



In this Issue

Editorial

Page 2

It's all about reusing content

Feature Article

Page 3

Designing content reuse: The "tract housing" model

Best Practices

Page 5

Maximizing content reuse, reuse, reuse

Information Architecture

Page 7

Constructing with content

Tools and Technology

Page 10

Support for reuse

People, Processes, and Change

Page 13

Change management issues in implementing a reuse strategy

Tools and Technology

Page 15

The marriage between dynamic and static web content

Case Study

Page 8

A case study in modular documentation

In the news

Page 21

Contributors

Page 23

Feature Article

Designing content reuse: The "tract housing" model

Once you're ready to implement a content management strategy in your organization, it's useful to look at other industries that rely heavily on standards. This article compares the development of a content repository with the development of tract housing to identify best practices for content reuse, focusing on planning, identifying reuse, and incorporating new content.

Read more on page 3 ...

Best Practices

Maximizing content reuse, reuse, reuse

This article defines content reuse, reviews the types of content reuse, and outlines how organizations can best leverage content reuse when working with structured content in a single-source component-level content management system.

Read more on page 5 ...

Tools and Technology

The marriage between dynamic and static web content

Many organizations have independent designs and systems for their dynamic and static web content. But the trend right now is to find a way to give dynamic sites more flexibility, rounding out their transactional capabilities with a more content rich user experience that can be found on static html sites. In this article, Ted Spencer, a web content management consultant, describes how to adapt traditional transactional web site designs to deliver html site flexibility, and facilitate reuse across static and dynamic pages.

Read more on page 15 ...

Case Study

A case study in modular documentation

A group of Unisys technical writers (located in Mission Viejo, CA; Roseville, MN; and Malvern, PA) recently moved to a more modular approach to creating manuals. The more modular approach to documentation was precipitated by the company strategy to use common server technology across multiple product lines. In this article, Debbie Donahue, the project manager who led the writers in the Information Development group through this change, describes their approach to and rationale for modular writing.

Read more on page 18 ...

It's all about reusing content

Reuse, reuse, reuse, reuse... content reuse is key to content management so it bears repeating repeating repeating. In fact, the desire to reuse their content is one of the main reasons organizations decide to move to a content management strategy. Content reuse means that you can write content once and use it wherever required, but it also means that you have to write content consistently so that it can be reused. This issue of *The Rockley Report* explores various aspects of content reuse including suggestions for maximizing content reuse, preparing content for reuse, and implementing content reuse. Suzanne Escoffier from Symantec opens this issue by showing us how a content reuse strategy can follow a tract housing model and Suzanne Mescan from Vasont Systems outlines how organizations can best leverage content reuse when working with structured content in a single-source component-level content management system. Ted Spencer, a Web Content Management Consultant, describes how to adapt traditional transactional web site designs to deliver html site flexibility and facilitate reuse across static and dynamic pages. Debbie Donahue describes how her team at Unisys implemented a modular writing approach and Steve Manning further describes some of the change management issues you need to consider when implementing a reuse strategy. Our Information Architecture column compares content reuse to building with Lego (TM) and as always, Scott Abel rounds out the issue with a look at some resources where you can learn more about content reuse.

We welcome your feedback. Please send comments, as well as suggestions for stories in future issues to kostur@rockley.com. Our Call for Submissions describes the kind of stories we're looking for and how you can submit articles for publication in future issues.

THE ROCKLEY REPORT

Editor

Pamela Kostur
The Rockley Group
moreinfo@rockley.com

Reprints and Reuse

The Rockley Group grants permission to educators and academic libraries to photocopy articles from this publication for classroom purposes. There is no charge to educators and academic libraries, provided they give credit to *The Rockley Report* and The Rockley Group. Others must request reprint permission from The Rockley Group at moreinfo@rockley.com.

Copyright

The Rockley Group holds copyright on all material published in *The Rockley Report*, but grants republication rights upon request.

Contact Information

The Rockley Group Inc.
166 Main Street, PO Box 163
Schomberg, ON L0G 1T0
Canada
Phone: 905-939-9298
Fax: 905-939-9299
Email address: moreinfo@rockley.com
Corporate web site: www.rockley.com

Feature Article

Designing content reuse: The “tract housing” model

Suzanne Escoffier
Symantec Corporation
sescoffier@symantec.com

Once you're ready to implement a content management strategy in your organization, it's useful to look at other industries that rely heavily on standards. This article compares the development of a content repository with the development of tract housing to identify best practices for content reuse, focusing on planning, identifying reuse, and incorporating new content.

Introduction

Symantec is the world leader in providing solutions to help individuals and enterprises assure the security, availability, and integrity of their information. Consumers can purchase Symantec software through retail outlets, the Internet, as part of a subscription package with a service provider, or as part of a new computer purchase. They can choose from a variety of product offerings that range from bare-bones virus protection to comprehensive solutions that include protection against intrusion, virus, spyware, spam, and identity theft.

To support Symantec's family Consumer products, we have developed a matrix of content that is as flexible as our product offerings. Over the last year, we've migrated that content from individual documents to a content repository. In the process, we've looked beyond the information development community for models of mass-production, personalization, and rapid delivery.

One unexpected source of inspiration came as a result of hours spent commuting through Los Angeles and Orange counties. Following the same route at relatively slow speeds each day provides plenty of time to watch the development of new tract housing projects. During construction, each new development looks the same but when completed, each has a unique look and feel. The philosophy underlying the development of new tract housing looked like it could inform our development of reusable content for the variety of product offerings we support.

Don't shortchange planning

What's clear from the start is that a tremendous amount of planning goes into each new tract housing development before the first bulldozer appears. The land is mapped, divided into plots, and the basic unit of construction determined — home, townhouse,

apartment, or condominium depending on the market to be attracted. Standards are set for architectural style and density, often in cooperation with local government. Standards for plumbing, electrical, insulation, framing, and roofing are adopted from rigid codes and guidelines set by states or industry organizations. Personalization and customization are applied over this standard core to create engaging, relevant housing geared to a specific market. And each new owner adds their own unique touches when they move into their new homes.

Before assembling a content repository, we engaged in a year of planning and pilot projects. We organized and reorganized our content, searching for ways to get the most reuse. We discovered that, regardless of how we deliver content, our basic unit of construction is the section. We codified our guidelines for the construction of sections in our company's version of the DocBook DTD. We are as rigid as the local building inspectors when it comes to building structured content with the result that we can mix and match sections as needed when creating manuals, HTML help, and cards.

Identify opportunities for reuse

Having a structure in place has freed us to focus on the content and its reuse. Using the section as our basic unit of construction accommodates three types of reuse: exact reuse, substitution reuse, and conditional reuse.

Exact reuse

Exact reuse forms the framing, insulation and roof of our reuse strategy. Some sections can be reused exactly as created. Exact reuse is most successful with content that is fundamental to our business. It provides a consistent framework on which to build the product-specific content. Exact reuse generates increased consistency and a clearer message for our

Feature Article

users, especially if they own more than one Symantec product. Exact reuse guarantees that a feature or task is given the same treatment, regardless of the product it appears in. Examples of exact reuse include sections containing software update or virus removal procedures. An added benefit of exact reuse is that it removes all duplicate text from the translation process.

Substitution reuse

Substitution reuse adds an initial layer of personalization to our content, similar to the choice of carpet color or floor finish in a tract home. In our content, personalization occurs when feature and product names change. By placing variables (entities) in content instead of specific product or feature names, we can specify product or feature names to substitute in their place each time the content is extracted. Substitution reuse guarantees that the remainder of the content stays the same when product or feature names change, and that those changes can occur as often as needed. By changing a minimum amount of information, substitution reuse supports those customers who move up from basic to more advanced products. An added benefit of substitution reuse is that only the variable text needs to be translated after the initial content translation.

Conditional reuse

Conditional reuse maximizes customization while retaining the benefits of reuse. These are the floor plan and upgrade decisions that make the tract house into your home. In our content, customization occurs when features or configurations change to support different markets and product lines. For example, some products include a CD that is used during installation and emergency tasks. If a CD is not available, the product provides other means to complete these tasks. To simplify the user's experience, we don't provide the CD information with versions of the product that don't include a CD.

By identifying text that is associated with a specific product configuration, portions of the text can be reused while still preserving the section as the base unit of construction. In our content repository, we store paragraphs for each supported configuration together, setting an attribute to identify each. When we reuse the section, the extract checks the attribute of each element to determine if it should be included. The reused text maintains consistency across configurations and limits the amount of rewriting necessary to support new configurations.

Incorporate new content

As developers devise new model homes, they adhere to standards to guide their design. By adhering to standards when designing new homes, developers can respond to changing marketplace demands without sacrificing the gains in production and efficiency that are already in place. As we create the documentation set for each new product release, we also respond to demands to add content about new and enhanced features. Standardization helps us quickly create consistent, reusable content that is easy to localize.

We adhere to our content model guidelines to avoid reinventing the structure each time we add content. Our content models incorporate both guidance on what content to include and how to structure that content. Content models have also made it easier to identify content that is not unique, but that can be added to an existing structure and reused conditionally.

We adhere to editorial guidelines to avoid introducing new language into the repository and to guarantee that terms are used in a consistent manner. We maintain a master list of index entries and put new entries through a review and approval process. We do the same for any new terms added to our master glossary.

Finally, we work closely with the product marketing and development teams to improve the consistency of the product UI. With our content structured and in a single repository, it is much easier to identify and resolve UI issues such as terminology and casing so that we do not needlessly create derivative content just to adjust for the capitalization of a product feature or button name. And while it feels good to streamline the content repository, the real winners are the users of our products and documentation.

Summary

A long planning cycle helped us prepare for our transition to a content repository. Looking at best practices both within the information development community and in other industries, such as tract housing, helped us identify process checkpoints as well as strategies to improve the structure of our content. Using these strategies, we have seen immediate improvements in reuse. We will continue to look at work done in our field and others as we develop best practices to help us shorten the time between content development and delivery.

Best Practices

Maximizing content reuse, reuse, reuse

Suzanne Mescan
Vasont Systems
mescan@vasont.com

This article defines content reuse, reviews the types of content reuse, and outlines how organizations can best leverage content reuse when working with structured content in a single-source component-level content management system.

Content management and reuse – the two go hand in hand. It is one of the key reasons for purchasing a content management system. The most effective implementations of single source content management systems lower translation costs, improve time-to-market, and maximize the production and management of content by optimizing content reuse. A system where content reuse is optimized makes the most efficient use of all available content. Whether a department is in the process of implementing a content management system or has been working with one for several years, achieving a high level of content reuse is the ultimate goal.

What is Content Reuse?

Single-source content management systems allow organizations to save useful “chunks” of content a single time in a central repository and then reuse those chunks as often and in as many places as needed. This allows organizations to use the same information over and over – i.e., a product definition in printed manuals, on customer service Web sites, and in CD-ROM instruction manuals - while still only saving it in the repository one time. The ability to work with content at a granular level and manage where and how it is used is much more powerful than working with content at the document level, where reuse is minimized, multi-channel publishing is complex, and tracking the content itself is difficult at best.

Types of Content Reuse

Depending on your system’s capabilities, the content in your repository can be reused in two basic ways - Component reuse and Pointer reuse.

Component reuse is the most common form of reuse. The product definition mentioned above is a perfect example. The definition of the product is a chunk of content, stored a single time in the repository, and

then reused many times within a document or across many different documents.

Pointer reuse is different. Sometimes instead of reusing a component, it makes more sense to use a pointer to reference that component. For example, imagine the editors of the technical documentation department decided that they do not want their authors to have access to change the product definition. If the authors need to use the definition in a manual, they “point” to it. The product definition will be in the final manual, but the authors have no editorial control over it. Pointers are often used for legal statements or product warnings – content that must stay strictly consistent and should never be changed because of legal issues or compliance with government regulations. In a robust single-source content management system, pointers are also only saved a single time in the repository.

How to Leverage Content Reuse

Global change and modular (or pointer) change provide two ways to take the best advantage of content reuse. Global change applies to component reuse. Imagine that our product definition is reused in 25 places throughout the organization – in users’ manuals, marketing materials, packaging, and help documents. The system administrator gives someone within the department global change authority over that particular piece of content. If the definition needs to change, he or she is authorized to make the change. Remember that the definition is only saved once in the repository, regardless of how many times it is reused. Once a change is made, the system will give that person a choice – do they want to automatically (or globally) update all 25 instances of the definition? Do they want to only update one specific instance and ignore the other 24 instances? Or do they want to pick and choose, among the 25, which they would like to update? Whatever decision he or she makes, the changes made will be recorded in the global change log. The log tracks who made what change, when,

Best Practices

and to what. In robust systems, all changes can always be reversed. The system quickly alerts the user when changes have taken place to the log, when something is pending, and if something has been approved.

Global change is the key to taking the best advantage of component level reuse. First and foremost, it eliminates time spent and mistakes made when updating individual instances of content. If we tried to update each of the 25 instances of the product definition individually, it would take a long time to search through all of the documents to find every occurrence of the product definition, and there would be a high risk of potentially missing some occurrences. Global change eliminates the time and the risk. The global change log gives users control over all global changes, so that they know when the product definition was last updated and can revert to that form if needed (globally, or individually as needed). Global change truly enables content to be stored and managed for multiple uses.

Modular change, or pointer change, is very similar to global change, only it deals with references instead of components. Imagine that 25 different documents reference that same product definition. What if that product is discontinued? Each pointer will have to be redirected somewhere else. The authorized user has the option to redirect all 25 pointers at once (in batch), pick and choose which pointers should be updated, or delete all the pointers. Each change is logged, just like in global change, so users will always be able to look back and see what pointers used to reference and where links need to be fixed. The log is important because it helps users to understand how a change will affect other content within the system.

Modular change is important for maximizing reuse because it allows reuse without editorial privilege. Content remains consistent because editorial access to it is limited, but reuse is still possible. This form of reuse is particularly useful for graphical elements – allowing authors to easily place graphics without having editorial control over them. Pointers also allow for template-like workflows, establishing a large amount of control over content and how the user's time is spent. The log minimizes mistakes, making broken references immediately obvious, so that content can be reused with maximum efficiency.

Conclusion

Organizations that master global change and modular change take the best advantage of their content reuse and see dramatic benefits in production and time-to-market.

Constructing with content

Pamela Kostur
The Rockley Group
moreinfo@rockley.com

Content reuse is key to content management. Content reuse means that you can write content once and use it wherever required, but it also means that you have to write content consistently so that it can be reused. This article describes the concept of “constructing with content”, comparing it to constructing with Lego™, and describes how to prepare your content for reuse.

Content as Lego™

I like to think of content as Lego building blocks – those colourful plastic blocks you played with as a child (or as an adult!), building castles, forts, and other architectural wonders. Lego blocks come in various sizes, each designed for a particular function, and with colours that indicate their function. You select the appropriate pieces, then build your creation. If you use a piece in the wrong place, it will not fit with the other pieces and your masterpiece could fall apart.

Constructing information products follows a similar process. In a content reuse situation, information products are constructed from elements that are designed and written to a standard that supports their various purposes. If the elements do not conform to the purpose for which they are intended, your construction may fall apart, the same way your Lego castle could fall apart if you use roof pieces to construct the walls! Also, the roof pieces must conform to a “roof standard” and the wall pieces must conform to a “wall standard”, ensuring that the roofs and walls will be solid and that the pieces will fit together.

Designing content building blocks

Likewise, content must be always designed for the ways in which it is used, especially when content components are reused to create other information products. So, what do you have to do to prepare your content so that it is reusable? Just like with Lego building blocks, you have to make sure that the pieces fit together and that they are easily identifiable so their purpose is clear to those who use them. To ensure your content pieces will work together, you need to create a standard for each component included in your “box of content pieces”. Think of the box of Lego as your CMS, or database. Think of the Lego pieces as your content components.

So, where to start? When you’re constructing with content, the first thing you have to do is to figure out where the content will be used – where, in your information product construction must that piece fit. This is what you’re doing when you create content models; you’re designing the structure of the “building”. Then, you have to figure out the structure of each of the pieces in the “building”, ensuring they will fit together, even if used in a different building. For example, when building with Lego, you can make the item identified in a Lego kit, or you can use the Lego pieces to make other items. Whatever you decide to build, the pieces have to work together.

It’s important to define the structure of each of the pieces because unless they are written according to a standard, the integrity of your information products in which the pieces are used may be compromised. From a content modeling perspective, structure defines *the hierarchical order in which information occurs*, but from a writing perspective, structure defines *the way the content within each hierarchical element is written*. The content model documents the structure of information products, which is then implemented in a DTD/authoring template that helps to enforce it. (Similarly, the Lego set may come with a design showing you how to put a particular item together and while not enforced, it provides a guideline – a set of “rules”.) However, structure goes beyond defining how an information product is put together. For information products to be truly unified, content must be structured and written the same way, so it “works” wherever it is used, just like the Lego pieces have to be unified so they work regardless of what you decide to build with them.

For example, the model for a procedure will tell you the elements that make up a procedure (such as overview, steps, results, cautions), which of those elements are mandatory and which are optional, the order in which they appear, and which elements are reused elsewhere. However, the model doesn’t tell

Information Architecture

you how those elements must be written. This is where structured writing comes in. Structured writing provides the standards for how to structure and write the elements of content in your information products and is necessary to prepare your content for reuse.

Why is structure so important?

When implementing a unified content strategy, it's critical that authors structure and write their content consistently. Well-structured content leads to more opportunities for reuse across product lines, audiences, and information products. In a structured-writing environment, authors follow the same structure rules or guidelines for each element of content, ensuring its potential reuse.

Many problems can arise when content is not structured to support its various uses and users. Not only is unstructured writing difficult for users to follow, it's also difficult for authors to create. For example, without structured writing guidelines for procedures, some authors may include results within their steps, while others don't. And, even if they do include the result portion of the step, they may include different information within it, or use different grammatical structures than other authors. If steps are to be reusable across information products, they must be structured and written the same, so their reuse is transparent to both authors and to users. Imagine trying to build a Lego construction with pieces that *look* like they should go together, but just don't fit. They appear to be wall pieces, but the wall pieces just don't work together. You end up trying to cram them into place, most certainly compromising the integrity of the structure.

Structuring and writing content consistently not only makes it possible for you to reuse content transparently; it also enhances information products' usability. Implementing a unified content strategy is an ideal time to examine your content for usability, to create usability criteria that define what makes content usable for each of its intended audiences, and to include usability criteria in your writing guidelines.

Simply reusing content can facilitate usability (by reusing content, it is the same wherever it appears), but if that content is poorly-written or is open to interpretation, it is not usable, regardless of how well it conforms to the structure or how frequently it is reused. If you reuse unusable content, it's certainly

the same wherever it appears, but it's also *unusable* everywhere it appears. In addition to defining consistent structures for your content you must also examine the content itself to ensure it is accurate, readable, and not open to interpretation. Then, you can decide how to structure and write it to enhance usability.

Implementing structured writing

Implementing structured writing as part of a unified content strategy has three main steps:

1. **Analyze** - Define all information requirements up front, determine what information products do you need to create, for whom, for what purpose (decide which kinds of constructions your Lego set will support, e.g., houses or vehicles).
2. **Design** - Create content models to accommodate your information requirements, indicating where content can be reused (figure out where your Lego pieces can be used to create other constructions, e.g., can the house and vehicle sets share pieces).
3. **Write** - Create structured writing guidelines. Determine how to structure and write the content for each element in your models (determine how the Lego pieces must be constructed to support their use in different constructions, e.g., how must the reusable pieces be constructed so they work in both the house and vehicle sets).

Some tips for creating structured writing guidelines are to:

- **Separate content from format**
Because the pieces of content may be used in different places, think about what the content must say and how that content must be written instead of how it will appear on the page. Lego designers don't always know how their pieces will be used, so they must focus on how that piece must function to work wherever users decide to use it.
- **Structure similar kinds of components in the same way**

So reuse is transparent, design standards for similar types of content. For example, design a standard for steps so that steps are always structured and written the same way. Do they contain a result or not? How is that result reflected in the step? Once you've defined the standard for a step, you can use that standard for all steps, in all information products. Think of what information

Information Architecture

types you have (e.g., concepts, processes, procedures) and design standards that accommodate that particular information type.

- Refer to information theory

Allow information theory to inform the standards for your content. Look to information theory for guidance in such things as chunking (how big should a chunk be), labeling (which chunks are given labels and how are those labels written), relevance (what constitutes relevant content for each chunk), and above all, usability (what usability criteria apply to each content component).

Summary

In a content management environment, you decide what the structure of your information products should be, document the structure in models, implement the models in authoring templates, create writing guidelines for each element, then manage the elements in a definitive source where they are accessible to all those who need to “build” information products. Just like Lego pieces are designed to support reuse in building Lego constructions, content must be designed to support reuse in constructing information products. When you’re constructing with content, you need to think about how each of the components is designed (both the structure and the content within the structure), ensuring the integrity of your construction.

Tools and Technology

Support for reuse

Ann Rockley
The Rockley Group
rockley@rockley.com

Knowing how your system supports reuse guides you in the development of your authoring reuse processes and in the way you design your content for reuse. This article looks at the features of reuse support to look for in your system.

How technology supports content reuse is a major factor in the success of your project. Technological support of reuse impacts:

- How authors reuse content
- Your information architecture of reuse

Knowing how your system supports reuse guides you in the development of your authoring reuse processes and in the way you design your content for reuse.

This article looks at key areas of reuse support including:

- Nesting
- Granular reuse
- Reuse management

Nested reuse

Nested reuse can be one of the more problematic areas of support for reuse. Nested reuse occurs when there are an number of reusable elements nested (enclosed) within one element. Examples of nested reuse include:

- Content which is common to many information products, but there are slight variations for each instance of reuse. Variations could be due to product, geographic region, audience etc. You can write one element that includes the core content and each of the variations within the same element. When you publish the element, content not appropriate to the desired publication is filtered out.
- Content which is derived from a superset of content. For example, you could have quick reference information that is a subset of user information or content for a product label that is a subset of the product specification. In this case, it is easiest to write the superset content and identify which sub-elements are appropriate to the subset. When

the content is published, the subset information can be automatically extracted.

There are two main ways in which nested reuse is supported by technology:

- Metadata and filtering
- Master element

Metadata and filtering

Using metadata and filtering is by far the easiest method for nested reuse. This involves adding metadata to the content that varies or that you wish to extract. For example, you would add metadata that defines that certain sub-elements are only applicable to a specific region or audience. Or you would add metadata to a series of sub-elements to identify that it is appropriate to a subset content publication. Alternatively, if you are using a semantic structure for your structured content (e.g., Semantic DTD/Schema), certain elements can be automatically extracted (e.g., title and steps in a procedure).

This method enables writers to write in context and quickly identify the content that should be filtered or extracted. It is an easy way to keep all the content use and reuse in perspective and it is an easy way to identify all the outputs for the content.

Master element

This form of support for nested reuse is a little more complicated. There are some content management systems that don't support nested reuse using metadata and filtering. In that case you may need to build a "master" element and then selectively use the pieces you need for each instance of the content. For example:

You have a feature description that is 95% the same for six different variations of the product, 5% varies for each product. First you create the common feature

Tools and Technology

content, then you create individual elements for each of the six variations. Then you group the content together in a master element so that you can see the common content and each of the variations at-a-glance. However, you cannot use the master element to produce each of the variations, as the content management system does not support that type of reuse. Instead you select the common content then select the variant content to create an element specific to the variation. You repeat this until you have each of the six variations. On first glance this seems like more work; however, we have found that by keeping the common content and each of the variants together, it makes it easier to maintain a better understanding of the complete set of content. In addition, having the content in a master element makes it easier for reviewers to review and ensure that all variations are correct, rather than having to review each variant individually.

Granular reuse

You will need the ability to manage small granular elements of content where necessary. Some content management systems can manage elements of any size, others cannot.

Writing content in context

Typically you create content in context (e.g., a section). When you check the content into the content management system the content is “burst” apart into its element parts so that the elements can be reused as desired. However, some content management systems require that you create the smaller elements then assemble the larger element from the smaller elements. This is more awkward for authors, as authors don't write small isolated pieces of content then assemble them, they want to write a piece of information in context. Look for a content management system with a sophisticated bursting mechanism.

Managing small granular elements

For the most part you do not want to manage large volumes of small granular elements of content. The smaller the element, the more pressure there is on your content management system performance when you need to assemble a larger element and the more difficult it is for you to store and search for these reusable elements. However, there are times when small elements are necessary. Some content management systems make it very difficult to manage small reusable

elements. They don't provide an easy methodology for naming, storing or assembling these elements. Verify that your content management system can handle small reusable elements and determine how they are managed.

For more information on supporting granularity see *Issues with Granularity, Vol 1, Issue 2, The Rockley Report*

Reuse management

Good content management systems enable you to effectively manage your reuse by enabling you to:

- Identify where content is reused (Where Used report)
- Select from multiple update options
- Selectively change reused elements
- Notify you of reuse

Where Used report

Having the ability to run a “Where Used” report on your content is invaluable. A Where Used report identifies every instance of reuse of your content. This enables you to see at-a-glance where content is reused. You can run it for a particular reused element or to get an understanding of reuse in general in your entire content set. This report is particularly important in companies which use “opportunistic” reuse (reuse which is not planned in advance, rather reuse occurs when authors choose to reuse content). Many times authors don't know who is reusing their content. A Where Used report identifies where your content has been reused making it possible for you to understand the impact any change you make will have on the reuse instance. In addition, it enables you to understand how widespread your reuse is in the organization.

Multiple update options

When reusing content it is important to be able to identify how you want to update the content when the original source is updated. Look for the following options:

- Automatic Update
Automatic Update ensures that when the source element is updated, everywhere that element is reused is automatically updated. This is beneficial when you want to ensure that content is updated.

Tools and Technology

However, it can be problematic if the changes to the source are not appropriate when the element is reused in another instance. Use caution when using this option.

- Selective update

If you are using Automatic Update it is valuable to have a Selective Update option. This means that when you update the source you are given the opportunity to look at all the instance of reuse and determine which instance of reuse you want to update. In this way you can update all, some or none. When you don't update the reused element the unchanged element is automatically branched from the reused element, as it is now a derivative.

- Notified Update

Rather than automatically updating each instance of reuse you can notify the individual who has reused your element that a newer version of exists. At that time they can choose to accept the update or reject it. This is very useful because it gives the author who has reused the content the option of updating or not. In addition, this puts the control of the update in the reuse author's hands not the source authors. However, if there is a lot of reuse, reuse authors can be inundated with notifications. You may want to set the option such that the notification does not occur until they update their information product or access the reusable object. It is a good idea to ensure that reuse authors are notified of changes before they finalize their content to ensure they are aware of all potential updates.

- Severed Update

Severed Update is really an instance of copy and paste. When you choose this option you have decided that you do not want the reused object to be automatically updated or to be notified of an update. This will become a new element with no relationship to the previous element. Use this option with caution, as content can diverge significantly when you "copy" rather than reuse.

Notify you of reuse

When you own an element you determine if it is updated or deleted. The process for update is described above in Multiple update options. However, deletion is often not controlled through update options. To ensure that the deletion of an element does not cause problems for someone else you can run a Where Used report, but that can be an onerous task

if you have to perform this function each time you decide to delete an element. Some content management systems automatically notifies you when an element is being reused elsewhere and prevents the deletion. This ensures you don't accidentally delete content being reused. If you still want to delete the element you need to delete its reuse everywhere it occurs.

How technology supports content reuse is a major factor in the success of your project. Technological support of reuse impacts how authors reuse content and how you design your information architecture of reuse. Make sure you know how your technology supports reuse as you begin your content design process.

People, Processes, and Change

Change management issues in implementing a reuse strategy

Steve Manning
The Rockley Group
manning@rockley.com

The success of implementing a reuse strategy depends on many factors, not least of which is the effectiveness of your change management process. To be clear, in this context change management does not refer to the modifications to content, but instead how changes to the workplace are introduced to and rolled out to the writers.

There are many reasons why projects fail. Often, the issues are more human-related and less technology or project-related. They can include:

- Resistance to change
- Lack of a champion
- Lack of core competencies
- Lack of communication
- Failure to involve others

The success of implementing a reuse strategy depends on many factors, not least of which is the effectiveness of your change management process. Change can be difficult for people. A move from a traditional style of authoring, where authors have control to create and recreate information at will, to one with a focus on creating reusable content can be difficult. You will need an effective change management plan to help writers make the change.

The scope of change

To understand potential difficulties for authors, it is important to consider the ways in which their job will change when moving to a reuse strategy.

Authoring tools

One of the key changes to impact the writers is a change in tools. While reuse can be done using traditional word processing tools, for complex reuse companies are turning to XML. A move to XML usually (but not always) means a change in tools. (For FrameMaker users, it might mean a change from standard FrameMaker to structured FrameMaker.

There is real fear that writers will resist a move to new tools, as noted in this comment from Mark Baker:

Everyone has heard (or experienced) stories of CMS or knowledge management initiatives that did not work because content contribu-

tors refused to use the tools deployed or were unwilling or unable to supply content in the format required. The conclusion often reached is that writers cannot give up their WYSIWYG tools and that any attempt to make them do so is doomed to failure. [1]

The fear here is that writers have become so used to the power and the freedom of the word processing and desktop publishing tools that they will resist the move to tools that restrict those freedoms.

Processes

There are two important ways in which processes are directly affected by a move to reuse. First, reuse will not occur unless a writer's first step is to search the content database for reusable content. As one IT person at a company researching reuse strategies put it, "we are potentially turning writers into something more like researchers." For strategies with opportunistic reuse, that is a potential change. It is less likely, of course, where systematic reuse is employed. Writers must be motivated to find and reuse content, rather than recreating it.

Second, a reuse strategy means that content must be written as structured, standalone content. Structured because it must be predictable. For example, you might need to be sure that all procedures have an introductory paragraph. Without predictability, you risk compromises in consistency and quality. The content must be standalone, as you cannot always know exactly where a chunk of content may be used and so you must eliminate any dependencies on other content that may or may not be used at the same time.

Deliverables

One of the key differences that writers may face in a move to a reuse strategy is a change in their personal deliverables. In the past, many of us looked forward

People, Processes, and Change

to the boxes arriving back from the printer, so we could wave a copy of our guide for colleagues to see. Many of us got a lot of job satisfaction from being able to point to a specific book or file and say “that’s the one I did.” A move to reusable content means that more authors might have to share the writing credit. (It goes from being “my” book to “our” book.)

What you need to do

There are a number of things that you can do in your change management plan to help ensure that your reuse project is a success.

Sell the benefits

People are unwilling to change unless there is a very good reason for that change and they can see the benefits.

Writers in technical documentation are facing a productivity crisis because of shortened product cycles, increasing customization of both products and content, and, of course, ubiquitous headcount reductions. The ability to create each piece of information once and to maintain it in one place -- then create many different information products for different products, markets, users, and media from that single source -- can provide a huge productivity boost to writers. [2]

The benefits for authors can include things like:

- Less stress at deadlines
- Faster turnaround times and so more time
- Better tools
- More flexibility to deliver new information products

Communicate the issues

Identifying the benefits is important, but you must also put the changes in context. That means first identifying the pain, issues, and consequences. What are the real issues facing your organization? What is the impact of not addressing these issues? For example, on one project, a client indicated that if they couldn’t produce content faster, there was a real risk that the company would outsource the documentation effort.

Reach out and listen

Listen to what people have to say about the issues and the solutions. If you involve people early on, really listen to what they have to say, then show them that you are addressing their requirements, they will be among your strongest supporters.

Elicit the help of “change agents”

The best way to convince people of a change is to have “one of their own” communicate the excitement and possibilities. A change agent is someone who is not necessarily part of the assigned implementation team, but who will be a user of the new system and methodologies. Make sure that you help the change agents to prepare a consistent message to take back to their team. A consistent message reduces possible misinterpretations.

Get a champion

A champion (someone high enough up in the organization to effect change) needs to endorse the cause and ensure that different content areas understand the need for change and buy into it. The champion may have to insist on them adopting the change or make a change in personnel to facilitate adoption.

Anticipate resistance

You will need to anticipate the resistance that you might get and be able to respond. Some of the common issues for authors include:

- Not invented here
- We do it differently
- Loss of creativity
- There are benefits, but this is too much work
- If fewer people can do more, I may lose my job

Get the right tools

Make sure that the tools you select are right for the authors and not just the right technology. XML editors come in all shapes and sizes. If you are moving to XML, purchase authoring tools that will provide the right functionality for the authors.

References

- [1][2]Structured Content: What’s in it for Writers?, CMS Watch, <http://www.cmswatch.com/Feature/79-Writers,-XML,-and-CMS>

Tools and Technology

The marriage between dynamic and static web content

Ted Spencer

Web Content Management Consultant

ted@tedspencer.ca

Many organizations have independent designs and systems for their dynamic and static web content. But the trend right now is to find a way to give dynamic sites more flexibility, rounding out their transactional capabilities with a more content rich user experience that can be found on static html sites. In this article, Ted Spencer, a web content management consultant, describes how to adapt traditional transactional web site designs to deliver html site flexibility, and facilitate reuse across static and dynamic pages.

Introduction

Large organizations such as financial institutions often base their transactional sites on coded web pages (web pages that are compiled code and execute business logic/display logic) and their marketing sites on flat html. From a content perspective, coded pages sew together small snippets of text and links to wrap around your account data and produce the html that hits your browser. However traditional coded web pages just aren't cutting it these days. In fact, many organizations want their transactional sites to become more content rich, and in the process reap some of the benefits that their static marketing sites have had for years. They want the flexibility to add, remove, and change content quickly, and at the same time be able to reuse that content across both dynamic and static pages.

But along with a merging of abilities comes a dilemma – the common dynamic and static approaches allow for elements of reuse within each approach, but not across them, so a unique approach needs to be developed to bring the two designs together.

Better defining the scope of change

Content architects may not be overly concerned with how a content architecture is “absorbed” by an end application, however, the limitations and benefits of the end application do put huge pressures on content design. And because of this, many organizations view a universal content architecture across both transactional and marketing sites as almost impossible.

To illustrate the pressures imposed on dynamic content design, let's take a look at the present design of many transactional sites at financial institutions. On these sites, dynamic web pages are created from compiled web pages. One popular example is java server

pages (or ‘.jsp’ pages) that you'll see at the end of your URLs. These coded pages often retrieve their content elements such as text, links, and images directly from set relational database fields or from name value pair-based fields in files. Storing content directly in small fields has created a precedence within the content design of these sites; the coded pages organize most of the content, whether the page has lots of dynamic content or not. The pages are responsible for layout, table generation, wrapping html tags around text elements and paragraphs, and even link generation.

All of the code required to piece together a page from small chunks of text is cumbersome, and this is where straight html pages still satisfied a need for flexibility for so many years and made separate marketing sites so popular. Straight html can be changed quickly without any real programming, and its flexibility is close to limitless.

This is why many companies will still argue, “well, who cares - our marketing site allows our customers to find static content (such as new products and company news), and our transactional site allows them to see their balances online. Customers are getting what they need!” But companies with this point of view soon hit a road block when they want to start enriching the user experience on their sites; the dynamic pages need to have the flexibility of the static pages to change content rapidly, and the static pages need to have dynamic capabilities to accommodate for dynamic navigational components or customized user experiences. On top of that, once the company is mature enough from a content management point of view, they realize the downfalls of not sharing their content for common product descriptions, company communications, or online help across these page types.

Tools and Technology

Picking a content design to pull it all together

To pick a content design to support the sharing of content across pages, you need to figure out what exactly you are starting with, then decide where you want to go with it. On transactional sites, the responsibility for pulling the content together is solely placed on the coded page. This is where the big limitation of many transactional sites lies, and is a good start for the initial focus of any design that tries to create greater layout and content flexibility for a dynamic site.

Designing the base content archetypes

Here's an overview of how your design can get around the limitations of traditional coded pages:

1. Keep the small elements that are the cornerstone for the original transactional site design

You'll still need these small text, link, and image elements because you will need to display small snippets of text on pages, depending on what the backend systems see. For example, you'll always need the flexibility for the page to conditionally display a link, such as an additional description or a note about a transaction. At a minimum, these small elements will include:

- Text (one or more words that could contain links or images.)
- Links
- Images

These small elements should also be based upon an xml schema, the rationale for which will be discussed later on.

2. Introduce a more flexible content type based on xhtml

Create a content element that has all the flexibility you're looking for, and at the same time, structure. How? By using xhtml, which quickly introduces the flexibility of html into your transactional site. An area of the page no longer needs to be defined just as a paragraph, text item, link item, etc. It is actually considered any valid representation of xhtml. Then just like a normal html page, that area can grow, shrink, or incorporate any valid xhtml tags it likes. But why are we using xhtml? Because of the ability it gives you to enforce content element reuse.

3. Customize your new xhtml content type

Create your own schema around these xhtml content types, or blocks of xhtml, so that you can enforce reuse of certain content archetypes. For example, if you want to ensure links and images are reused across the site, enforce this in your own customized xml schema, indicating that links are not allowed explicitly in the xhtml. Instead, allow for a reference to a link to be used. When that xhtml block is pulled from the repository by an extraction application, the link gets resolved. This customization step is the real key to enforcing reuse at the element level in the system. Above all you can ensure that links are properly reused across your static and dynamic pages.

Adding structure to organize content

The next step after setting up your primary archetypes is to organize them outside of the coded pages. This allows your server side code to focus more on what it pulls from the backend, rather than on what its doing to weave together the webpage.

References

You'll need some way to reference objects within other objects. As described in the link example above, this is the key to enforcing reuse in the system and not having just a million instances of the same link all over your transactional site. Updating that link would become a nightmare. These references should be allowed to exist within your xhtml block schema, and also potentially exist within text and link elements as well.

There would likely be references for each of the major archetypes

- Xhtml block reference
- Text reference
- Link reference
- Image reference

In addition, a mapping of references to archetypes will be needed to keep track of what references are allowed in each given archetype. Mapping is necessary, since the interrelationships can get intricate depending on how you want to enforce or not enforce reuse on your site.

Tools and Technology

Groups

Grouping is one of the most elemental tasks of making associations, in any type of system. By creating a generic content structure, you open the door to using the structure in many ways. Individual xhtml blocks can be grouped together to form pages, links can be grouped together to form left navigations, and other xhtml objects can be grouped to constitute your standard headers and footer. The coded pages can pull in these groups of xhtml onto their pages very easily – and the code plays no role in formatting or display. The coded pages just pull a set of groups and display them sequentially on a page. But how will the pages know which groups to pull?

Pages

Your coded pages get the groups by referencing their own instance of a page content type. Just as your archetypes form the lowest level basis for your content, pages will form the highest level, and at the very least will only need to contain groups. These groups can be classified or labelled within the page content type in terms of header, footer, main page content, left navigation etc., making it very simple for a coded page to pull in the group to the appropriate area on the page.

Getting the old coded page to work with the new design

Coded pages will now call their respective page content objects and reference a specific labelled group, which will in turn retrieve the large xhtml blocks that are pointed to in each group. The page won't apply any logic when it pulls a group. And that's the key to this design. If a page does not require any coded logic or data retrieval from a backend system, then it just goes to its respective page, pulls in all the appropriate groups of xhtml blocks, and renders the page. When the coded page needs to add in a table etc, it will apply its own logic to that small area of the page, pulling in the smaller text elements or link/image elements as needed, retrieving the base archetype from the repository directly.

The coded pages and the repository now have almost everything they need. They can choose whether to apply logic or not, have the flexibility of xhtml through the newly designed content types, and through the repository's customized schemas for xhtml blocks and smaller schemas attached to the

base archetypes, the system can reliably enforce reuse. With this design, reuse can now exist across your static and dynamic pages since all content elements are driven out of a single repository and page design, not two siloed systems.

You've got the perfect marriage now – with a few base elements – and a bit of structure.

Where to place this content design

Many possibilities exist for storing a content design based on xml. The xhtml block concept definitely requires it, and it's a natural fit to choose a database or repository with a strong support for xml. Xml repository options include native xml repositories or xml enabled relational databases such as Oracle 10G or SQL Server 2005. Xml-enabled relational databases can offer developers easier access to traditional table creation which many developers are comfortable with and can use quite easily to create a lightweight repository application to add/edit/delete/promote content from. Many organizations have been reluctant to invest in large scale CMS applications, and xml-enabled relational databases give many options for home grown solutions.

Conclusion

The cornerstone to a successful marriage between dynamic and static content on a single website is in letting the code pull the data and the repository organizing the content. Increasing a dynamic site's flexibility by adding a content type based on xhtml is a good first step. Then, giving responsibility to a repository for weaving together your content instead of coded pages will solve traditional flexibility limitations.

Through this method a hybrid site will have a way to reuse content across dynamic and static pages since content will now reside in the same repository in the same format, and flexibility will be high since the transactional site now has almost all the attributes of xhtml.

Case Study

A case study in modular documentation

Debbie Donahue
Project Manager, Information Development,
Unisys
Debbie.Donahue@unisys.com

A group of Unisys technical writers (located in Mission Viejo, CA; Roseville, MN; and Malvern, PA) recently moved to a more modular approach to creating manuals. The more modular approach to documentation was precipitated by the company strategy to use common server technology across multiple product lines. In this article, Debbie Donahue, the project manager who led the writers in the Information Development group through this change, describes their approach to and rationale for modular writing.

Issues

The Information Development department needed to keep pace with the Engineering development process. Unisys had started to use a modular approach to developing its products, using common server technology in multiple product lines. By doing so, Engineering was able to cut development time because they were reusing components. In the meantime, the writers couldn't keep up with the pace. The writers were also realizing that there was much overlap in the documentation they were creating, and that they could reuse common content. However, the reuse was manual, with writers e-mailing each other and using the "word of mouth" method to find content they could reuse. By employing a more managed approach to content reuse, the writers hoped to be able to keep up with the fast pace set by Engineering.

What we did and why

The previous documentation development process at Unisys went something like this very traditional approach: writers reviewed a combination of engineering documents, performed some hands-on testing of the products in the lab, interviewed the SME (subject matter expert), created the documents, sent the documents out for review, and so on. Writers were responsible for an entire document and saw that document through its entire life cycle. However, to keep up with Engineer's development life cycle, writers had to change their mindset about how they created documentation.

The new approach to developing documentation is a modular one, mirroring the Engineering approach to developing hardware and software. We established a taxonomy that includes various roles required to cre-

ate the documentation. Within the taxonomy, object owners are responsible for a set of "objects" related to a particular area. For instance, one writer might own 15 objects related to the management software. We decided to have object owners throughout the functional areas because of our continued emphasis on writers to know about and becoming experts in the content they write. It's impossible to know everything, so we thought that if writers became responsible for objects related to a particular product area, they could become experts in that product area.

Within the taxonomy, we also have document leads who are responsible for taking all the objects (or content modules) and putting them together into manuals. The document leads also create the unique content required to complete the manuals. The document leads have the most product knowledge and are able to use their expertise to build documents from the various modules created by the object owners. So, object owners create the various modules required for a particular product area, and the document leads compile those modules and other unique content into manuals.

Challenges

The process is working extremely well, but that's not to say there are no kinks to be worked out! Probably our biggest challenge occurs when we get "squeezed." We spent a lot of time flowcharting our processes, and when there is enough time in the development cycle to follow these processes, all goes smoothly. But, when there isn't enough time, it seems we still have to scramble to get things done. When squeezed, the writers tend to go back to the old way of doing things, sometimes creating new content objects instead of reusing existing ones if they feel it

Case Study

takes too long to get to the existing content objects through the process.

Another challenge in adopting a modular writing approach is the creation and enforcement of guidelines that describe how the content is to be written. Because content is being used in different places, it has to be written so that the modules all fit together. We have begun putting together writing guidelines to address both structure and style and are realizing that we need to be stricter in what those guidelines say and in how they are enforced. Some guidelines, especially structural ones, can be enforced through the DTD and through different mechanisms of the CMS. However, in other cases, the editing team must review documents very closely against the established standards and ensure that all pieces of the modular content adhere to the documented standards. This also means that the writers and the editing team must work most together, making sure that both writers and editors understand specifically what the guidelines mean.

Another challenge is in the changing roles of writers and editors. Our modular approach works really well for some writers, but for others, the change from ownership of a complete document to ownership of a set of objects has been challenging. We did some things early on that helped writers to make the transition, including

- Creating a mock CMS in Word and introducing component-based authoring to an initial team of about 10 people.

This initial team worked on creating components for release announcements and overview documents; team members created components and several writers would bring those components into the appropriate documents.

- Training our writers in structured and topic-based writing, and creating a structured writing environment with specific rules for how content must be written and structured.
- Having general communication events during which we talked about content reuse, content management, and structured writing, so that writers and editors were exposed to the new concepts and methodologies.

Naturally, some writers are more open to doing things in a new way, but the more reluctant are also realizing the benefits of content reuse. They can see

the advantages of finding content in the CMS and using it to help them meet their deadlines. Many objects are reused without alteration; only about 25 percent need some additional information specific to a particular platform. It really comes down to explaining the business reasons for creating modular documentation. The trend is for companies to do more with less, and from a business perspective, companies, including Unisys, would not be able to meet the clients' needs because they don't have the manpower to re-create content over and over again.

Benefits

Besides being able to create documentation more quickly, we are also starting to see some improvements in the content. Objects are rigorously edited according to style and structural guidelines, and we're seeing fewer inconsistencies in content because it is being reused. So, if the content needs to be in two manuals, it's consistent in those two manuals. When we began creating modular documentation, we focused more on the processes, which left some content as just "okay." Now, we are focusing on getting the content right and on getting good, solid guidelines put into place based on what we've learned about our content while ironing out the processes! Implementing a new method of creating manuals is always iterative. While working on the processes, we saw issues in the published content that can be addressed in the next iteration, and in the meantime, we honed our writing guidelines.

Lessons learned

While the modular approach is very useful and has helped us to keep up with the fast development pace, it's very important to bring writers and editors on board incrementally, showing them the benefits along the way. Training and communication are critical throughout the transition because we're changing the way that many writers and editors have worked for years. The mock CMS was very beneficial in introducing the concepts to the "early adopters" in the writing group, and through the process, other writers were able to see the benefits. It's also important that you define the scope of your project. We have such a big group of writers and such a diverse group of products. Our initial scope was to use our modular approach and the CMS for one product area, with about 30 writers. Start small and let your project expand once you're successful in one area!

Case Study

References

- [1] Rockley, Ann, Pamela Kostur and Steve Manning. 2003. *Managing Enterprise Content: A Unified Content Strategy*. Indianapolis, IN: New Riders.
- [2] Anderson, R.E. 1992. "Social impacts of computing: Codes of professional ethics." *Social Science Computing Review* 2, 453-469.
- [3] Schwartz, M., and Task Force on Bias-Free Language. 1995. *Guidelines for Bias-Free Writing*. Bloomington IN: Indiana University Press.

In the news

Scott Abel
The Content Wrangler
abelsp@netdirect.net

Content reuse is one of the biggest benefits content management provides. Effective reuse of content improves productivity, reduced inconsistency, and makes content more accessible. In this installment of In the news, you'll find links to resources that can help you master the fundamentals of reuse, make sense of statistics on content reuse, and find other content-related metrics that can help you make the business case for content management in your organization.



Opportunities for Cost-Savings and Efficiency from Content Reuse and Re-purposing

The TechDoc Community of Practice (<http://www.idealliance.org/news/2003/idea1030.asp>) provides a sample content reuse return on investment analysis in Opportunities for Cost-Savings and Efficiency from Content Reuse and Re-purposing (<http://xmlu.com/resources/TechDocWebinar3.pdf>). They also explore reuse versus re-purposing, describe the framework needed to enable content reuse, and provide commonly overlooked content reuse opportunities.



Failing To Reuse Content: Ortho Hornet And Wasp Killer

Content management provides a plethora of potential benefits, one of which includes content reuse (write it once, use it often). Reusing content that should be identical in all instances in which it is used is a smart move. It can increase productivity and reduce content creation expenses by eliminating the need to rewrite content that exists elsewhere. It can help ensure consistency of information, can reduce translation

expenses, and can help stave-off legal and regulatory compliance issues.

In Failing To Reuse Content: Ortho Hornet And Wasp Killer (http://www.thecontentwrangler.com/comments.php?id=P155_0_1_0) *The Content Wrangler* documents a real-world content snafu that could have been eliminated by effectively reusing content. Read the article and take a look at both the product label and companion website (links provided).

How much time does it take to unleash web rage?

Search Engine Watch: "On average, web rage is uncaged after twelve minutes of fruitless searching, although about 7% say ire starts rising within three minutes," according to Roper Starch Worldwide (<http://searchenginewatch.com/sereport/article.php/2163451>). "The main culprit is all that information – overwhelming at times – which is actually driving some people offline and back to telephoning customer service or other information resources from the pre-cyber generation."



How much time do we waste searching for – but not finding – information that exists?

According to research from International Data Corporation (IDC), "the amount of time wasted in futile searching for vital information is enormous, leading to staggering costs to the enterprise." IDC studies

In the news

have found that the cost of searching for – but not locating – information costs an organization of 1,000 workers about \$6 million US a year. IDC says, “That’s not including opportunity costs or the costs of reworking information that exists but can’t be located.” When these costs are added to the calculation, an extra \$27 million US a year is lost. These statistics are included in insightful overview from *Thomson Scientific* entitled *Strategies for Search, Taxonomy and Classification: Getting Just What You Need* (http://scientific.thomson.com/knowtrend/ipmatters/infosearch/8228762/search_tax_class.pdf).

How much time do we waste recreating content that already exists?

Kit Sims Taylor, in a paper presented at the *International Conference on the Social Impact of Information Technologies* in St. Louis, Missouri, finds that knowledge workers spend more time recreating existing content than creating new content. Now here’s a case in point for content reuse.

Taylor says, “Roughly one-third of productive time is spent in knowledge reworking. The other nearly two-thirds is spent in knowledge finding and communication, with only about 10% of time spent in actual creation of new knowledge... While most of us do not like to admit that much of our creative work involves reinventing the wheel,” Taylor writes, “an honest assessment of our work would indicate that we do far more recreating than creating.”

The logo for Intelligent Enterprise, featuring the word "intelligent" in a large, bold, black sans-serif font, and the word "enterprise" in a smaller, red, bold, sans-serif font directly below it. A thin red vertical line is positioned to the right of the text.

Reusing content provides Hewlett Packard will significant savings

Doug Henschen, Editor, *Intelligent Enterprise*, writes: “There’s plenty of hype about *enterprisewide* content management, but few companies have come as close as Hewlett-Packard to taking a truly holistic approach.” Henschen interviews Mario Queiroz (http://www.intelligententerprise.com/channels/content_management/showArticle.jhtml?arti-

[cleID=165701667](#)), HP’s vice president of content and product data management, who led the company’s three-year effort to rationalize taxonomies, metadata, technologies and management approaches spanning 17 business units. “The deployment touches some 85 percent of the products sold by an \$83-billion technology giant, yet the practices aimed at efficient content reuse are pertinent to any size organization.” Although Queiroz is tight-lipped on the specifics, he provides some cost savings figures that may prove valuable.

Contributors

Scott Abel

Scott Abel is a freelance technical writing specialist and content management strategist whose strengths lie in helping organizations improve the way they author, maintain, publish and archive their information assets.

Debbie Donahue

Debbie Donahue is a project manager in the Information Development organization within Unisys Corporation. She has been in the technical writing field for 15 years. Debbie and other members of the Information Development team starting researching component-based authoring and content management systems more than 7 years ago. After building a business case and showing sound return on investment, the department purchased a CMS 2 years ago. Debbie was excited to take on the challenge of leading the CMS implementation. With the help of an excellent technical staff and a dedicated team of writers and editors, the implementation was a success. Debbie looks forward to the continued growth of the system.

Suzanne Escoffier

Suzanne Escoffier is a Senior Manager of Information Development at Symantec Corporation. In that capacity, she manages the development and delivery of all in-product user assistance materials for the Consumer product line. Prior to joining Symantec in 2000, Ms. Escoffier was President of mediamerge, Inc., a company specializing in documentation system design for small- to mid-size businesses. Ms. Escoffier has been active in the technical communications and multimedia fields for 20 years, publishing computer-based training, educational software, and help systems for both the Windows and Macintosh platforms.

Pamela Kostur

Pamela Kostur is a Principal with The Rockley Group, specializing in information analysis, information modeling, and structured writing to support a unified content strategy. Pamela has over 18 years experience developing information solutions. During that time Pamela has completed many projects and presented papers at numerous conferences on topics including iterative usability, miscommunication, structured writing, editorial “magic”, building and managing intranets, creating usable online documentation, uni-

fied strategies for web-based learning, information modeling and analysis. Pamela is a co-author of *Managing Enterprise Content: A Unified Content Strategy* with Ann Rockley and Steve Manning.

Suzanne Mescan

Suzanne Mescan is Vasont Systems’ (www.Vasont.com) Vice President of Marketing for its flagship Vasont software, a content management system. She has more than 18 years’ experience in the information management and publishing fields. She may be contacted at smescan@vasont.com.

Ann Rockley

Ann Rockley is President of The Rockley Group, established to assist organizations in adopting content management, unified content strategies, and information architecture for content management. Ann has been instrumental in establishing the field in online documentation, single sourcing (content reuse), enterprise content management, and information architecture of content management. She is a frequent contributor to trade and industry publications and a featured speaker at numerous conferences in North America and Europe. Ann is the author of *Managing Enterprise Content: A Unified Content Strategy* with TRG Senior Consultants Pamela Kostur and Steve Manning.

Ted Spencer

Ted Spencer is an independent web applications consultant who specializes in integrating content management approaches into large scale web apps. His most recent experiences have been in the integration of the Oracle 10G XML repository to BroadVision and Weblogic. Ted has a broad technical lead background in the design and implementation of ERP, CRM, enterprise integration and CMS for web applications. He is presently using this experience to bridge the gap between content and application design within organizations.

Contributors

Steve Manning

Steve Manning is a Principal with The Rockley Group and has over 16 years experience in the documentation field. He is a skilled developer of online documentation (WinHelp, HTML Help, Web sites, XML, and Lotus Notes) and has created single source production methodologies using key online tools. Steve has extensive experience in project management and has managed a number of multiple media, single source projects. Steve teaches “Enterprise Content Management” at the University of Toronto, and is a frequent speaker at conferences (ASIS, AUGI, STC, ACM SIGDOC, DIA) on the subject of XML and Content Management. Steve is a co-author of *Managing Enterprise Content: A Unified Content Strategy* with Ann Rockley and Pamela Kostur.