



Elkera Pty Limited

Open Publish 2005

Planning a single source publishing application for business documents

29 July 2005

Peter Meyer

Copyright, Elkera Pty Limited, 2005

Contents

Introduction	1
Vision statement and business case.....	2
Setting up the project.....	2
Don't start at the wrong end.....	3
The analytical approach.....	3
Develop requirements	5
Develop a strategic architecture and implementation plan.....	5
Select or develop a schema.....	5
Off-the-shelf schema options	5
Capacity of the schema to model the content.....	6
Capacity of the schema to provide for consistent markup of content.....	7
Ease of use for content authors	7
Evaluate & select applications.....	8
The need for requirements	8
Editor selection	8
Rendering application selection	9
CMS selection	9
Develop applications.....	10
Rollout	10

Planning a single source publishing application for business documents

This is the text of a paper presented by Peter Meyer, Managing director of Elkera, at OpenPublish, Sydney on 29 July 2005.

Introduction

This paper will discuss the critical steps in planning a single source publishing application for narrative business documents that have an important impact on ease of use of XML authoring systems for content authors.

While it is well accepted that the use of XML for single source publishing is an effective solution to the content publishing problem, many organisations have difficulty getting content into XML. The key thesis of this paper is that these difficulties arise from poorly designed DTDs or schema and inadequate planning for author needs in interface design. To overcome these problems, content publishing systems must be designed from the ground up with author usability in mind. If this is done, the switch to XML authoring can be a productivity gain to the enterprise and a further benefit in the business case for the single source publishing system.

What is single source publishing? This is the process that allows content to be maintained in one place and either used in multiple publications or published in multiple outputs (eg, print and web renditions). This might be to permit multiple versions of a publication for a product family with shared and custom content in different languages or it might be simply to allow print and web versions of a complete document.

What are narrative business documents? We are talking about such things as:

- product, process and training manuals
- technical specifications
- contracts
- regulatory policy materials
- simple marketing literature that has to be published in print and on the web
- articles, white papers and consultants reports.

In virtually all cases, it is possible to define standard publishing layouts for a class of documents and automatically apply those layouts as renditions are generated.

In all these cases, the use of XML for content management can be a useful foundation for a single source publishing strategy. Where documents are held and updated for long periods, the use of XML may also avoid technological obsolescence of the content due to changing software versions and publishing styles.

This paper is not concerned about business documents such as invoices, purchase orders and other similar structured data records. They are not narrative business documents.

Vision statement and business case

Setting up the project

The decision to develop a single source or publishing application needs to be based on a clear, comprehensive vision and business case. If this step is not completed correctly, things can go wrong very quickly.

Lets revisit the reasons for wanting to establish a single source publishing application. Assume that XYZ Enterprises develops computer software products and needs to produce software manuals for particular variations in the product. It wants to publish both print and on-line versions. Currently, they are using MS Word for document authoring. Content that is common to multiple publications is separately maintained in each publication. Inevitably, these become inconsistent. Considerable manual intervention is required to properly display documents in HTML on the web or else web publications are based on PDF documents. XYZ Enterprises wants to overcome the inaccuracies and delays in the current system. It wants to reduce the costs of manual intervention. It wants the full functionality of browsesable, searchable HTML on its web site.

There are at least two ways that XYZ Enterprise might set out to handle this problem. One way, might be for someone in IT to say "We need a CMS". Some cost saving assumptions will be developed and various CMS systems costed. On this basis, a business case might be established.

The other way is that someone in IT might say "We need to work out exactly what we want to achieve here and then the best way to go about it. We need to make sure we identify all benefits and see how we can minimise the costs of gaining those benefits. We had better make sure we understand our data and our user's needs." This will set in train a more complex process but it is more likely to produce the best results for the enterprise.

There are many variations on these themes but these two represent the basic issues that need to be considered. The consequences of each approach are considered in the next section.

Don't start at the wrong end

The "we need a CMS" approach leads to serious problems for the project. It is the equivalent of a cargo cult mentality. It is possible that the project can be made to work if XYZ is lucky and has enough money but problems along the way may include:

- XYZ may not actually need an enterprise CMS. It may already have a file system or document management system that can be used as a suitable platform with some creative system design. It is quite possible that money will be spent on systems that are not really needed because of a lack of understanding about how XML publishing systems can be created.
- The features and mode of operation of the chosen CMS will define the way that XYZ's content is organised, rather than the other way around. The process can become one of attempting to push a square peg into a round hole.
- The mind set of starting with a CMS will divert attention from other analysis that should be undertaken.
- The CMS vendor will offer packaged schema and applications that may not be the best tools for XYZ. It may be difficult to obtain the vendor's assistance to look at alternative products.
- Many stakeholder needs may not be properly identified until the system is to be installed, causing delays, disruptions and budget overruns as problems are identified and dealt with.

One way in which enterprises can fall into the "we need a CMS" trap is during preliminary investigation of options. It can be very useful to understand the features provided by various enterprise and web CMS systems and other candidate software applications during early planning. This can help in the development of requirements, if handled carefully. However, it is easy to be captured by vendors at this stage. Ideally, an expert, independent consultant should be employed to assist during the planning stages to help prevent premature capture by a system vendor.

The analytical approach

If XYZ Enterprises starts by identifying all relevant users or stakeholders in the system and then by analysing their needs, there is a good prospect XYZ will be able to identify the widest range of possible benefits. It is also more likely that the true costs of deriving those benefits will be identified. If XYZ does this correctly, it will have a robust vision and business case that will drive the rest of the project.

The initial analytical process should lead through these steps:

- Identify all system internal and external system users, interfaces to the system and relevant data flows.

- Identify the high level requirements for each interface (more detailed requirements will be gathered later).
- Understand the problems and needs of each set of users. What problems are going to be solved for those users?

In this process, it is necessary to re-consider work flows and work practices. There is no point in applying new technology to old, manual work practices.

Commonly, benefits will be derived from:

- quicker production time;
- improved accuracy of content; and
- reduced publishing costs, particularly by automation of various tasks.

Speeding up production time may have flow on benefits for other parts of the enterprise, permitting faster delivery of new products, for example.

Once the benefits are identified, these need to be related to real world work flows and work practices. Just what changes must be made to achieve those benefits? Functional estimates must be developed of the likely level of savings that are to be sought. From there, estimates of monetary savings or gains can be calculated.

The benefits side of the business case should define measurable goals that can be tested throughout the project and on completion.

The other side of the business case is to correctly identify the costs. This can only be done properly if all problems have been identified and provision made for their resolution in the overall strategy.

One of the major problems observed with some single source publishing system proposals is a complete failure to appreciate the needs of content authors in the early stages of planning. It seems to be assumed that authors will pick up XML content authoring with minimal change management issues. If this issue is not properly handled, many of the expected benefits will be eroded through delays in project roll out, high training and support costs and from problems dealing with inconsistent data.

There are two sources to this problem:

- a misplaced expectation that authors will become XML *geeks* and quickly learn a new paradigm; and
- a lack of appreciation of the role of the DTD or schema in making the work of the author simpler and in helping to ensure that data is not only valid but consistently marked up.

If the authoring team is only 2 or 3 technically minded people, these problems might be managed fairly easily. If there are 10, 20 or 30 authors, they can become a major impediment to the project. Both issues are discussed in detail in later sections.

Develop requirements

Once a business case is developed and a project established, it will be necessary to develop more detailed requirements in order to develop a comprehensive system architecture. In reality, a certain amount of high level requirements analysis will have been done for the business case. In this phase, more detail is needed.

It is likely that this requirements phase will be a middle layer in the requirements development process. Detailed requirements for specific components, such as the drafting and rendering applications may be done separately, later in the project.

Requirements must be clearly documented and be related to all the system users and stakeholders identified earlier.

Develop a strategic architecture and implementation plan

It is essential that a detailed strategic architecture is developed to determine the overall scope of the new system and how it will be integrated into existing systems. A certain amount of this is done during the business case development. In this stage, much more precision is required to guide the rest of the project.

This paper does not seek to further investigate the development of a strategic architecture.

Select or develop a schema

Off-the-shelf schema options

In an XML based single source publishing system, the DTD or schema is at the heart of every application component in the system. If the schema is replaced or changed significantly, the costs of changing or replacing software can be very high. It is essential that, before any development takes place, the schema is matched to the system requirements and that it comprehensively models all the enterprise content to produce the desired outputs for all stakeholders.

Ideally, for narrative business documents, it should not be necessary to develop a schema from scratch. The choice of available schema is growing. If an existing schema can be used, there is a good chance that some tools will be available for that schema, thereby reducing the cost of application development.

A comparison of four freely available schema that may be considered for use with narrative business documents is available on Elkera's web site at www.elkera.com/ in the white papers section under "Articles".

Some CMS vendors provide their own proprietary schema. Use of a proprietary schema that is supported by only one vendor may represent a long term lock in for all dependant software components.

A schema has to provide the representation of data needed to support the content publishing and information management requirements. Those requirements must be carefully worked out before making a schema selection. Schema selection criteria may be broken into these facets:

- capacity to model the required data;
- capacity to provide for consistency of data; and
- ease of use for content authors.

Each of these points is discussed in the following sections. Unfortunately, it is common for system designers to concentrate on the first facet, pay insufficient attention to the second and ignore the third.

Capacity of the schema to model the content

Many off-the-shelf schema provide very flexible content models that can be used for a very wide range of documents. As discussed later, this can be a problem, as much as a benefit.

Issues that need to be considered when evaluating schema include:

- Does it provide an accurate representation of the basic document structure?
Many narrative business documents contain distinct components, particularly in the front or back. Explicit markup of these components may be necessary for layout or component numbering purposes. If the schema does not provide elements for these structures, it may be necessary to create them. Subverting the use of other containers provided by the schema may be confusing and may create a misleading appearance of compatibility.
- If there is a significant amount of legacy content to be converted to XML markup, this will have a big impact on schema design. Documents created using word processor software are notoriously inconsistent in their structure. Conversion to XML is a difficult exercise. Invariably, there will be anomalous structures that do not suit the schema, unless the content models are very loose. Usually, it is necessary to choose a balance between rectifying the data and allowing a very loose schema. This requires a thorough analysis of the legacy data at the earliest stage of system planning.
- Does the schema provide adequate metadata or provision for adding metadata?
It is almost always necessary to devise a metadata structure that meets specific enterprise requirements.
- Does the schema provide adequate granularity for content re-use?

- Does the schema provide a mechanism to either constrict or extend the schema to suit the specific enterprise requirements?
- Once changes are made, will those changes create incompatibilities with off-the-shelf tools for the schema, thereby negating many of the benefits of an off-the-shelf schema?

Capacity of the schema to provide for consistent markup of content

Some off-the-shelf schema provide a vast number of elements to cater for a wide range of possible uses. They may also permit content to be marked up using several different conceptual approaches. Unless the schema is drastically constricted and content models tightened, there are several serious problems with these schema:

- Different authors will almost certainly apply different markup to the same content. Often, one author will apply different markup to the same content within a document. It can be very difficult to determine if the markup differences are based on a genuine semantic differences in the content or if they are merely idiosyncratic. This can affect content reuse and publishing consistency.
- A schema with a very large number of elements and with loose content models permits a vast number of element contexts that must be handled by rendering applications. This makes it very difficult to accurately specify publishing layouts and to develop rendering applications. Rendering applications are likely to be expensive to develop and unreliable.
- There is a great deal for authors to understand about the schema, this creates author resistance and requires extensive training. This can frustrate XML content authoring or make it unnecessarily expensive.

Ease of use for content authors

XML authoring frees authors from many tedious tasks required by word processing software. They do not have to worry about document layouts nor how to create cover pages, contents listings, headers and footers. However, these benefits can be heavily outweighed by an inconvenient interface that requires authors to constantly find the correct element and insert it in the correct location before writing content.

It is possible to conceive of XML content authoring as a substantial productivity gain in itself. Unfortunately, very few system developers approach it in this way. Rather, it is seen as something that must be endured by content authors for the greater good of the enterprise. This is an unacceptable and wasteful approach. This problem is one of the major barriers to the more widespread adoption of single source publishing systems.

XML editing tools take many different approaches to try to assist content authors to create markup as they create content. Out-of-the-box, most XML editors require that

authors must have a thorough understanding of the schema and that they must work in a tagged display to effectively insert new elements. Clearly, this is not satisfactory for the majority of content authors. The process of having to find the correct element from a list and insert it in the correct location interferes with the natural authoring process. It creates a barrier to the adoption of XML authoring. Where authors do learn to work with such a tool, productivity can be badly affected.

In any group, it can be expected that around one quarter of the authors will be able to work with a moderately complex schema without extensive training and support. Around half the group may learn to use the application but with varying degrees of efficiency. Around one quarter will likely have severe problems adapting to the new content authoring process.

Various approaches can be taken in XML editor design to minimise these problems. However, unless the schema is reduced to a very simple model that requires authors to understand only a few basic concepts, it will be difficult and costly to implement and support an XML authoring environment.

Planners of XML based single source publishing applications need to consider how to comprehensively overcome these problems and gain the full benefits of a simpler content creation system for their authors. To do so requires careful planning from the start of schema development but it will produce major benefits for the project.

Evaluate & select applications

The need for requirements

Software applications can be evaluated only on the basis of clear requirements. Many of the issues discussed affecting schema design and author usability will have a direct impact on specific requirements for applications.

Editor selection

A critical issue in editor selection will be whether authors are to create content directly in XML or, whether content will be created in another environment and converted to XML. If authors are to work natively in XML, it is necessary to determine exactly how authors will operate with the selected schema and how authors will be trained and supported.

It is suggested that all out-of-the-box XML editors require extensive customisation if acceptable usability levels are to be achieved. Schema designs vary considerably. No XML editor can provide a universal approach that works for all schema. Different content demands different approaches to the way authors will work with that content and the schema.

Planning for the XML editor must include a strategy to tailor the editor to the chosen schema and author needs so as to simplify the work of content authors. While this may involve an additional layer of development or expense, it ought to enhance the business case by minimising training effort, reducing ongoing support costs and improving overall author productivity.

The costs of failing to pay proper attention to author convenience are never factored into the business case and represent a time bomb that may only detonate after the system is exposed to authors. By then it is usually too late to properly fix the problem.

Rendering application selection

Rendering application development is directly affected by schema complexity. Simplification of the schema and elimination of unnecessarily loose content models will reduce the cost of application development.

When planning rendering applications it is necessary to develop functional requirements and formal output specifications to ensure that rendering applications are comprehensive and reliable from the outset. Formal specifications will greatly facilitate ongoing application maintenance.

CMS selection

Enterprise CMS selection involves a clear understanding of the application requirements and system architecture. Some of the critical issues that need to be considered include:

- integration with existing document management, web and intranet publishing systems, if any;
- document storage, retrieval and version management;
- records management and archive requirements;
- client and project information storage and relationships with documents;
- user rights and privileges;
- work flow support;
- integration with XML authoring applications and rendering applications;
- the level of granularity of storage of XML documents;
- web publishing support, including automated link creation from XML markup and link validation;
- metadata management and thesaurus or taxonomy support;
- searching and file format support;

- collaborative editing support, particularly involving persons outside the enterprise; and
- technical performance, scalability and security issues.

Extensive single source publishing functionality can be created without an enterprise CMS. Careful design of the schema and selection of suitable applications can greatly minimise reliance on a complex, expensive enterprise CMS. Where an enterprise CMS is needed, planning for it will benefit from a user and content driven approach to requirements development.

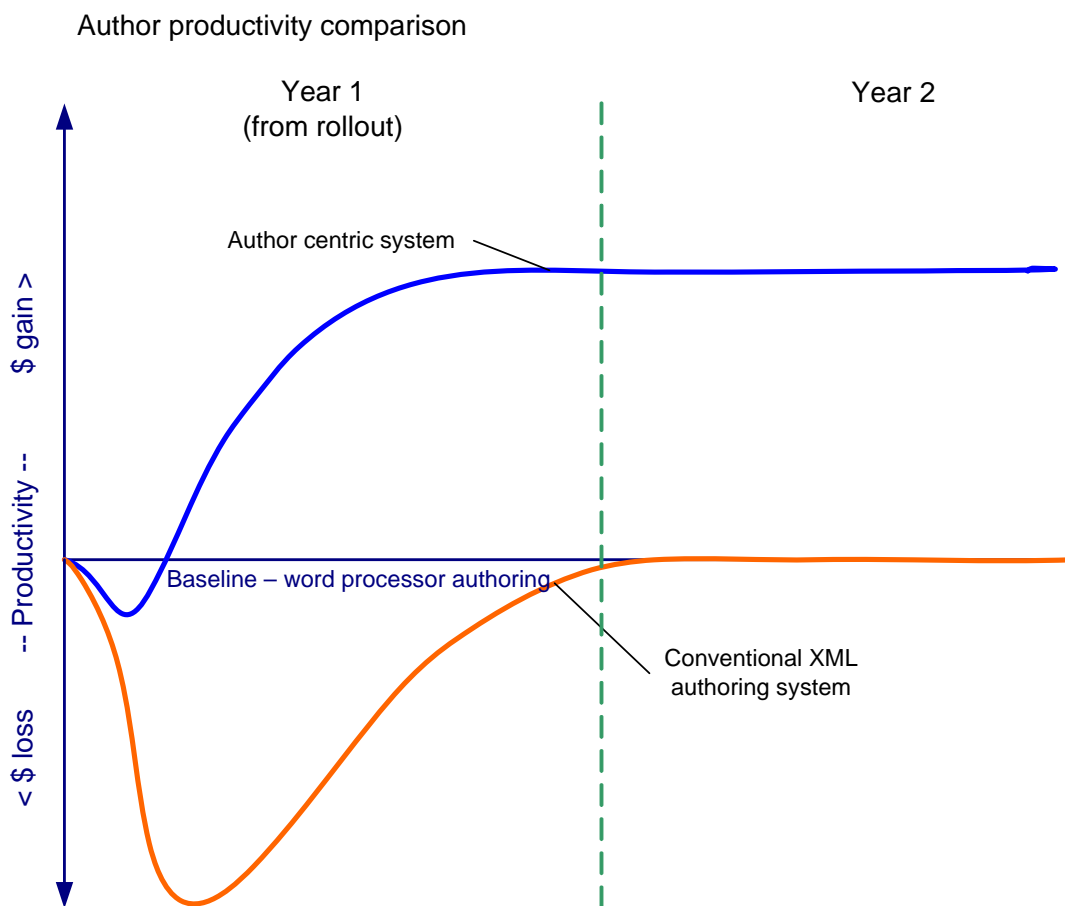
Develop applications

Clearly, application development and configuration is a major part of the project. If the project has been carefully planned, requirements fully developed and the schema carefully designed to model the data, application development should proceed efficiently. If the preparatory steps are incomplete, application development will be severely disrupted. Beyond that observation, application development is not further considered in this paper.

Rollout

One of the objectives of careful planning and attention to user needs in the planning and design stages is that application roll out problems will be greatly reduced. Inadequate attention to author needs has been one of the major failings of many single source publishing projects.

The potential difference between a project that places high importance on author needs compared to one that does not is shown in the following figure.



In this figure, the blue line shows the relative productivity of authors using an author centric XML editing system. The orange line shows that of authors using a conventional XML authoring system.

The key assumptions behind this figure are:

- The site has upwards of around 10 to 15 authors.
- In the author centric system, the schema is designed for author convenience, strictly limiting the options available to authors and the concepts they need to understand.
- In the author centric system, the XML editor interface has been carefully tailored to allow most, if not all, authoring tasks without the need to show XML tags.
- In the conventional system, the application is based on a loose schema with many elements. The XML editor is essentially 'out-of-the-box' and has undergone little, if any tailoring to provide a tags off interface for authors.

In the figure, the region between the two curves is the cost of not taking an author centric approach to the system planning and design. The quantification of this cost will

depend on many factors in each case. The costs of a conventional XML authoring system are quite high and require a lengthy period to bring authors up to even the same level as a word processor based system.