

## CONSIDERING CONTENT MANAGEMENT IN A SOFTWARE DEVELOPMENT COMPANY:

Taking single-sourcing seriously

Kirsty Taylor

*School of English, Media Studies, & Art History, University of Queensland,  
Brisbane, Australia*

Roslyn Petelin

*School of English, Media Studies, & Art History, University of Queensland, Brisbane,  
Australia*

### **Abstract**

*Advocates of single sourcing maintain that managing content via single sourcing provides documentation and technical publications departments, and the software development companies that they are part of, with the opportunity to develop content-focused, format-independent, quality documentation. Technical writers can write with the audience as their focus, without being distracted by the formatting of their final output. The paper focuses on the example of a software company and the potential improvements this company could achieve by pursuing a single-sourcing management plan for their documentation. This company is a composite of various software development companies.*

### **Introduction**

Single sourcing, which is the process of producing a chunk of content once and using that chunk of content many times for various output formats, has become a hot topic in the field of technical communication over the past several years. Its advantages and disadvantages have been debated on technical writing e-mail lists (such as the techwr-l list) and in sessions at industry conferences, such as the Society for Technical Communication (STC) annual conference. For technical communicators who are engaged in single-sourcing projects, this methodology and approach is more than just the latest 'cool thing' to work with, it can be a solution to the laborious process of producing multiple outputs from multiple source materials that should at the very least be consistent with one another, if not drawn from the same source.

Using single sourcing, a chunk of information about a product can be used in the user manual for that product, in the user manual for a complementary product that is also made by the company, on the company's website as marketing collateral, in product brochures, and in tender or quote response documents. This chunk of information is written independently of the format of the final output, that is, authors do not concern themselves with issues of formatting that they would need to worry about if they were writing a single, larger document. Authors focus on the content of what they are writing and aim to keep that content self-contained, so that it can be used many times in various documents without another author needing to copy or re-create it, without needing other chunks to accompany it, or having cross-references to other chunks that will be broken if the chunk is used by itself.

Re-using text or content extends beyond the standard technical communication practices of re-purposing content for multiple outputs. Frequently, a technical

communicator will author a user manual and then use that manual as the basis for the online help. A conversion tool is used to convert from, for example, the Microsoft Word source to HTML Help format when the authoring process is complete. Different formatting is applied to the two outputs; maybe some of the content that is included in the print manual is removed from the online help, but essentially the overall content of the two outputs is the same in structure and level of information. Content is not finely tuned to the different user requirements of online help from the user manual. Using single sourcing, authors can “identify all the information requirements up front, then [develop] them from a single source” (Hackos and Rockley 1999, p. 3) rather than simply converting one file into two output formats.

From the first discussions of single sourcing in the late 1990s, many of the theories and understanding of single sourcing have developed into the concept of content management. Rockley et al. (2002) define content management as “a system or the capability to manage and track the location of, and relationships among, a firm’s content at an element level” (p.433). Single sourcing and content management are not equivalent; a technical communications group can implement single sourcing of content without using a content management system, but implementing content management usually involves implementing some level of single sourcing. Content management moves away from the technical publications focus that single sourcing can have and into a wider view of corporate content, and knowledge storage and use. Content management can be used and applied even at the enterprise level, so that content that is created is used in as many different documents as possible, without the necessity to rewrite or copy the information from one source to another. The original source is maintained and linked into various outputs (potentially including press releases, user manuals, online help, support websites, company websites, and so on) so that when it is updated, all of the different outputs that use that content chunk material are also updated. Content units, or chunks, can be stored in a content repository at differing levels of granularity—chapters, pages, paragraphs, sentences or even words. How large or small your chunks of reusable content are depends entirely on what, how, where, and how often the chunks are re-used.

Before considering content management in much more depth, we need to understand what content is and what kind of content the company examined in this case study has right now.

### ***Sample Analysis***

Currently within GuruCom, a variety of staff produce a variety of documents for a number of different external audiences. Similar products in different output formats have been produced by different teams. An online help system exists in Karma, GuruCom’s major software product, that has a very different look and feel depending on which screen you are looking at. From the end-user’s point of view, it is the same online help system. A number of technical staff who are not technical writers are responsible for writing and maintaining user reference material. A large volume of end-user product documentation has been written without an audience analysis or profile, without strict, enforced content guidelines, and without necessarily being kept up-to-date. This set of documentation cannot be spell checked, is not edited by anyone at any stage, and has inconsistent language throughout the 5000-plus files.

A second set of Karma product documentation—the Online Reference Manuals—has over 4000 files, which have been created and maintained by technical writers and subject matter experts. As an automated version control system has not been implemented over the Online Reference Manuals, it has been difficult to produce a full set of Online Reference documentation that matches minor releases of the Karma software. The Online Reference Manuals have been written with a particular audience group in mind (key users of Karma) and all files are edited before being distributed to customers. When documentation updates are required, technical writers need to update two entirely separate sources, the online help and the Online Reference Manuals, to ensure that all product documentation is complete and correct.

The technical writers and editors need to use a range of different authoring tools for the various sets of documentation that they work on. For the Online Reference Manuals, writers use Microsoft FrontPage to produce HTML source and editors use HTML Help Workshop to produce HTML Help output. For one part of the online help, writers use Word documents for the source and then HDK is used to convert the Word documents into context-sensitive HTML output. For the other part of the online help, writers use an in-house UNIX editing program to update the files. This editing program is a powerful conversion tool, but when you convert to PDF format for printing or to HTML format for online help, you include all content from every file; it is not possible to limit what is converted.

In the past few months, some processes for technical writers to follow when they have to make updates to any product documentation have been implemented. The technical writers can follow these new processes when they are told about functionality changes to Karma or incorrect material in the user reference, but this process is instigated by others (business consultants, technical development staff, and so on), so it is imperative that all staff raise the appropriate documentation request forms for documentation changes.

Employees who could be classed as “non-writers” also produce documents that are sent to customers, including, but not limited to, Design documents (DDs), Overview documentation, Release Notes, Program specifications, documentation for new software, technical bulletins, and technical training material. At least some of these writing tasks involve rewriting existing material or writing documents that will be the basis of other documents (for example, DDs are a common resource when developing end-user documentation). Non-writers use Word almost exclusively to create and maintain their documents, as it is a part of their Standard Operating Environment (SOE), but, where necessary, they also use the UNIX editing program for any updates they make to the user reference material.

The existing online help has been created and maintained over time without any clear audience definition or focus. In Hackos’s *Managing Your Documentation Projects*, she emphasises how important it is for documentation to be based on the audience’s needs: “without firsthand knowledge of the audience, without a rich picture of the people who will be reading and using a document, you are very likely to design and build documents that are of limited success in meeting your audience’s needs” (1994, p. 120). Regardless of who is writing a document, the author needs to consider their audience as they write, so that they can write meaningful documents. Documents need

to be focused on a very specific audience definition; otherwise, the information created may fail to be of any use to any readers.

### ***Content and effort duplication***

One of the main points of content overlap is between the OnlineReference Manuals and the online help. By analysing just the end-user documentation, it is clear that the two groups of Karma documentation are going to be a burden to maintain. Technical writers need to make content changes in two or more files whenever Karma functionality changes, using at least two different authoring tools and two different conversion tools to produce the final outputs. This duplication of effort takes up time that could be spent creating new content or reworking and refining existing documents to better suit the needs of the intended audience; after only a few months of this duplication, the technical writers are already frustrated with making changes twice, using different tools and writing in different styles. GuruCom needs to change the way it creates and maintains end-user documentation to provide a better customer experience as well as a better authoring experience.

### ***Rationale***

Given the existing range of documentation and authoring tools, GuruCom needs to move towards a more robust, repeatable, refined, and holistic documentation solution. Karma has a wide range of users for the various areas of functionality. In order to increase customer satisfaction with their overall experience of Karma, GuruCom needs to make it easier for customers to use Karma, to find the information they need in the supporting documentation, and to make the information relevant to their specific installation and job role. To achieve this, GuruCom needs to redefine and re-create documentation processes.

Alone, single sourcing and content management cannot solve all of the documentation problems. These methodologies and the technologies that implement them will not make the writing styles of our existing documentation sets more similar, nor will they inherently help us to create more user-focused documentation. What these methodologies can help to create is a model and a structure for developing consistent, structured, reusable content that can be deployed in multiple formats and even create dynamically, depending on the requirements of the customer. With sufficient planning and specification, GuruCom should be able to produce online (online help or web-based) or printable documentation sets, to customers, that provide them with information specific to the Karma processes that they are using. This same set of source information can also be used to create documentation for internal audiences, but at a greater level of detail. Rather than trying to maintain multiple documents, all with different audiences and at varying levels of detail, using single sourcing, there can be one repository of information for all documentation required or produced by GuruCom—regardless of the desired output medium of the different audiences.

The basic premise of the single source vision is to write a chunk of content once and reuse that content many times. In this instance, reuse does not mean to copy that content and paste it into multiple other documents; it means to pull in that content from a repository and use it in other documents or files, but still maintain the reference back to the source chunk in the repository. This way, updates can be made to the content and then be propagated throughout all instances where that chunk of

content is reused. In a standard copy-and-paste situation, it is a difficult and manual process to propagate such updates. Using content management also removes the content from the output medium and its formatting. When authors can write in an environment where they do not have to concern themselves with the formatting of the final output, they “can be 50 to 100 percent more productive when they are relieved of the responsibility of formatting” (Bartlett 2002, qtd. in Hackos, p. 320). This extra productive time can be used to create new content. All formatting is applied when the output medium is chosen; it is at this stage that the templates for the output medium are applied to the content. By keeping the content separate from the formatting, it is also much easier to reuse that content in multiple deliverables, for example, an overview paragraph can be used in a user manual as well as a Marketing product overview. Content can also be tagged semantically and the semantic tags used to build output, for example, everything that is tagged “detailed” goes into the manual, but everything that is tagged “simple” goes into the online help.

When content is single sourced, it only needs to be translated once, and that translation is propagated to all instances of the content chunk. Customised deliverables can also be created for individual customers, with only information about the particular product or components that they purchased. The users do not need to determine whether they have a particular process installed, as they have information only about their own system.

Depending on the requirements that come from a full user-analysis and information model, a repository of content units (chunks of information) can be created at a particular level of granularity. That level may be the paragraph or it may be the sentence; the information model should define how granular is practical for the specific purpose. Whatever level of granularity is appropriate, the content units must be reusable. Last year, we updated the Karma 2.0 Online Reference Manuals for the changes made to the User Interface (UI) in Karma 2.1. Many of these changes had to be made on an individual, screen-by-screen, file-by-file basis—for almost 5000 files. The Submit Now field was removed from almost every screen in the UI, and therefore, screen summary; if this field had been a content unit in a single source repository, we could have deleted it once rather than many hundreds of times. If it takes two minutes to identify and delete a field on a screen summary and there were approximately 1400 screens updated, then more than 40 hours—one whole working week—would have been saved that could have been spent developing new Karma 2.1 content.

In a recent survey, customers were quoted as finding the documentation too high-level and not actually useful from a user’s perspective, as well as lacking content they expect to find. Interviews with staff in GuruCom about the kinds of product documentation they expect and need to perform their jobs have revealed that even internal audiences cannot rely on the documentation that is created, either due to its not being up-to-date or lacking in detail. We need to try to ensure that the documentation meets the needs of the target audiences and that it is of a professional and international standard. If we focus first on the Karma product documentation and bring it into a consistent, single content repository, then we can use this framework to incorporate the documentation of the other GuruCom products.

### ***Where to from here?***

The first step we need to take before looking at authoring tools that enable single sourcing or investigating content management systems is to analyse our audience types (and where possible, interview individual users in their working environments) and learn what type of documentation and level of detail they need. As well as the external users, we have internal users whose information requirements are different from the end-users. After we have thoroughly analysed our audiences, we need to develop an information model that precisely categorises all information resources that are or need to be developed. This framework of the information model “provides the basis on which you base your publishing architecture” (Hackos 2002, p. 124). When the information model is developed, we can start developing content plans for the deliverables we need to produce to meet the audiences’ needs. From that point, we can develop a single sourcing strategy to define how we can produce quality documentation for multiple versions of Karma, in multiple formats, for a variety of audiences, which is easily translatable and localisable. Then we implement the strategy.

When we have defined our audience types and understand exactly what information they require and what type of documentation they want, then we can develop an information model. The goal of the information model is to create a “complete specification outlining how content is used—and reused—throughout your organisation” (Rockley et al. 2002, p. 161). The information model is the crucial blueprint that defines how content should be organised and structured to create deliverables. In creating our information model, we will most likely work from neither the top down (basing everything on the user’s needs) nor from the bottom up (basing the model on our existing documents and information), but a combination of these approaches. We have a large amount of existing information—in the two sets of Karma product documentation alone, there are almost 10 000 files of information—but we cannot assume that what we have is ideal and has the user’s needs in mind.

One element of the information model will be to create the taxonomy we will use to categorise and label the dimensions of information. These labels need to be meaningful and need to be aligned with the content of the information types rather than a format. Typical templates in products such as Word or Adobe FrameMaker assign formatting to a content unit (for example, paragraph or heading 1), but they do not enforce the content that can be used in the element or in what structure content units can flow; such template tags only define that a paragraph must be single spaced, in Times font, with a height of 12 pixels. The content of a paragraph can be anything. If meaningful, semantic tags are applied to content chunks, then authors can search for existing content before writing anything, just in case the content already exists. If the tags are not meaningful, then when the technical writer searches for existing content, they may not be able to find the desired chunk of content. Technical writers and other authors then need to overcome any desire to alter the existing content (unless it is false) because it has not been written the way they would write it. To help guide the technical writers and other authors in their writing process, a very strict structure and set of rules for what is allowed to go into a content unit must be enforced. These labels are also called metadata, or information about the information.

Within the information model, we need to define the information types that will be developed. Typical information types in technical documentation include concept, procedure, task, and warning. Each of these information types should have its own defined structure, which will make it easier for technical writers to write and easier for users to read and understand the information flow. If a procedure is defined to have a structure that consists of an introduction and procedure steps, then every procedure will have that structure. The items within the structure of the information type are called content units. Content units are the smallest chunks of information within the information model and, if appropriate, they can be used in many information types. For example, if both a procedure and a task need an introduction, for ease and consistency, that introduction content unit can be used in both information types. There might be other content units that are unique to an information type and their definition cannot be reused across many information types. Content can be reused at either the information type or content unit level, whichever is most appropriate for the information model and deliverable type (for example, online help or website). Content must be structured if it is to be reused throughout various documents, document types, or groups within an organisation. Any information types in the information model should be based on the user's needs and on our business and product needs. We will need to carefully look at our existing information to see how it will map to the information types and content units that we develop in the information model.

To help us to produce documentation of a professional, international standard, which meets the needs of its audiences, we need to put together a project team to examine the possibilities in more depth. The team should comprise the following roles:

- Project Manager
- Information Architect/Information Designer
- Editor
- Technical Writers
- Tools Specialist
- Database Administrator/Repository Manager (this depends on the tool chosen)

When further investigation has been done, a detailed project plan can be submitted to management for approval.

### ***References***

- Hackos, J. (2002). *Content management for dynamic web delivery*. New York: John Wiley and Sons, Inc.
- Hackos, J. (2001). *Developing a content management strategy*. Training Course Workbook.
- Hackos, J. (1994). *Managing your documentation projects*. New York: John Wiley and Sons.
- Hackos, J., Rockley, A (1999). Single Sourcing White Paper. Single Sourcing Associates.  
<[http:// www.infomanagementcenter.com/pdfs/white\\_paper\\_001.pdf](http://www.infomanagementcenter.com/pdfs/white_paper_001.pdf)>
- GuruCom Customer Health Survey*. (2002). GuruCom Marketing, Internal information. *GuruCom Development Charter*. (2002). Internal Document.
- Rockley, A. (2001). Dynamic Content Management. *Intercom*. February: Pp. 28-32.

- Rockley, Ann. (2002). *Enterprise content management through single sourcing: White Paper*. Rockley Group.
- Rockley, A., P. Kostur, S. Manning. (2002). *Managing enterprise content: A unified content strategy*. Indianapolis: New Riders.
- Society for Technical Communication. (2002). *STC's 49<sup>th</sup> Annual Conference: Proceedings*. Virginia.
- Techwr-l technical writing email list from [www.raycomm.com/techwhirl](http://www.raycomm.com/techwhirl)

***Address for correspondence***

Dr Roslyn Petelin  
School of EMSAH  
University of Queensland  
St Lucia Q 4072  
r.petelin@uq.edu.au