

Taming the Information Jungle:  
A Brief Treatment of Creation Dichotomy in Content Management Systems

May 3, 2004

Kalev Leetaru

## **The Cacophony of an Information Jungle**

The World Wide Web is an untamed jungle of information strewn about cyberspace. This cacophony of disarray echoes through the internal organization of many large web presences. When tens or hundreds of thousands of web pages must be continually updated and maintained by a small army of individuals hailing from every branch of an organization and every location it has offices, chaos reigns supreme. Enforcing a single external design, a single personality, a single set of terminology, quickly becomes an exercise in futility without some sort of central mediating force to bring together all of these disparate entities. Fundamental differences in the way different departments communicate ideas and their notions of how that information should be published to the web, must all be melded into a single corporate image, presenting the external world with an illusion of seamless conformity.

## **One Lion's Roar to Rule Them All**

The term *content management system* (CMS) is used to describe the broad spectrum of software packages and their supporting processes and methodologies that take on this role of central content mediator. Rather than allowing each team member to develop a section of the site in isolation, with ad-hoc teams of developers creating their own subgenres of the corporate image, CMS packages allow a single global image to be defined and subsequent design decisions to be arbitrated against this global template. Advanced packages even incorporate cross-domain ontologies to map terminology across departments within a company to ensure consistent language use.

Under this model, the design process becomes symbiotically linked with the content management system. Decentralized development of sections of the site can still occur, allowing maximum productivity, but critical components of the design are globally enforced by the CMS and the major phases of the project involve checkins with the CMS, which ensure global cohesiveness. The traditional CMS workflow involves the initial definition of a global image for the site and subsequent creation of a set of global templates. These templates will serve as the containers for the site's content, ensuring macroscopic unification. As individual content areas are produced, they are uploaded to the CMS, which links them to the appropriate templates and performs any assigned validation tasks. In a fully decentralized publication environment, content may be generated by team members without organizational authority to externally publish, and hence their content may be automatically queued by the system for approval by more senior members before actually being published to the web.

## **The JungleMaker**

A web site is driven by its content and hence one of the most important interface aspects of a content management system is how content is actually submitted to the system. The intuitiveness, performance, and robustness of the content submission mechanism heavily influence the way in which users interact with the system. If users have a difficult time submitting content, they are more likely to perform more of their work outside of the system and only submit sporadic updates. This offsets many of the

gains provided by the constant asynchronous collaboration of regular content submissions.

### *The Creators Diverge: The Creation Dichotomy*

There are two primary classes of content management systems that use two very different models of content representation and submission. The most common is the *intrapage management system*, which operates on a page-centric content model. Content is uploaded as individual web pages and the links between those pages are manually defined. Template-defined regions and dynamically generated content are established through the use of special demarkers within the web page. For an organization used to dealing with content in terms of web pages, intrapage management systems are an evolutionary step towards bringing an additional layer of cohesiveness to the site.

In contrast to the page-centric model of intrapage systems, *total management systems* decouple content from presentation. In these systems, content is uploaded as individual *entities*, such as an article, a calendar event, or a photograph and its associated caption. The system assumes complete responsibility for generating the web pages themselves and establishing the link structure for the site. Thus, instead of the content author *defining a page and marking where template and dynamic elements should appear*, the author *defines where content should appear within the templates* and then submits individual content entities for display within those templates. This is a substantial paradigm shift from what most technical users are familiar with, and hence the transition to this type of system has the most rewards in terms of the new capabilities it offers, but also the greatest cost in learning curve and retraining of technology staff.

### **Choosing Sides on the Verge of Divergence**

The dichotomy of content representation and submission between intrapage and total management systems presents an organization with its first major interface decision when implementing a content management system. It must choose the model that most closely matches its web publication needs, while keeping in the mind the thought process of its developers. If its developers are veterans of the page-centric web world, then the human costs transitioning to a entity-based system may be too great. If the bulk of the organization's proposed new content is page-based to begin with, such as a static site with very little reuse and inter-referencing of content, then there are no inherent advantages to an entity-level system.

### *Inside the Intrapage Experience*

The user experience with an intrapage management system is not fundamentally different from traditional web authoring. Such systems are often implemented on a technical level through *server side includes* which invoke static HTML pages and dynamic CGI scripts to produce content to be inserted inline with the original page content. Content is usually submitted outside of the system through an FTP or shared filesystem, requiring a fair amount of technical fluentsness. Tools for inserting dynamic

elements into pages range from just a toolbar of proprietary tags that is displayed in an HTML editor, to WYSIWYG tools for interactively placing dynamic elements.

The creation environment provided by intrapage management systems is decentralized and highly technical in nature. Very little is done to hide the underlying technology, meaning that such systems really just provide a more efficient workflow for advanced web authors to achieve content unification. They do not stature themselves as enablers for the average company employee to contribute to the content production process, as the technical details of authoring and submitting the content under such a system are often above their skill level.

### *A Tear in the Fabric of the Intrapage Universe*

The page-centric model of intrapage CMS systems has one critical flaw from an interface standpoint: not all data that such systems must manage is page-based. For example, a calendar event must be presented to the system as a date-driven event, with sufficient semantic metadata to allow the system to internalize the event and display it appropriately based on that date. An intrapage CMS system does not have the interface mechanisms to allow such a calendar event to be submitted through the CMS' page submission interfaces, and thus the calendar subsystem of the CMS must provide its own external interfaces. This interface disjunction fragments the user experience, forcing a mental context switch amongst different types of content submission. When submitting a web page, the user employs an external FTP or file sharing mechanism. When scheduling a calendar event, the user navigates the calendar subsystems' separate interface. When posting a message to the bulletin board, yet another independent interface must be utilized. Forcing the user to continually switch between different interfaces to perform different authoring tasks produces an environment where a wide variety of skills across many different interfaces is required even for basic competency with the CMS. This workflow will often be more comfortable to highly technical users who have grown accustomed to continually switching amongst a set of development tools to perform specific tasks. However, non-technical users, who often strive to gain a working familiarity with just one or two essential job tools, will obviously not be candidates for this system.

### *Totality to the Rescue*

Total management systems treat content on the entity level, rather than the page level, providing a consistent metaphor regardless of semantic meaning. A calendar event is submitted via the same overall interface mechanism that an article or a bulletin board entry is. Even though independent modules may be providing the functionality to support each of those content types, the content management system provides a unified submission workflow and often additionally enforces a consistent set of interface metaphors. Hence, the user does not perceive any kind of paradigm shift between submitting an article for a web page, scheduling a calendar event, and posting to a bulletin board. Even though the three actions necessarily require different interface elements to provide their domain-specific information, the overall interaction experience is the same, and all actions are initiated through the same entrance points in the content

management system, such as a set of administrative pages that branch off to the specific submission or management task.

The real power of total management systems lies in the separation of content and presentation. This separation often presents a steep learning curve to web authoring personnel, in that it forces them to recreate their established web publication workflows. Simultaneously, however, it greatly lowers the bar of required skills for publication by empowering the average employee to contribute to the content production process. By allowing employees to just focus on writing a particular article or scheduling a particular calendar event, without worrying about how that article or event will be displayed and where in the site's organizational structure it needs to be located, the user is free to focus on content instead of design details. Non-technical users are more apt to feel comfortable with this content-centric model, where all of the technology and web presentation details are hidden.

The content-centric model also enjoys the benefit of potentially enabling greater productivity in decentralized publication environments. Under the page-centric model of intrapage management systems, the non-template links on each page must be fixed at creation time. Hence, if the human resources department wishes to have a link on its employee benefits page to the business department's payroll query page, the final URL of that query page must be known before the link can be made on the human resources page. In a large-scale corporate deployment, this may not be a problem, as every single web page may be accounted for in an enterprise deployment document, but this can become a problem down the road as the site evolves through time and new sections are added. The content-centric model, on the other hand, leaves content linkage to the CMS system itself, which it performs at runtime when it displays a given content entity. Hence, content can be submitted in ad-hoc fashion, with no specific ordering requirements, and the system will calculate and display the inter-content links. By eliminating the tight synchronization of the intrapage model, greater productivity is possible and more non-technical content authors can participate in the process.

### **A Tale of Two Audiences**

When deciding what type of content management system to deploy, an organization must look very carefully at the type of content and services that it is proposing for its new site. However, often the decision of which system to adopt is made solely on the contents of the new web site and not on the process involved with interacting with the new system. Hence, a system is deployed based on its features and employees are simply expected to adapt to the requirements of the new system, instead of tailoring the system to the employees. What is overlooked in this case is that to end users, the interface *is* the application, and so instead of exploiting the advanced featureset of the CMS system, users will regard the entire system as encumbering and primitive if the interface to those features is poorly designed.

To choose between an intrapage and a total management system, an organization must recognize that there are really *two* audiences that must be identified. The most obvious, and the one that often receives the most attention, is the audience for the content itself, *the consumers*. However, the content creation demographic, *the producers*, is

almost more important, in that a disenfranchised publication audience will demonstrate very poor productivity towards generating and maintaining content.

### **Taking the Path More (or Less) Traveled**

There are many variables that an organization must consider when choosing which type of content management system to deploy. Rather than focusing exclusively on the technology side of the new system and on the consumer, an organization would be wise to also look within, to the producer. Conducting an audience assessment of the content publication team and whether an enterprise wishes to concentrate its publication power within the hands of a select group of technically savvy individuals, or whether it endeavors to involve the entire organization in the process, can help determine whether a page-centric or content-centric model would be more appropriate. By more closely matching the interface with the audience, both consumer *and* producer, everyone is better served.

### **Suggested Further Readings**

- <http://www.cmsreview.com/>
- <http://www.cmswire.com/>