity. In the middle we need to split their output into smaller sections and reassemble it into meaningful combinations. At the end we need high quality output for print and other media. Designing the technology for this is hard, putting it into practice even harder.

So, why is this so difficult? There are XML editors, content management systems and PDF generators by the dozen. Just put some of it together and off we go.

Unfortunately, reality proves otherwise. Most XML technology on the market is geared towards relatively simple applications like websites or simple print publications. It all seems to assume that its users are IT proficient, know what an XML tag is and how to handle it.

Another problem is rooted in the holy grail of XML publishing: Medium neutrality. High quality print output requires an enormous amount of detail in the XML sources. So what to do with designers that need fine grained control over the placement of illustrations on the page? You somehow have to add all this information somewhere and before you know it, your content is no longer medium neutral but very print oriented.

Yes, it is possible to create educational books with XML technology. No, it is definitely not yet the smooth production facility we want it to be.

This whitepaper was written as a result of the, sometimes unpleasant, experiences with XML publishing of educational material. We analyzed what happened and came up with some interesting thoughts and ideas that we would like to share with you.

Necessary background for this whitepaper is some experience with XML content production. In-depth knowledge about XML is not required.

This paper is written for a presentation for the XML Europe 2004 conference.

1.1. ABOUT THE AUTHOR

I am an independent ICT architect/consultant with over 15 years of experience in the ICT industry. My main focus is on the implementation/technical side: A customer knows what is needed for the business, but often doesn't know how. As an architect I design the necessary technical components, including vendor and/or product selection. After this I accompany the implementation as a project manager, ICT architect and sometimes even as a part-time implementer/programmer. The last three years I have been working in the educational publishing industry, helping them to automate their business processes with XML related technologies. With this, my current focus is on XML: Producing, editing, storing, maintaining and processing it.

For more information, have a look at my site `www.siegel-ict.nl`.

2. BACKGROUND INFORMATION

One of my customers is ThiemeMeulenhoff, one of the three large Dutch educational publishers. ThiemeMeulenhoff publishes educational resources in almost all subject areas in primary, secondary and professional education. Publishing channels are print, cd-rom and a fast growing volume of Internet publications. It publishes around 4000 titles. ThiemeMeulenhoff is located in Zutphen and Utrecht in The Netherlands and has around 300 employees.

The market for educational publishing is changing fast. Innovation is therefore an important drive. The company has experimented with publishing using XML and other related technologies for several years now. Some of the drives behind this are:

- The need to publish much faster than before without loosing quality.
- The need to use the same content for different output channels. E.g. to be able to publish the content of a book on a web site also.
- The need to use the same content in different publications.

Looking at what an educational publisher does, you can identify the following important characteristics that make educational content management and production stand out from, for instance, the same thing for the web:

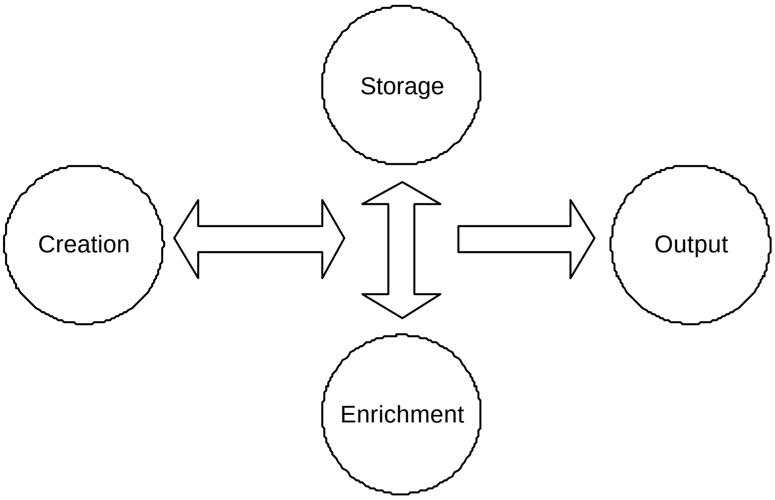
- High volume of content: For instance, an educational resource for secondary schools needs materials for at least 15 separate classes/years/levels (4 VMBO, 5 HAVO and 6 VWO). For one class/year/level you need an average of 5 books (two textbooks, one workbook, one book with answers and a teacher’s guide). This means, for one single method, a volume of at least 75 books!
- A lot of non-IT proficient authors: Usually educational texts are written by teachers who only do this part-time. They have only a very limited experience in IT, in most cases only as a user of simple applications like word processors. Experience has learned that you must not bother these people with mark-up or other details. The authors wants to be creative and should be able to concentrate on the content, the text. All other details must be handled elsewhere.
- High quality output: Print publishing is different from web publishing. It has a long history of producing quality output and this means the content model is very rich. For instance, print distinguishes several kinds of hyphens: In formulas, between words, as a hyphenation marker at the end of a line, etc. All these kinds of hyphens are presented different.

  You can argue about whether this is necessary, but all these details are there with a reason. It differentiates quality printed materials from simple home printing or web publishing. Publishers are rightly proud of this and, if ever, it is not going to change soon.

So at the beginning of a publication we have non-IT minded authors and the need to be creative, but at the end we need high quality, structured, complex output. Add to this the high volume of the content and you have an excellent mix from which challenges arise.

3. THE CHALLENGES

The challenges we faced were tightly connected to the lifecycle of the content. Below is a simplified model of this lifecycle:



- On the left the content is produced and maintained. This is where the authors and editors add their work.

  What matters here most is the information, getting the message across, the educative side of the content, not its technical correctness.
- In the middle we have some kind of storage (a Content Management System, CMS) that takes care of versioning, authorization, etc. However, besides storing, the content is enriched with all kinds of necessary data authors don't care about: Metadata, extra mark-up, links, etc. Another thing that is done here is chunking: Authors like to work in chapters, but for re-use, smaller chunks are better.

  What matters here most is more on the technical side of the content: Is it correct, does it contain enough information, is it linked to related content, can I find it again through metadata, etc.
- On the right we produce the output: print, cd-rom, web, etc. To be able to do this the content must be rich enough. However, you also have to add extra publication dependent information to your content: Should this illustration be placed left or right, how do you want to represent this table, etc. This information is not part of your bare content, but belongs to the specific publication you are making.

  What matters here most is the presentation of the content: How to make it look good.

3.1. CREATING THE CONTENT

Creating the content is a challenge because there is a serious gap between what an author wants and what a, XML minded, publisher wants. With some exaggeration:

- An author wants to be creative and just type in their texts. Technology should not stand in the way of the creative process. You must be able to change your mind easily, promote or demote paragraphs, cut and paste without problems, go smoothly from drafts to end product, etc. Structure is there but the technical details must be invisible.
- On the other hand, a XML minded publisher wants well-formed and valid XML. This is absolutely necessary for the rest of the production process.

We have looked at several XML authoring tools for use by authors and editors. There is no perfect fit. The problems are usually in one of the following categories:

- Too complicated: There are some great general-purpose XML editors, but these are much too complicated for end users like we have. With a little programming you can hide some of the complexity, but what remains is enough to scare away the casual user.
- Too simple: On the other hand there is a whole class of editors that is much too simple. They focus on either XML data forms or XML for the web. These editors can't handle the complexity that arises from structure (books, sections, chapters, paragraphs) and are usually very bad in handling XML elements with mixed content, tables or formulas.
- Too restrictive: If you create XML, your end product must be valid according to some schema or DTD. Editors that check this are experienced as too restrictive. Authors can't "play around" with the content, typing along, changing paragraph levels, dragging and dropping, etc.
- Not restrictive enough: On the other hand, most editors allow you to create content that is invalid. So what happens is that an author performs some legitimate looking action (like cutting and pasting some text) that makes the document invalid because unknowingly and invisibly tags are moved also.
- Too much freedom: Word processing software is often used as an input tool for authors. With a template and some instructions it is usually possible to get content that can be converted more or less automatic into XML. But you can never be sure, because these tools allow the user too much freedom that cannot be restricted.
- Too expensive: Last but not least, most serious tools are pretty expensive, especially if you have hundredths of authors out there.

3.2. Round tripping

Another serious challenge is in re-editing the content, something also known as round tripping the information.

As explained at the beginning of this chapter, content coming from an author is enriched with extra mark-up, metadata, etc. Sooner or later, this content will need to be revised and if the revision is major, an author will have to work with it again. So you have enriched content that you send back to an author that does not know anything about these enrichments. Probably his or her authoring tools don't even know how to handle it.

There are several ways to deal with this, but none is very elegant:

- Strip all the enrichments, let the author do its work and enrich the content again. This means a lot of double work. However, you are completely free in using different environments for authoring and enriching.
- Leave the enrichments in but render them invisible to the author. Current technology however has no means to do this reliable enough. There is a fair chance an author inadvertently changes or damages the invisible information.
- Strip the enrichments in such a way you can put them back in later. This requires some very careful programming, but in theory it should be possible, although probably not 100% reliable.

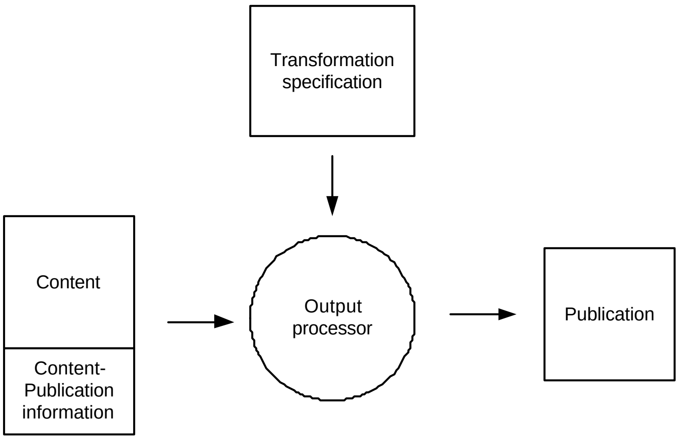
3.3. CONNECTING THE PRESENTATION DETAILS

To get from content to output media like print, you need several things:

- Something that transforms content into your end product. This is done with style sheets, XSLT transformations, etc. Such a transformation is the same for a whole group of content, e.g. a series of books.
- But if you want to output to high quality media like print, you usually need extra information for each separate publication also. This is information that combines the content and the publication. Things like: in this book, how big is this picture, should it appear on the left or on the right hand side of the paper, does it have a border, does it need clipping, etc.

  This information is different for the various output types and publications. You usually don't need it at all when publishing to the web. However, when we reprint the same material on a bigger paper size, we might want, for clarity or esthetical reasons, change it.

This is shown in the diagram below:



Having to add content-publication information complicates production: You have to model it, create it, maintain it, etc. However, for certain types of output media it is a necessity.

4. ANALYSIS

Having to work with non-IT proficient, creative, authors combined with the need for maximum reusability and high quality output creates a gap. This gap creates a tension that enlarges problems present in all content processing environment but usually stay invisible.

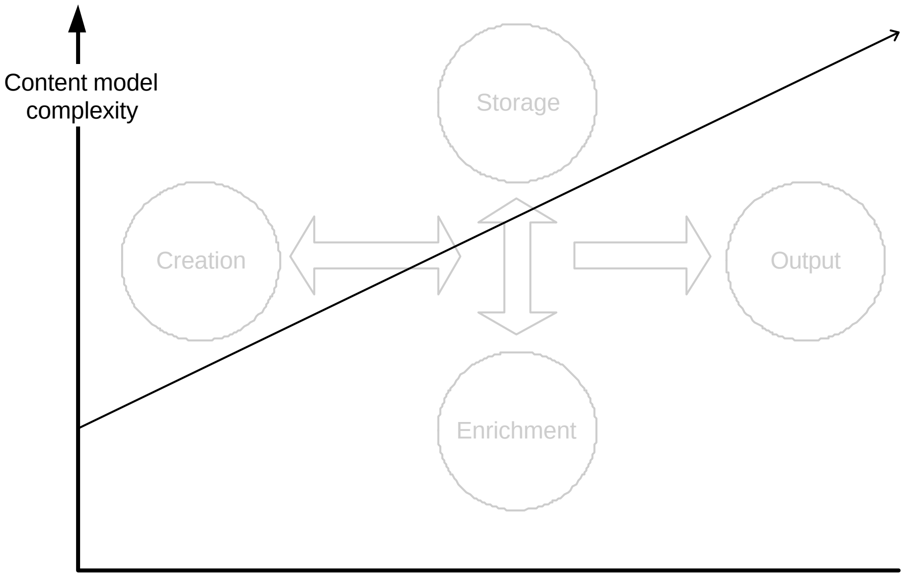
The content model is the main source of this tension. Usually you create a single content model and work with this model through all the production stages. This makes sense because creating a content model is a time consuming, complex and expensive undertaking. Also maintaining it can be quite a challenge.

However, if the tension becomes too high, it starts to make sense to use more than one content model. Otherwise you will never be able to meet the requirements for the different production stages.

This chapter explains the main differences between the content models for the different production stages.

4.1. DIFFERENT COMPLEXITY REQUIREMENTS

Between the various production stages, the level of required complexity of your content model differs. If you put it in a diagram, it looks like this:



- At the left hand side of the diagram, where authors and editors live, you do not want a complex content structure. People should concentrate on the content's topic, the text and how to educate, not on technical details. If the structure of your content is difficult this gets in the way. You can never hide it completely.

  As you can see in the diagram above, complexity and structure does not start at zero. Authors will always get guidelines and rules to work with (e.g. work in chapters, write section no more than 3 levels deep, no images in the introduction, etc.).
- In the middle your main goal is to have correct medium neutral and reusable content. This, more often than not, requires a complicated structure. All details must be present and all mark-up must be right.
- On the right, complexity increases even more. As explained in the previous chapter, you will need to add details for your specific output channels.

4.2. DIFFERENT MEDIUM NEUTRALITY REQUIREMENTS

Before going into the subject of medium neutrality in the various production stages, I would like to make a remark about the concept of “medium neutrality” itself:

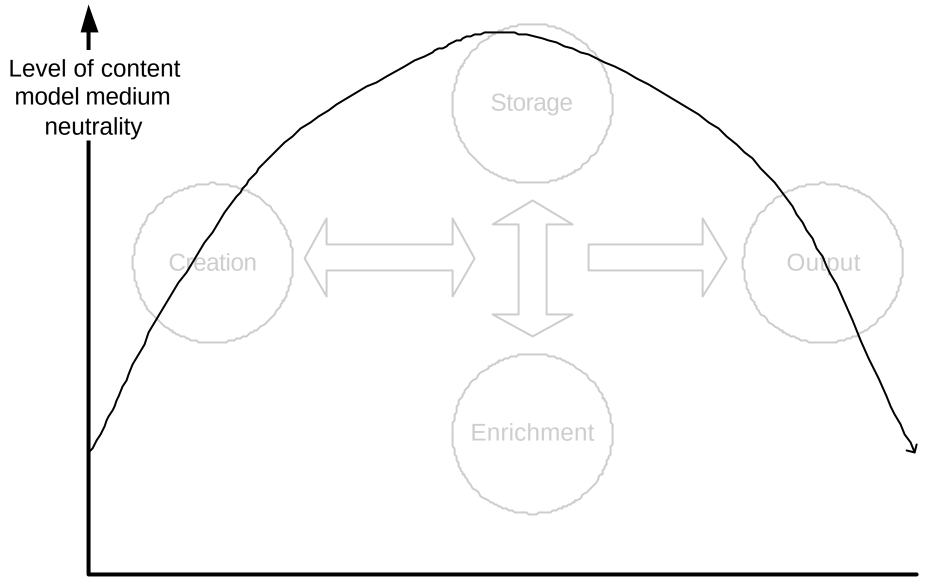
Medium neutrality is the “holy grail” of XML publishing. Theoretically, given the right medium neutral content, you can publish to any output channel without difficulties. It is therefore wise to make your content as medium neutral as you can get it.

However, pure medium neutrality can never exist. A piece of content is always made for specific types of media:

- Creating content for media you will never produce is technically and economically unsound. Every media type has its own peculiarities and resulting mark-up requirements. Some media types (like print) are much more complex and require much more details than others (like web). If you only expect to publish to simple media, why would you ever add all the mark-up for the more complex ones?
- Even if you decide to add more mark-up than you currently need, the quality of it will be low. People will not be very motivated to do it with the required precision and since it is never really used, subtle errors will creep in (it is not “debugged”).
- A second reason why adding superfluous mark-up is not a good idea, is that you will never know if it will be enough. New technology will require new mark-up details. Imagine that in ten years, the schools start working with VR helmets and data gloves. Do you expect current content with the current level of mark-up to be immediately useable?
- Last but not least, in most cases content, the wording, the text, is written for specific kinds of media. Text for the web is different from text for a book.

Pure medium neutrality does not exist. But medium neutrality for a specific set of media, for a specific number of output channels, does. It is important to remember this when we look at the measure of medium neutrality in the various production stages.

If you make a diagram of how medium neutral your content is in the various stages of production, it looks like this:



- On the left hand side the content is usually a little bit medium neutral, but in most cases not rich enough. It needs extra mark-up and other enrichments medium neutral in a useful way.
- In the middle, after enrichment, your content usually reaches its summit in medium neutrality. You are able to use it for the media and output channels you have foreseen.

- If you start using your content for real publications, the medium neutrality decreases again. You need to add publication details and other medium related information.

4.3. ANALYSIS OUTCOME

The analysis above shows that there are at least two major factors that determine the content model during the content's lifecycle. When these factors become too large, working with a single content model for the complete lifecycle is no longer a good idea. You need to look very careful at the stages your content is going through and match the stage's content model with the stage's requirements.

The table below summarizes some of the differences that are important determining your content model's requirements:

	Production	Storage and enrichment	Output generation
Focus	Text, information	Reusability	Automated output generation
Extra information	Nothing, as simple as possible	Metadata Extra mark-up	Positioning and lay-out information
Validation	Lax	Strict	Strict
Users	Authors, editors	XML and information specialists	Output specialists Automated procedures
Tool requirements	Playing around with the text Cutting/Pasting	Validation Suggestions for metadata Finding non-enriched spots	Adding lay-out information
Content model	As simple as possible	As complex as necessary for all output channels	Geared towards a specific output channel

In most of the production environments there seems to be no difference between the various content models. In some cases this is not even possible. Why doesn't this problem occur more often?

- A lot of content management systems and production environments are created for the web. (X)HTML is a simple and well-understood content model that requires no or very few conversions or enrichments.
- Other content production facilities use a more data or form oriented approach. If your authors/editors can use forms, you can hide the complexity of the underlying XML.
- Most content editing environments assume that the authors have at least some knowledge and understanding of the underlying mark-up.
- Most production environments have no need for publication specific information and can do full automatic output generation without it.

5. SOLUTIONS?

Having said all this, what kind of solutions are we looking for? What kind of things is an educational publisher looking for to overcome the hurdles described above? Some suggestions:

- **Tools that facilitate different content models**: If you want to use different content models for your different production stages, you will need to create and maintain different XML Schemas (or DTDs) and the transformations between them. However, the current tools on the market are geared towards designing single and isolated schemas and transformations. Maintaining a set of related items is not yet possible.

  Inspiration how this might be done can be found in the world of programming: Preprocessors, object orientation, modularization, etc. Development tools can put layers on top of the standards to make this possible. Some support for this is present in the current standards themselves (e.g. XML Schema's have extensible types), but when things get complicated this is not enough. Hopefully we will see more support for this in the coming versions.
- **Simple and cheap XML authoring tools**: Authoring tools that can handle text oriented XML and schema's well (like XMetaL or Epic) are neither cheap nor simple. This of course due to the fact that they have to support an awfully complicated standard and a wide range of possible user requirements.

  Maybe there is a solution in limiting the complexity of these tools. Make them simple, only support a very, very limited set of schemas but do this very, very well. Provide authors with a simplified Word like interface in which they can only use predefined styles. Make sure that actions like cutting/pasting or promoting/demoting paragraphs never invalidate the document. Of course, such an editor is still a complicated thing to create, but it is probably a lot simpler than having to support every possible schema.
- **Round tripping without information loss**: This is a hard and maybe even unsolvable problem. We have rich information that we have to edit in an environment that cannot handle all the enrichments.

  There are several technical tricks available that help but I haven't seen a satisfactory one yet. However, my feeling is that we can at least make round tripping bearable.
- **Integration of layout information**: For high quality (print) output we have to add layout information on a per publication basis to the content. This comes down to adding little pieces of information (left or right, size, etc.) to specific elements in your content (pictures, tables, etc.). This information does not belong to the content but to the publication/content combination.

  Elegantly integrating the layout information is a question of the right tools and a workable user interface. It is technically simple but just isn't there yet.

6. CONCLUSION AND SUMMARY

Educational publishers want to create high quality print output but have to work with non-IT proficient authors. This gap creates a tension in the content processing. This tension enlarges problems I think are present in all content processing environment but usually stay invisible.

Important factors that create this tension are:

- The need to be able to use non-IT authors and to keep them creative, happy and productive.
- The need for a very rich mark-up in the end products.
- The need to add a lot of metadata to the content to make it reusable.
- The need to incorporate layout information on a per publication basis somewhere in the production chain.

If you analyze what is happening, there are some grave differences between your optimal content models during the production stages. The main differences are:

- The complexity of the optimal content model increases from simple during authoring to very complex for producing output.
- The level of medium neutrality of your content reaches a summit during the storing and enriching stage.

If you have a production environment where these differences are present, you may be better of working with different content models in the separate production stages.

To support this we have to address the following consequences:

- We need design tools that are able to maintain a set of related (XML) schemas and the transformations between these.
- We need simple authoring tools but not too simple.
- We have to solve the problem of round tripping the information.
- We need an elegant way to integrate layout information on a per publication basis.