

Structured documentation: principles and advantages, formats and tools

March 2019

Historical background as a foreword

When mass-market micro-computing was born in the 1980s, centuries of book printing had already set up the layout and grammar standards that were designed as a consequence of Gutenberg's invention of the printing press. These conventions were used by the American IT industry leaders (IBM, DEC, Sun, Apple, Microsoft) for their first printed user manuals and other reference guides, quick reference cards and marketing leaflets. The zero paper trend in the 1990's and the Internet age then led to pdf and to HTML. That's when the first shift occurred from paper to electronic documentation, with needs for new formats, based on the separation of markup and text contents, leading to new information design.

Towards Structured documentation

In the process of defining the first standards ODM (an object oriented document structure) and SGML (the first document markup language), comprehensive studies on the nature of technical information were conducted, especially in the US.

Among other things, this led to the commercial offer "Information Mapping", which is a very detailed method for breaking down information into modules, widely based on tables and lists. It did not invent technical writing, but it gathered information into a consistent model that certainly helped many technical writers.

Software editors produced DTP software like PageMaker, InDesign and QuarkXpress, much more powerful than word-processors. Macintosh and its WYSIWYG revolution took over most of the professional DTP market.

Information modules were then used as the building blocks of online help. Followed versioning applications, to keep track of the versions and also to produce documents tailored to the various models of the same product sold on the market. It was also the time we moved from Systran's disastrous automated translation to translation memories based systems. The notion of controlled languages with a limited set of precise technical words dates back to this time as well.

Principles and advantages of structured documentation

Structured documents enable us to deliver documents for multiple media and for different audiences. The fact that they use the same unique source modules guarantees consistency and simplifies updates. The information modules use a set format, designed to make contents fit on many different media, as well as in various documents.

A CMS stores all documents in a central repository. It keeps track of changes and guarantees only one person at a time updates a document. It also eases maintenance of modular documents and generation of specific versions.

The format for documents is now increasingly XML. Specific XML editors provide the writer with all the tools he is now used to, like spellcheckers and grammar tools. Some provide context dependent text suggestions. They also offer a great variety of output formats.

For detailed information on migration to structured documentation, see the following well documented paper <https://comtechp7.hypotheses.org/files/2015/11/2013-Moving-to-structured-documentation.pdf> (**Comtech**, based in Denver (Colorado) is a company owned by JoAnn Hackos, and it delivers trainings on DITA).

Formats and tools

Today, in the era of big data and with a mass market of smart phones, information processing has reached a new dimension. Navigation is also changing again on the smart screens of our phones, deeply impacting in turn the layout of websites. Artificial Intelligence and robots are also new actors on this scene. Also, more visual information is finally available.

All this gives **DITA** (OASIS), which originally was an IBM open standard, and later the **DITA Open Toolkit** (Apache) a new momentum. Either of them enable many companies to define their own documentation processes and they are well known in the technical communication world.

The Open Toolkit uses a rich document format that can be used with several XML editors and CMS, and it offers many output formats like XHTML/HTML5, pdf, Markdown, normalized DITA, etc. It can be used with several XML editors (see www.dita-ot.org).

Adobe Framemaker 2019 is HTML5 responsive and supports XML and DITA (see www.adobe.com/fr/products/framemaker.html).

MadCap Flare (for Windows only) is a newcomer (2010). It supports DITA and also interfaces with WebHelp and Framemaker (see www.madcapsoftware.com/products/flare/).

oXygen and **Xmetal** are other DITA-compatible XML editors.

Outside the DITA constellation, **Scenari** is an open source multimedia documents design suite for Linux, Microsoft Windows and macOS. It was designed by Compiègne Technological Institute, and it is gaining momentum. It is very powerful and open, and it can be tailored to your needs. Like DITA, it is not an off-the shelf product (see <https://ubuntulandforever.blogspot.com/2009/05/scenari-is-open-source-application.html>).

Remaining non-technological quality issues in technical communication

Over the past 30 years, the focus has been set on tools and processes, but the design process of targeting contents to its audience seems to have been lost on the way. It periodically comes back under new terms like “usability”, “information design” or now “user experience”, but it requires shifting the focus to users, if they are to be effectively included in the design process itself.

How? By meeting and interviewing users, and watching them at work before designing the document structure. Gaining more quality feedback in a systematic way is also possible with little effort, using the existing departments of sales, training and hotline. Further, the online format also permits users to give instant feedback, a kind of short loop to improve the quality of the provided information.

Last, testing is widely used in most industrial processes but not in user information, although it is very simple to implement. Why not just do it and measure how good existing documents really are? **Ginny Redish** (USA) (see <http://www.redish.net/about-ginny>) documented extensively documentation testing and **Ulf L. Andersson** (Sweden) invented the concept of “humanware” (see <https://www.fnac.com/livre-numerique/a11325516/Ulf-L-Andersson-Humanware-Practical-Usability-Engineering>). This is not of technological nature, but relates to the information users need. You can only obtain it from the users themselves, by placing them in the center of your documentation process. Today, we tend to call this collective intelligence.

Jean-Paul Bardez

<http://www.jean-paul.bardez.com/recherche/>