

Designing documentation
for different user profile needs and expectations:
process and purpose

Research Article

Laure-Amélie Le Stang ~ Camille Riche ~ Alizée Rodriguez

M2 Conception de Documentation Multilingue et Multimédia

~

Paris 7, Université Denis Diderot

~

2013/2014

Table of Contents

Abstract	3
Introduction - What are the user profiles and why is it necessary to define them?	3
I. Solutions and examples of screen-based documentation engineering in Dassault Systèmes and Thales	4
A. Dassault Systèmes.....	4
Context	4
Solution	4
Critical analysis	5
B. Thales Global Services	5
Context	5
Solution	6
Critical analysis	6
II. The solutions we foresee.....	6
A. Graphical indicators	6
B. Structured documentation and DITA	7
C. Databases and PHP	7
D. Help authoring tools and conditional tags [Latthida Traymany]	8
Conclusion - What is the best method to implement user profiles in technical documentation?.....	8
Bibliography and Webography	9

Designing documentation to address different user profiles' needs and expectations: process and purpose

Abstract

This research article aims at studying how to create different levels of usage using structured documentation, once user profiles have been defined. From the review of different articles written by technical writing experts, to the analysis of various use cases, we will examine several solutions to document functionalities or products for a complex and diversified audience. In the first part of our paper, we will analyze existing techniques experienced in our companies, such as single sourcing and filtering metadata. In the second part, we will focus on a specific use case, and present the challenge of a company which is trying to optimize costs, improve the quality of its documentation, and at the same time, address different user expectations.

In conclusion, we will see there is not one perfect solution, but some useful techniques, each adapted to a particular issue technical writers must identify before selecting the right one to fix it.

Keywords: documentation levels (of usage), complex audience, user expectations, user profiles, audience analysis, structured documentation, single sourcing, filtering metadata, documentation cost optimization

Introduction - What are user profiles and why define them?

Nowadays, people are more and more aware of technologies. They can use a computer and master web surfing more easily than they could do ten or fifteen years ago. Thus, there are no longer two distinct categories of users - experts and novices - but a wide range of users, from the most experienced ones to those discovering new technologies (aged people), including the new generation who grew up in this environment. Therefore, the user panel has been expanding over the last decades and the documentation must be adapted to answer those new issues.

According to Paul V. Anderson – Director of Writing Across the University, at Elon University NC – a complex audience is a “group of people [...] reading [a document] from many perspectives.” This affirmation relies on the personae concept, a user-modeling technique developed by Allan Cooper in *The Inmates are Running the Asylum* (1999). Basically, this approach works at describing real people and their needs, rather than stereotypes. It focuses on describing people from a sociological point of view – for example, thinking “*our users are students*”, does not tell you who they really are and what they really need. Before the introduction of this concept, one set of documentation addressed one predetermined user profile, which meant: increased printing maintenance and translation costs, as well as redundancies between documents because there was no gathering of common knowledge. To reduce these costs, companies tend to pay more attention to the diversity of users profiles when they write their documentation.

As such, this technique aims at building a comprehensive, user-centered documentation for a complex audience, that is an audience which has: different roles, needs, expectations and levels of expertise. The stakes are to meet the needs of each user profile and thus to increase the usability and the efficiency of the documentation.

Based on those definitions, we chose to analyze the subject from the reality of two big companies, constantly trying to innovate in terms of documentation engineering: Dassault Systèmes and Thales. We also focus on various techniques to improve the efficiency of a documentation targeting different user profiles.

Given the increasing importance of screen-based documentation in the technical writing world, this research article will focus on this type of documentation engineering in order to propose an overview of various solutions that match modern expectations.

I. Solutions and examples of screen-based documentation engineering in Dassault Systèmes and Thales

A. Dassault Systèmes

Context

In the 1970s, Dassault Systèmes has developed 3D Computer-aided design software named CATIA. It is made up of a series of different applications, each one dedicated to a business activity (such as product design). As this software includes a large number of applications, there is a massive amount of knowledge formalized in the user documentation. To fit this wide range of use cases, documentation has thus to be segmented to avoid the so-called “infobesity”.

In order to organize information, documentation targets three different user profiles over three predefined guides:

- Installation guide
- Administrator guide
- User guide

Before 2014, Dassault Systèmes’ documentation was already segmented: the common base of knowledge was written in guides, and specific client information (for example, particular context) was stored in a knowledge base. However, these multiple materials made it complicated for the reader to seek information, as difficult for the Research and Development department to maintain. Today, Dassault Systèmes aims at gathering all this information in one material.

Solution

Based on those observations, to increase the usability of its documentation, Dassault Systèmes’ Research and Development department offers a system of filters to let readers access the content. With this solution, the whole content is made available to readers, who have to interact with it to find the information they are looking for. This technique means readers are active in their search for information. Such a type of organization is called *progressive disclosure*. It is efficient for massive documentation and relies on reducing cognitive workload, that is, the amount of information displayed on one screen at the same time. And information is then sequenced with one particular technique: the tooltip.

When an item calls for more detailed information, a tooltip is made available by moving the mouse over the element. Then, readers can either consult the tooltip if it is relevant for their use case, or continue reading without losing track of the content.

Critical analysis

The most obvious flaw is that tooltips offer very little space to write additional information. For an advanced user, that needs to read all additional information available in the tooltips – going from one tooltip to another – is a genuine intellectual gymnastic that requires concentration not to lose the track of reading. Worse, it seems to be optional information, more noise that signals, given the fact that it appears in tooltips.

From this point of view, we can ask ourselves if tooltips really are the best solution to manage user profiles. Indeed, tooltips can rather be used to refer to another document, or give a short definition of a term, than to provide additional small documentation in it. In other words, if terminology is the criterion that differentiates user profiles, then tooltips do offer an interesting alternative to multiple materials.

Moreover, progressive disclosure still seems an interesting technique for improving usability, because it reduces cognitive workload. Since optional or additional information is moved out of the main user interface, the minimum information for performing the task is displayed, and the user understands faster because he/she does not feel drowned in information, leading to better results. This design technique is also largely used for e-Learning development, for example. To be more specific, the information is sequenced into layers: the surface containing the concept and reference topics, and the deeper layers containing the task topics for advanced users.

Despite the efficiency of tooltips, something is missing in Dassault Systèmes' approach to progressive disclosure: how to implement it? From this perspective, other solutions [Tom Johnson's] are worth exploring, such as hotspots. This technique consists in including expandable task topics in drop-down hotspots on the page, below the concept topic, fully displayed. This technique reduces the amount of information on the screen and lets the user decide if he/she wants to read the tasks topics by the labels given to the hotspots. It could be a solution to develop.

B. Thales Global Services

Context

The question of defining a common documentation based on a unique media is relatively new in Thales. In the past, Thales had a simple approach: one user profile corresponded to one documentation set, which induced high costs in terms of printing – as mentioned in the introduction above. But the equation is in reality a good deal more complex:

1 item of information + n user profiles = n Group-level documentations + n Entity-level tailored documentations

Because Entities are entitled to tailor Group documentation for local needs (different operational context, different roles...), for the Group, this means that the objective is to write one “common” documentation with a low level of granularity that the Entities can raise for their specific organization. The challenge is thus to write content that is specific enough, but not too specific, to address all the Entities of the Group.

Solution

To answer those particular needs, the final deliverable was composed of a unique material with a system of filters by role. Once again, the user is active in his/her search for information, because he/she needs to specify his/her role before accessing the content. However, the total content remains completely available if the user wants to read it. This technique avoided compartmentalization of documentation. Concretely, such an approach is inspired by single sourcing because all modifications of the content will occur in one place. This makes the initial equation easier:

1 item of information + n user profiles = 1 multi-level Group documentation + n Entity-level tailored documentation

Critical analysis

We saw this system of filters by role is interesting because it improves the relevance of the information for the user. But it does not seem to be the most appropriate solution in Thales' context, when most users have at least two functions involving various activities. And with this system, the user can only select one profile before accessing the content. As such, users who want to read about both function 1 and function 2 activities, must read the documentation twice. This is time-consuming.

Considering this aspect, are filters by role the best solution to manage user profiles?

It is true: filters by role are useful when user profiles are strictly delimited. Nonetheless, there can be intersections between them, but the more specific the content, the easier the technique to implement it in the documentation.

II. The solutions we foresee

Having described the context of both our companies, let us focus on multi-profile documentation from the point of view of any company trying to optimize costs while keeping a highly-accurate documentation that answers users' needs. We already discussed progressive disclosure and filtering by role: are there any other existing solutions, or can we imagine some more? To answer those questions, on one hand, we asked for opinions from professional teachers, while on the other, we tried to solve the issue by defining our own solutions, based on the observations we made and the lessons we attended this year.

A. Graphical indicators

The solutions we studied so far were techniques supported by tools. But what about graphical indicators? Visual elements identify the relevant sections for the user quicker than a short description would [Marc Swanson]¹. Cookbooks, for example, use this system to identify the difficulty level of a recipe. So, why not transpose this technique to technical documentation?

It is easy to implement when the criterion that differentiates the user profiles is difficulty, because it echoes a code that we all know: green is easy, yellow is medium, red is difficult. But when it comes

¹ Marc Swanson: Senior Technical Writer at Capsule Technologie, and Teacher at Paris 7, University Denis Diderot.

to more complex differences between the user profiles (for example, a use case), these codes can be tricky to implement, as images alone need to be defined in graphic standards as not to be confusing.

B. Structured documentation and DITA

The DITA documentation model seems a good candidate for our problem, because it was designed with the intention of supporting both coarse and fine-grained reuse from the start [Rory Keeley]². DITAVAL files are used to filter chunks of information by their attributes, in order to display some chunks and hide others for publication.

If we go deeper into the technical aspect of this process, we must talk about metadata. Basically, these are data used to describe other data: they can be used to enforce traceability of modifications, specify content format... And – something of particular interest – to specify the target audience of a chunk of information with the <audience> attribute.

What is good about this solution is that filtering can be applied at the level of words, sentences, sections or even topics. This feature offers a variety of possibilities for the writer, because the metadata attribute can be placed anywhere in the content.

Once all attributes have been defined, it is time to publish the document: the technical writer builds a DITA map containing all topic references, specifies which topics need to be displayed for audience A, and uses a transformation engine called XSL-T to generate the output format (several transformation scenarios are available and include WebHelp and PDF.)

Then, the technical writer restarts the process for audience B: specification of displayed topics, transformation, publication, and so on for each audience of the document.

This is of course time-consuming, because attribute labels must be clear and unique to avoid any mismatch. Nonetheless, once everything is well defined, the publication process is a lot faster: the writer skips directly to the transformation step with XSL-T and republishes the document, which answer our cost constraints.

C. Databases and PHP

Let us now connect the classes we attended this year with our problem. Our programming classes taught us to modify the interface after users perform an action (such as hovering the mouse or clicking an item). This would be interesting because users would again be active in their search for information, and only relevant information would be displayed. So the question is: how to implement that? PHP, MySQL and metadata have a lot to offer. Let us imagine:

Before accessing the documentation, readers could “log in” by selecting personal information (such as user role, activities, tasks) in pre-determined drop-down lists managed by the technical writer. Once this information is collected, it would be analyzed in a database with MySQL to only display the relevant content associated to the user’s role, activities and tasks. This three-stage process (log in, database analysis, displaying of the results) would of course be costly to implement because it requires advanced skills in PHP, MySQL and XML to attribute metadata to each information chunk as well as maintaining the database. But it would implement a customized, accurate documentation answering the readers’ needs.

² Rory Keeley: Documentation Development and Training Manager at IBM, and Teacher at Paris 7, University Denis Diderot.

D. Help authoring tools and conditional tags [Latthida Traymany]

Of course, few people can say that they manipulate PHP, MySQL and XML easily. So, why not considering a solution less demanding in terms of skills? Help Authoring tools such as RoboHelp, Flare or AuthorIt offer an interesting alternative because they support a technology called conditional tags. Let us illustrate this solution with a case study:

A software evolves with its documentation. All information chunks relative to version 1 can be inserted in <v1> conditional tags, and all information chunks relative to version 2 can be inserted in <v2> conditional tags [Latthida Traymany]³. This enables the technical writer to go back and to republish <v1> version anytime if there is a problem with <v2>. This flexible system enables the technical writer to choose what chunks of information need to be displayed for a specific audience, which answers our problem. Besides, to avoid confusion, the writer can remove some conditional tags after publication.

As for DITA, the preparation step with tag definition is also time-consuming, but the publication process is equally quick: the technical writer skips directly to the generation step and republishes the document, which also answer our cost constraints. For both solutions, initial software costs might be high, but computer-aided tools are easier to use than DITA.

Conclusion - What is the best method to implement user profiles in technical documentation?

At this point, a question remains: from the review of all these solutions, is there one that clearly stands out from the others? It will not be a surprise to anyone if the answer is: THAT DEPENDS! Depends on what? Actually, it first depends on an obvious point: the type of company the technical writer works for. It goes without saying that a big company can afford a lot more than a start-up. But this was only the financial aspect.

Then, another criterion to take into account is the documentation type. Help Authoring tools offer very good solutions for online helps, while metadata filtering are more transverse and apply well to manuals.

Moreover, it also depends on the criteria that differentiate user profiles: as we previously explained, graphical codes are appreciated for representing difficulty levels, while for other criteria, DITA provided most solutions.

We also discovered an interesting principle: progressive disclosure. This solution should be taken advantage of for screen-based documentation. And this is the biggest advantage: information can be presented layer after layer to enhance usability. And this principle can apply to all these solutions.

At last, we could almost say: there is no single perfect solution. Each technique we presented along our research article could be applied as a good solution so long as it fits the specific issue it is made for. Indeed, every single solution has its advantages and its drawbacks. Nonetheless, if we are able to clearly identify the purpose of our documentation and the problem it draws, we should be able to pick the right technique to fix this issue.

³ Latthida Traymany: Technical Writer at Sidetrade, and Teacher at Paris 7, University Denis Diderot.

Bibliography and Webography

- Paul V. Anderson (2010) “Chapter 3: Defining Your Communication’s Goals” in *Technical communication: a reader-centered approach*, p. 74
- Allan Cooper (2003), *The origin of personas*,
Founder & President of Cooper Design and Strategy (San Francisco)
<http://www.cooper.com>
- Swapnil Ogale (2013), Justifying Single Sourcing, a Case Study,
Technical Writer based in Australia
<http://www.enjoytechnicalwriting.com>
- A. L. Ames (2012) *Improving the User Experience by Applying Progressive Information Disclosure*,
IBM Senior Technical Staff Member, Information Experience Strategist and Architect
<http://fr.slideshare.net>
- Applying Progressive Information Disclosure to Online Help Navigation (2012) T. Johnson
technical writer, California
<http://idratherbewriting.com>
- Rory Keeley, Information Development Manager at IBM, and Teacher at Paris 7, University Denis Diderot:
XML and Single Sourcing university lessons (Master Degree: Conception de Documentation Multilingue et Multimédia)
- Virginie Ahrens, Content strategist, Learning and Knowledge solutions at IBM, Science Po Paris, and Teacher at Paris 7 Université Denis Diderot:
Information Architecture university lessons (Master Degree: Conception de Documentation Multilingue et Multimédia)
- Marc Swanson, Senior Technical Writer at Capsule Technologie, and Teacher at Paris 7, University Denis Diderot.
- Latthida Traymany, Technical Writer at Sidetrade, and Teacher at Paris 7, University Denis Diderot.