

WHITE PAPER:

The Ten Commandments of Content Management

By Jack Danaher, Vasont Systems

This white paper explains the ten most important principles of content management.



Vasont Systems
Phone: 717-764-9720
Email: info@vasont.com

www.vasont.com

The Ten Commandments of Content Management

- I. Content must be stored only once.
- II. Content must be stored separately from tagging.
- III. Granularity of content must be available at any level.
- IV. Use metadata to enhance content management capability.
- V. Loading, editing, and extracting must be independent operations.
- VI. Objects must have power.
- VII. Think enterprise solution, not just specific content.
- VIII. Content views must support the users' needs.
- IX. Your current DTD is the start, not the end, of your content management.
- X. Your content management system must promote reusability and extensibility.

I. CONTENT MUST BE STORED ONLY ONCE.

Points to look for:

- ▼ When new content is added, is existing content checked for its prior existence and reused if appropriate?
- ▼ When content is changed, is the changed content compared against existing content for possible reuse?
- ▼ Can the system immediately tell you and act upon all the current uses of a given content component?
- ▼ Can the system show the content components independently of any specific use of that content?
- ▼ Can the system easily allow the same content to be used in many different views, hierarchies, and configurations without ever duplicating the content?

Eliminating duplicate content in the system through single sourcing supports four related ideas:

1. Keeping track of all of the uses of a given piece of content
2. Separating the content itself from its uses
3. Accurately reflecting and tracking change when content is modified
4. Minimizing the content that must be maintained to satisfy the content's audiences

The bottom line: Storing components of content only once allows for the most effective and efficient system.

II. CONTENT MUST BE STORED SEPARATELY FROM TAGGING.

Points to look for:

- ▼ When a new tagging scheme emerges, can publications automatically use the new standards with existing content, or only with new content?
- ▼ Does each new standards announcement require the installation of yet another new release of the product?

There are many standard tagging schemes (i.e., XML, HTML) and other proprietary tagging schemes (i.e., Quark and Microsoft Word formats). If content isn't separate from the tagging, costly and time consuming data conversions are required to transform between these standards. But if the design has already separated the tagging from the content, then no conversion of content is necessary—simply another definition of which tags to use with the content in a given instance of content use.

Hence, if the content is separate from the tagging, the content can be reused with multiple tagging schemes as appropriate. For example, an SGML scheme, an HTML scheme, and a variant XML scheme can all be independently applied to the same content. In addition, any future tagging requirements can be implemented without any effect on the content itself.

This commandment also supports the previous principle of storing content only once, since a tag should also be stored only once and applied to content only as appropriate. Imagine a technical dictionary with 20 volumes and 11,000 articles. Each article has on average 50 paragraphs for a total of 550,000 paragraph tags. The tag <p> should be stored only once in the system. If you want the tag to be <par> instead of <p>, you are only changing two characters but leveraging the 550,000 instances. If there are three similar views for this content, a print publication view, a CD-ROM publication view, and a Web publication view, you are leveraging over 1.6 million instances in a split second.

III. GRANULARITY OF CONTENT MUST BE AVAILABLE AT ANY LEVEL.

Points to look for:

- ▼ The system should not just support the literal DTD, but should allow other component views of the content that are appropriate for the task at hand whether loading, editing, or extracting content.
- ▼ The DTD should help initialize your database system setup but not be the setup.

When content is broken down into components, the identification of the appropriate components depends on the requirements and uses of the information. For example, if a technical manual has warnings that can be used in other technical manuals, then the warnings should be identified as a component of the manual. Or perhaps, from a workflow point of view, some editors are specifically responsible for the content of the warnings. This is another reason to consider making the warning a component of the manual. Applying fine granularity to the content makes content more flexible. On the other hand, consider a complaint letter that is being stored in a database for text searching purposes only. There is no reason to break the letter into components, so we would be applying coarse granularity to the letter.

Any content scheme that is not totally flexible in the assignment of this granularity will inevitably be inappropriate for certain situations.

At times it is appropriate for the granularity to be coarser than the elements defined in a DTD. For example, consider a paragraph in an encyclopedia. There may be no value in defining components identified as italic within the paragraph. The editing view of the content hierarchy would be too deep and complex. The decision could add millions of components to the database that have little value, hence hurting efficiency from a human interface and a processing point of view.

IV. USE METADATA TO ENHANCE CONTENT MANAGEMENT CAPABILITY.

Points to look for:

- ▼ Does the system allow for adding metadata retroactively?
- ▼ Does the system easily identify components that are missing required metadata, even if the requirement was an afterthought?
- ▼ Does metadata for a given component include:
 - ▼ Its group of related components?
 - ▼ Its child and parent components?
 - ▼ Component uses in load, extract, and edit views?
 - ▼ Rules of behavior in different situations?
 - ▼ Database loading instructions?
 - ▼ Database editing instructions?
 - ▼ Database extraction instructions?
 - ▼ Component uses in content instances?
 - ▼ Component references in content instances?
 - ▼ Data type of the component: text, multimedia, pointer or special?
 - ▼ Component access rights for the users?
 - ▼ Component edit rights for the users?
 - ▼ Global affectivity of component changes?

Metadata is key to having control and power over existing content. A content management system should allow for the extension of information about existing content. These extensions can be in the form of attributes and related components. This metadata can be used for filtering, searching, and organizing the content for appropriate uses. The metadata does not have to be extracted from the database when finished products are created. As an example, consider newspaper articles. Having the ability to add attributes that classify and describe the articles enables the reassembly of articles as appropriate when needed.

Metadata attributes and components of the articles could be:

Scope: World, Asia, US, etc.

Subject: Economy, politics, science, crime, war

Time: Date

Main Point: Facts, warning, humor, criticism

Main Topics: US Congress, US Presidency, General Motors, prisons

Key References: JFK, FDR, Justice Department

A search could ask:

"Find me articles of US scope that

- ▼ address crime in the last month that
- ▼ contain facts about prisons and
- ▼ refer to the US congress or the Justice Department."

Without metadata, a pure text search might pull up too much or too little content. Metadata promotes productivity and accuracy.

V. LOADING, EDITING, AND EXTRACTING MUST BE INDEPENDENT OPERATIONS.

Points to look for:

- ▼ Can I load content components in multiple ways?
- ▼ Can I view and edit the same content in different combinations?
- ▼ Can I extract the various edit views in multiple ways?
- ▼ Can my load view be different from my edit view, which is different from my extract view?

Your business needs may require more than one way to load your content into the database. The same content may be included and edited in different views (publications). Multiple extracts of the components may be required that use different combinations of components, attributes and hierarchies. If I had to load, edit and extract the content in the same way for each operation, I would have a very limited system that would not take full advantage of the potential reuses of the content.

VI. OBJECTS MUST HAVE POWER.

Points to look for:

- ▼ Does changing the DTD make you cringe, or do you view it as an easily attainable opportunity to improve your content?
- ▼ Do spin-off publications make you nervous or excited at the prospect of easily attained additional revenue?
- ▼ Do you have to add staff to manage your new print publications, CD-ROM releases and Web offerings, or do you have the right productivity tools to handle the onslaught?

It is important to distinguish between a tool that supports object technology and the objects themselves. Tools like object databases help to build objects but do not insure good objects any more than a hammer insures a good architectural blueprint for building a house. The real proof of object technology is not what tool was used, but what capabilities the objects have to provide solutions and productivity. In the area of content management, the properties and methods of the component objects can be quite extensive. For example, object methods can include reordering, reusing, cloning, transforming, recombining, reanalyzing for behavior, extracting, loading, editing, adding, deleting, assigning attributes, and so forth. The properties of component objects relate to the metadata that is associated with them. (See item IV.)

VII. THINK ENTERPRISE SOLUTION, NOT JUST SPECIFIC CONTENT.

Points to look for:

- ▼ Do you have the ability to mix and match components from different groupings of content to create new content models?
- ▼ Does your database support multiple types of content?
- ▼ Does your database seamlessly support multiple languages?
- ▼ Does your database support a multiple site topology?
- ▼ Does your database support site management and sharing of your corporate content resources?
- ▼ Can the database support hundreds or thousands of users and terabyte size amounts of data?

When content is initially loaded into a content management system, it is usually associated with a specific publishable group of content such as a parts manual, a dictionary, or a user's guide. Furthermore, it will usually be entered in a base language such as English. But the full breadth of content in an enterprise is obviously much broader than this. The language needs will likely expand as your business enters the global marketplace.

A content management system should be able to group sets of related components such as those required for the parts manual and the user's guide. Additionally, the multiple languages should be maintained seamlessly within the database architecture. Components should be reusable across different groupings, if appropriate, and new groupings of components should be available for special applications, such as a Web query application.

VIII. CONTENT VIEWS MUST SUPPORT THE USERS' NEEDS.

Points to look for:

- ▼ Do all users edit off the same view of the content?
- ▼ Are you able to create Web applications with your content?
- ▼ Are users seeing more of the content than their job requires?
- ▼ Can users work with their favorite editing tools?
- ▼ Does your content management system integrate workflow with content editing tasks?
- ▼ Am I performing duplicate content management for each separate product derived from my content?

When loading, editing, or extracting content, one view of content may not be an optimal solution. Consider a dictionary of the English language. If you are an etymologist, you may only be interested in the word component and the etymology component, and not the definitions and pronunciations. Likewise, if you are researching dictionary terms on the Web, you would need a much different view of the content. The underlying content structure might require XML, HTML, or even WML.

An editing view can be constructed to support the user's specific need, which can be coordinated with the overall view of the full word and its components as seen in the printed dictionary or on the Web. Several extract views may also be appropriate. One extract may contain etymologies and long definitions. An extract for the pocket edition may not contain etymologies and only short definitions.

In these examples, workflow controls could assist in defining and presenting the required editing tasks. This workflow should be integrated in the content management system.

IX. YOUR CURRENT DTD IS THE START, NOT THE END, OF YOUR CONTENT MANAGEMENT.

Points to look for:

- ▼ Can you easily change content rules and behavior retroactively?
- ▼ Can you easily add or remove components and attributes from your content views, loads, and extracts?
- ▼ Can you easily create new content products as variations of your existing content products without extensive editing?

A content management system should allow you to easily evolve your content as time passes and new ideas and needs arise. You may want to add new components, change the behavior of existing components, change hierarchy relationships, order the content in a new way, reassign attribute values, etc. The system should be flexible to accommodate these new needs.

X. YOUR CONTENT MANAGEMENT SYSTEM MUST PROMOTE REUSABILITY AND EXTENSIBILITY.

Points to look for:

- ▼ Does your system's extensibility include:
 - ▼ API?
 - ▼ Reusable processing options?
 - ▼ Enforceable behavior?
 - ▼ Types of data?
 - ▼ Types of data access that can be enabled or restricted?
 - ▼ Editable views?
 - ▼ Components?
 - ▼ Extracts?
 - ▼ Loads?
 - ▼ Users?
 - ▼ Groups of related components?
 - ▼ Helper applications such as XML editors, graphic editors, word processors, sound editors, etc?

Your future requirements are unpredictable. Without reusability, you would be solving the same basic problems repeatedly for each new set of content. Without extensibility, today's perfect fit becomes tomorrow's organizational problem. A content management system that promotes reusability and extensibility is a future-proof system.

Headquartered in Pennsylvania, Vasont Systems is a provider of content management software and data services, backed by more than 55 years of experience in the information management and publishing industry. Its Vasont content management system enables organizations to manage and store their multilingual content once for multi-channel delivery. Fortune 1000 companies and organizations from industries including manufacturing, technology, publishing, government, and healthcare use Vasont to manage and produce multilingual technical documentation, product manuals, and reference materials.