Re-Forming Information: A Case Study in Teaching Content Encapsulation

Rebecca Matson
Computer Associates
Irving, TX 75039 USA
Rebecca.Sukach@ca.com

ABSTRACT

The challenges that face content management writers are not just tools, but the design principles and concepts underlying content management. This paper is a case study that explores the challenges of teaching writers both to create encapsulated content and to design that content within a user-centric approach to information design. The approach used in the class provided a fine-grained definition of information types to move the writers as far as possible from the traditional documentation approach. The insights presented in this case study are drawn from instructors' experiences in presenting a course in content encapsulation to technical writers, employed by a traditionally mainframe-based software company. Writer reactions suggested that the approach taught in the class assisted them in overcoming existing preconceptions of document design, user profiling, and content creation.

General Terms

Documentation, Design, Human Factors, Standardization

Keywords

Information design, information encapsulation, encapsulated content

1. INTRODUCTION

Creating documentation for multiple, diverse audiences is a challenging process, especially when writers are faced with documenting complex software products. Content management, a set of rules and procedures that regulate the creation, collaborative use, and delivery of information, may provide a way to look at the problems of providing multiple outputs to meet unique audience needs.

Content management software offers a ready solution to the mechanical problems of content delivery to multiple audiences across multiple media, yet it is the techniques of information design, supported by content management, with roots firmly in object-oriented programming, that may provide the most meaningful advances in this area.

Using content management structures and ideas, writers can reuse content and graphics from deliverable to deliverable, between related products and projects. This reusability of information objects allows greater freedom for, and assigns more importance to, the design of the final output. Through information reuse, writers can produce guides, help systems, quick references, and so on that take advantage of the format without repeating unnecessary information.

The challenges that face the writer are not simply learning new rules by which to create content, but also the design principles underlying user-centered design. The difficulty of designing systems without resorting to the book paradigm can prove challenging for writers trained to think of design as being linked to a printed page. Similarly, the creation of discrete stand-alone "chunks" for later assembly into one or more whole documents is difficult for writers trained to approach each document as a single continuous piece. Despite the prevalence of help systems and other more segmented formats, the "storybook" paradigm persists in technical writing as the ideal information presentation format.

Effective information reuse requires a more segmented approach to writing so that each created piece can be used in multiple deliverables without constant re-creation of common information pieces. Teaching the importance of information design as it relates to multiple presentations and stressing the importance of user profiling can underline this requirement. Information designers must learn to create an information space for users—a dimension of information customized to the user's information needs.

2. RELATED WORK

Encapsulated content can claim roots in a number of disciplines such as object-oriented design in computer science, information mapping in technical communication, information design, itself an interdisciplinary field, and heuristic design in usability engineering.

2.1 Information Encapsulation

Computer Science uses information encapsulation through object-oriented design [6]. Object-oriented design allows programmers to reuse bits of previously developed code. By carefully considering how each piece of code might be used in other parts of the system, programmers can reduce the complexity and overhead associated with a large software program. Reusing code saves both money and time since reusing code requires less resources than creating code. However, advocates of object-oriented design have experienced difficulties in teaching this approach to programmers—helping them overcome "global" thinking with "local" thinking [2].

SIGDOC'01, October 21-24, 2001, Santa Fe, New Mexico, USA.
Copyright 2001 ACM 1-58113-295-6/01/0010…$5.00. .

"Chunking"—creating pieces of text 7 (+/-2) sentences in length, a size based on Miller's "magic number" [10]—is a form of information encapsulation popular in the technical writing profession as part of information mapping, a method developed by Robert Horn [7]. Information mapping provides a structured way in which text can be created and categorized, by purpose, then organized into a flow.

Instructional systems design also uses information encapsulation to separate content from strategy [13]. Separation of different types of information allows readers to more quickly comprehend the information presented.

2.2 Information Design

Information design describes how to effectively present data to convey an idea. Drawn from such disparate fields as graphic art, cognitive science, hypertext navigation, writing, and more [8], information design can mean many different things depending on the audience. Guidelines for design vary widely from the minimalist approach to documentation [3] to cognitive graphic designs [16,17]. Even the choice of physical information transfer—book, audio, hypertext, film—can be approached in a variety of ways [1].

Information design includes helping users find the information they need and enhancing the comprehension of that information through visual aids, audience appropriate language, and affective considerations.

It has become increasingly important to include domain knowledge—the information required for the user to perform a task independent of the provided tool—in addition to the traditional tool knowledge in document design. The importance of the "why's" of the product as well as the "how's" guides what information must be provided [4].

2.3 Heuristic Design

Heuristic design is a method of using basic checkpoints or guidelines to guide the construction of a document or interface. Usability heuristic design [9] has proven an effective method of increasing the usability of interfaces cheaply and efficiently. By integrating heuristic design in the development process, potential problems can be avoided before the product reaches a point at which change becomes costly and difficult.

3. CLASS AND SYSTEM DESIGN

This case-study examines a class which teaches writers to create encapsulated content for use in a content management system. Each class contained 8 to 15 writers and was delivered over 3 days.

The training class was designed to introduce both experienced and novice writers to the idea of reusable information pieces, and their advantages over the book approach to documentation creation. To enforce the ideas and approaches required, no tool training was incorporated in the materials. Exercises were strictly paper-based to encourage attendees to begin applying the practices immediately, rather than waiting for a full implementation of the tool. The class included methods for user profiling, improving communication with developers, and identifying information sources as well as how to identify, write, and edit information elements.

3.1 Class Attendees

The course was presented to approximately 50 technical writers, employed by a traditionally mainframe-based software company. Class attendees' professional writing experience ranged from 6 months to 25 years.

Attendees were accustomed to delivering documentation in a variety of formats (PDF, HLP, HTML, BOO, JavaHelp, and CHM) with a single presentation. Documentation was traditionally written as a book then transformed into a different format with no reorganization of content.

3.2 Supporting Information

The class provided significant supporting information in addition to the methods of encapsulated writing. User profiling, information design heuristics, interview and communication techniques, and internal sources for support provided attendees with a context in which they could apply the information element ideas.

3.3 Content Management System Design

The content management system consists of a database of encapsulated content and a simple drag-and-drop GUI interface. The interface allows writers to create new elements and apply minimal formatting (bullets, numbered lists, and headings) as well as assemble documents, help systems, and other deliverables from the content items within the database. The database is searchable by index entries, created by the writer, or by full text.

Each content item is reviewed by an editor and, if applicable, a programmer, to verify accuracy and readability before it is fully entered into the database and available for reuse. Once the item has been entered, no changes can be made to the original item without the approval of all writers responsible for deliverables linking to that item to ensure that the changes will not adversely affect the linked documents. If the number of writers that must agree to that change exceeds a predetermined threshold, the editors and database administrator may instigate the change. If changes to the original item are not desirable, new instances of the item can be created. These instances must go through the same review process as new items to ensure consistent quality and voice.

4. CLASS CONCEPTS

A number of concepts were introduced during the course to allow the attendees to develop a common language base with which to communicate ideas. Encapsulated content items were named *information elements*. The parts of these elements were named *information atoms*.

4.1 Information Elements

Information elements are groups of information that are so closely related that the individual pieces of which they consist would not make sense separately, such as a single procedure (all the steps in creating a new file), a concept (an explanation of information elements), and so on. This approach builds on the information mapping method, increasing the specificity required by the writer in order to reinforce the discrete nature of information at a more basic level.

The class stressed that the size and length of an information element depends on the audience and the complexity of the topic. Initial information element types are defined in Table 1.

Table 1. Information Elements

Information Elements	Description
Code	Defines commands or code syntax and definitions.
Concept	Defines and explains an idea needed to effectively operate the program. Frequently includes required domain knowledge.
FAQ	Defines frequently asked questions that illuminate important aspects of the product.
Glossary	Defines unique terms used in the product that may be unfamiliar to the user.
Graphics	Provides a visual of an idea or process such as a flow chart or a system representation.
Item	Defines a part of an object, such as a button on a GUI or a command parameter.
Message	Defines text communications sent by the program, displayed to the user.
Principle	Defines short, important hints on what should or should not be done.
Procedure	Defines a series of steps you can perform in a given product.
Process	Defines and explains a series of events, stages, or phases.
Related Topic	Links information from one element to another.
Structure	Defines a physical object that can be broken into parts, such as a GUI.
Transition	Links two elements in a printed output, providing a smooth transition between them.
Code	Defines commands or code syntax and definitions.

These information element types were developed by a preliminary examination of a representative sample of existing company documentation (mainframe BOO documentation, distributed systems administrator guides, GUI help systems, and web pages). Information in these documents was broken down into stand-alone pieces and then examined for differences and similarities in the purpose of the information. That is, each information piece was evaluated for the type of content that it conveyed, rather than the literal content.

4.2 Information Atoms

Each element was strictly defined, providing a description of each part (called an atom) that was allowed within that element, whether that atom was required or optional, and the order in which that atom should appear in the element. For example, the following table defines the Code element:

Table 2. Code Information Element Atoms

Atom	Description	Required
Command Name	Identifies the name of the command.	YES
Command Syntax	Identifies the syntax for the command, including special characters.	YES
Item Element	Describes the function of the parameter.	YES
Command Parameter List	Identifies a list of items used by the parameter.	NO
Command Paragraph	Describes further detail for a parameter.	NO
Command SubParameter	Identifies a bulleted list of items used by a subparameter.	NO

The design of elements used in this system allows for recursion—that is, elements can be atoms of other elements. For example, in Table 2, an Item element can be an atom of the Code element. Similarly, a Graphic element might be an atom of a Concept element. This flexibility has the potential to introduce an additional layer of complexity in the change and configuration management of the system by potentially increasing the number of writers who must approve changes to the original element exponentially. This potential issue resulted in the decision that the editors and database administrator could allow change if the number of links to the element grew larger than ten.

4.3 User Profiling

The class reviewed techniques in creating user profiles using classifying, intuiting, and listening techniques [15]. Emphasis was placed on users having disparate information needs, and hence different information design needs.

For example, users and learners have very different information needs. Users understand the work that must be done—they have a thorough grasp of the domain in which they are operating. In contrast, learners do not understand the work that must be done—they do not have sufficient background in the domain in which they are operating. As a result, the optimal information deliverable may differ significantly [14].

Protocol-Aided Modeling (PAM) [15] was used to improve user profiling skills and reinforce the importance of taking user needs into account during both writing and design phases. PAM is a simple experimental method of checking the user profile developed by the writer. Taking a single page of grammatically correct documentation, the writer identifies all of the potential issues their user might experience while reading—terms that are not defined, graphics without explanations, and so on. Using a talk-aloud protocol, a user is asked to read the page, relaying the questions and problems they have, while the writer takes notes. The writer then compares the problems the reader experienced with the problems they predicted the reader might experience.

4.4 Information Design

The course separated the issues of information development (researching, writing, and editing content) and information design. Information design decides the organization and format (physical deliverable consideration such as HTML, PDF, and so on) of the final product. Writers in the class faced significantly different information design problems, from creating printed mainframe guides to providing JavaHelp for web interfaces.

Despite the disparate deliverables faced by the writers—from quick reference cards to printed manuals, from context-sensitive help to interactive web pages—a common set of heuristics can be used to improve and benchmark information design. General design principles such as "never create a jump link within a pop-up window" inform the creation of heuristics to design systems such as "Allow Undo." The following heuristics were used to evaluate information design problems within the class:

Table 3. Information Design Heuristics

Heuristic	Description
Allow Undo	Provides escape routes for users.
Be Consistent	Provide regular, repetitive language to support recognition.
Provide Feedback	Include consequence information for actions to confirm user expectations.
Provide Multiple Paths or Channels	Provide multiple ways to locate and recognize information.
Customize for User Needs	Provide multiple designs if user profiling indicates disparate user needs.
Support Scanning	Provide visual separation of ideas.
Use Natural Mappings	Use natural mappings provided by a common domain to reduce the data.

4.4.1 Allow Undo

Designers must allow users to undo or cancel an action. Mistakes made during navigation of online material, such as reaching an erroneous page, can cause users to become lost within the information space. For example, some web designers create "sticky" web pages—links that open new browser windows and close the original, preventing the user from using the back button if they find they did not intend to navigate to this page. This type of design thwarts any attempt by the user at self-correction, causing them to become lost and frustrated.

4.4.2 Be Consistent

Consistent design provides a familiar navigation structure throughout a document. Consistency makes comprehension of information easier by supporting recognition over recall [12].

Examples of consistency in documentation include using the same language to introduce list items or cross-references, using the same font and style for all procedural headings, and using the same format for definitions. Consistency allows the reader to quickly identify whether the type of information they are seeking is there without forcing them to read every word.

4.4.3 Provide Feedback

Designer must provide feedback to allow the user to monitor their progress. Feedback reduces the frustration of the user and allows them to confirm or deny any action they take [12]. Feedback in documentation can include descriptions of what should appear when an action is successful. For example, if the user enters the command "installme.exe", the writer may indicate that an installation screen appears. If the screen does not appear, the reader knows that something has gone wrong, and can take steps to correct the problem.

4.4.4 Provide Multiple Paths and Channels

Providing multiple paths to the information increases the ability of the user to locate and use the information. Similarly, providing media that delivers information on multiple channels (graphical, textual, and so forth) can supplement information losses due to information overload or cognitive difficulties. The effective use of layers, elements within a display, can create a synergistic effect—that is, two layers of information can transmit more signal if properly arranged [16].

4.4.5 Customize for User Needs

By understanding user perceptions, the information designer can more accurately gauge the tone and format for delivery. The needs of the user may not be identical to the needs of the writer [5]. Creating an effective information space requires knowledge of the user's domain knowledge, work environment, and information needs.

User profiling is an essential component in effective customization. Identifying the unique information needs of the user can only be accomplished when an accurate picture of the user exists.

4.4.6 Support Scanning

Most users scan the text to find the item they need to complete their task [11]. Every piece of information competes with the other pieces in a presentation, lessening the impact and visibility of the relevant information. Separation of content through design allows users to locate information quickly and accurately.

4.4.7 Use Natural Mappings and Constraints

Natural mappings are the relationships between objects and actions, between what is seen and the system state [12]. Designers can use these mappings to confirm a user's actions or prevent inappropriate action. Similarly, designers can use natural mappings based on common domain knowledge to reduce the amount of data presented without loss of information [15].

5. CLASS ATTENDEE REACTIONS

Class reactions were gauged through observations made by instructors, as well as verbal and written responses provided by attendees.

5.1 Consensus

Most class attendees showed enthusiasm and interest in trying the new approach. By breaking down examples in class, they were able to immediately see the improvement in readability brought about by encapsulated content. One attendee excitedly announced that "Maybe this document I'm working on isn't that complex—maybe it's just written badly!"

In addition, the freedom from formatting concerns and separation of writing and design concerns was noted as a significant advantage. “You mean I can just concentrate on writing and not worry about fonts and stuff?” was a common comment. Observations were made on how the new system would allow writers to concentrate on the area of creating documents that they found most interesting – design, indexing, graphics, writing, and so on.

Several writers commented on the PAM exercise, saying how it helped them understand how users felt when content was not encapsulated. The exercise underlined the fact that the user may not have the same information needs as the writer who created the piece. The users in this exercise were writers that had a different experience level than the testers. One participant said “I used to think that all writers would have the same information needs.” Another mentioned that “I was surprised—I thought if I wrote something so that I could understand it then anyone could.”

The supporting materials were commented on as being helpful in overcoming the barriers to thinking of information as small elements rather than a more holistic conceptualization. When writing about the design heuristics section of the class one writer said “It helped me understand how I could put all these little pieces together in new ways without just making a mess.”

5.2 Conflicts

Overcoming the belief that documentation is best presented as a storybook rather than discrete elements proved more complex and involved than originally anticipated. Opposition from writers came from the perceived complexity of initiating the method and ownership issues as well as the conceptual difficulty of thinking of encapsulated information.

5.2.1 Complexity

The method stressed that the writer must understand the product and the user in order to create or edit effective elements. Many writers believed they were incapable of understanding the product sufficiently in the time they had available because their products were “too complex.”

Access to more product training and more control over deadlines were offered as solutions to these objections. Most writers felt that, if training were provided, they could gradually move to content management. Many writers mentioned that the well-defined structure of the information elements—providing specific guidelines as to what could be used within each—made the process seem more “doable.”

5.2.2 Ownership

Since the elements in the database must be used as is, many writers expressed concerns that they would be unable to use elements created by others because they might not be up to their personal standards or style.

Standards enforcement was stressed throughout the class – both in writing style and construction of elements. All participants acknowledged that standards were important and should be followed, but seemed unconvinced that their peers would do so.

Distributed systems writers showed the greatest reluctance to use information elements created by others. Mainframe writers evinced the most acceptance. One mainframe writer who currently works on over 30 manuals said “This database could really save me some time—we get so many changes that have to be put in every manual. It’d be great to be able to do it in one place.”

The enforcement of a committee approach to changing existing information elements was offered as a solution to this objection. A structured editing organization (guideline on how and when to change an element) reduced many objections in this area as well as the ability to create new iterations of existing elements.

One unanticipated advantage noted by the participants was the ability to view someone else’s work to get new ideas on how to approach a particular writing problem. Since the database would provide the ability to search by element type and user type as well as content, writers felt they could get insight from their peers when faced with a difficult problem. “I can see myself searching the database for an element type to see how someone else did it—see how someone solved the problem I’m having.”

5.2.3 Design Challenges

The ability to think of their deliverables in different ways proved difficult for many attendees. When asked to name possible formats for their information to be presented, most thought in terms of books or books in online format.

Active websites, embedded documentation (documentation that becomes part of the interface through design improvements), tutorials, and quick reference cards were not mentioned and were greeted with some skepticism when listed by the instructors. Some writers felt that departure from the traditional document deliverables would be met by significant opposition from both users and programmers. “My users will never go for a tutorial or something like that—they don’t even like the PDF I have to send them now.”

Writers with an extensive mainframe background experienced more difficulty with the design ideas, such as offering alternate presentation methods based on user profiling, than those with a distributed systems background. However, most writers were intrigued by the flexibility of the database—allowing the writer to assemble existing elements into any type of flow or form they chose. It was indicated that increased support from management as well as availability of current research to use as a basis for change would help in mitigating responses from development. “Besides, if we do really good profiles, it’ll be what the customer wants—so how could they not like that?”

5.3 Improvements

All classes requested further explanation, supporting material, and exercises to be ongoing after the conclusion of the class. Many indicated that they intended to begin using the methods immediately, although the tool was not yet available. However, they wanted continuing guidance in deciding the size and composition of elements and the optimal design for their audience.

6. CONCLUSIONS

Thinking about information as discrete elements rather than a continuous stream provides significant challenges for traditional technical writers. The obvious benefits of improved readability and the ability to reuse information pieces can be used to leverage against contention due to ownership or complexity issues. By defining information elements with a high level of structure, writers are forced to think about the granularity of information in much greater detail—a process that can then be translated into the larger problems of design.

These challenges can also be overcome by demonstrating how the technique applies at all stages of documentation development—integrating the approach from inception to final product. By stressing how this method can be used throughout the process, writers were provided with a completely new coordinate system in which to operate—no longer approaching even the construction of a paragraph as a single piece.

Emphasis on user profiling to increase the awareness of different user needs that can be met by various presentation formats and designs underlines the ability of the writer to use content encapsulation to reuse information effectively. Recognizing that users have disparate needs that may not be met by a single deliverable encourages writers to think about alternate presentation methods and acknowledge the advantages of the approach. User profiling also increase the writer's sense of responsibility to the user by putting a face on the reader.

While the book paradigm is sometimes appropriate, with the advent of new presentation media, it is no longer the optimal approach to information design and delivery. Further research must be done to investigate new design methods and formats. Challenging writers with new concepts, we bring them into the idea of information space—a customized dimension that provides data designed for a particular user. Information reuse makes this design possible—providing the building blocks to create unique information spaces for each user type.

This preliminary study provided interesting insights into common writer reactions to a new approach to an old problem. Future studies will test the design heuristics developed for this class as well as probe optimal information element definitions. In addition, more research will be done on how elements are reused, observing the common characteristics of commonly reused pieces.

7. ACKNOWLEDGMENTS

Many thanks to Computer Associates for supporting the development and delivery of the class, to Robert Kennedy for his work on information elements and co-delivery of the class, and to Terry Sullivan for his invaluable input on this paper.

8. REFERENCES

[1] Augst, B. and O'Connor, B. (April, 1999). "No Longer a Shot in the Dark: Engineering a Robust Environment for Film Study" *Computers and the Humanities* 33, 345-363.

[2] Beck, Kent. (1989) A Laboratory For Teaching Object-Oriented Thinking. OOPSLA'89 Conference Proceedings, October 1-6, 1989, New Orleans, Louisiana.

[3] Carroll, J. (2001) *Minimalism* [WWW document]. URL http://www.gwu.edu/~tip/carroll.html (visited 2001 June 1).

[4] Draper, S. (21 Dec 1996). *Practical problems and proposed solutions in designing action-centered documentation* [WWW document]. URL http://www.psy.gla.ac.uk/~steve/MinMan2.html (visited 2001 June 1).

[5] Hackos, J. and Stevens, D. (1997). Standards for online communication. New York, NY: John Wiley & Sons.

[6] Hooch, G. (1991). Object-oriented Design with Applications. Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc.

[7] Horn, R. (1992). Clarifying Two Controversies about Information Mapping's Method. *Educational and Training Technology International* 29(2): 10917.

[8] Mazur, B. (April, 2000). "The Information Design Elephant?" *Design Matters*. *STC Information Design SIG*, April, 2000.

[9] Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, *Proceedings of the ACM CHI'90 Conference* April 1-5, 1990, Seattle, WA.

[10] Miller, G.A., (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Science*, 63, 81-97.

[11] Morkes, J. and Jakob Nielsen (1997) Concise, SCANNABLE, and Objective: How to Write for the Web [WWW document]. URL http://www.useit.com/papers/webwriting/writing.html (visited 2001 August 15).

[12] Norman, D. (1998). The design of everyday things. London, England: First MIT Press.

[13] O'Neal, A.F., Fairweather, P.G., and Huh Y.H. (1988). Instructional Systems Design. [WWW document]. URL http://www.whidbey.com/frodo/isd.htm (visited 2001 August 1)

[14] Quintana, C., Eng, J., Carra, A., Wu, H., Soloway, E. (1999) Symphony: A Case Study in Extending Learner-Centered Design Through Process Space Analysis. *CHI 99: Conference on Human Factors in Computing Systems*, May 19-21, 1999, Pittsburgh, PA.

[15] Schriver, K. (1997). Dynamics of documentation design: creating text for readers. New York, NY: John Wiley & Sons.

[16] Tufte, E. (1990). Envisioning information. Cheshire, CT: Graphics Press.

[17] Ware, C. (2000). Information visualization : perception for design. San Francisco, CA: Morgan Kaufmann.