

Documentation Tools to produce Articles and Presentations

Gerrit Muller

Philips Research
IST-SWA-AME
Prof Holstlaan 4 (WL01)
5656 AA Eindhoven
The Netherlands
gerrit.muller@philips.com
<http://www.extra.research.philips.com/natlab/sysarch/>

version: 1.0

22nd December 1999

Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

1 Introduction

The Gaudí project will produce a large number of articles and presentations about System Architecture. The first 3 months of the project were used for exploration of reader requirements, potential contents, distribution media and support tools.

This article describes the requirements, the choices and the underlying design of the tools to fulfil the quest of making the art of system architecting more accessible and understandable.

2 Requirements

The requirements for the tools are driven by the keydrivers of the author. Figure 1 shows the keydrivers and the first level of derived drivers. In the following subsections the requirements are enumerated as derived from the derived drivers and of course in line with the keydrivers themselves.

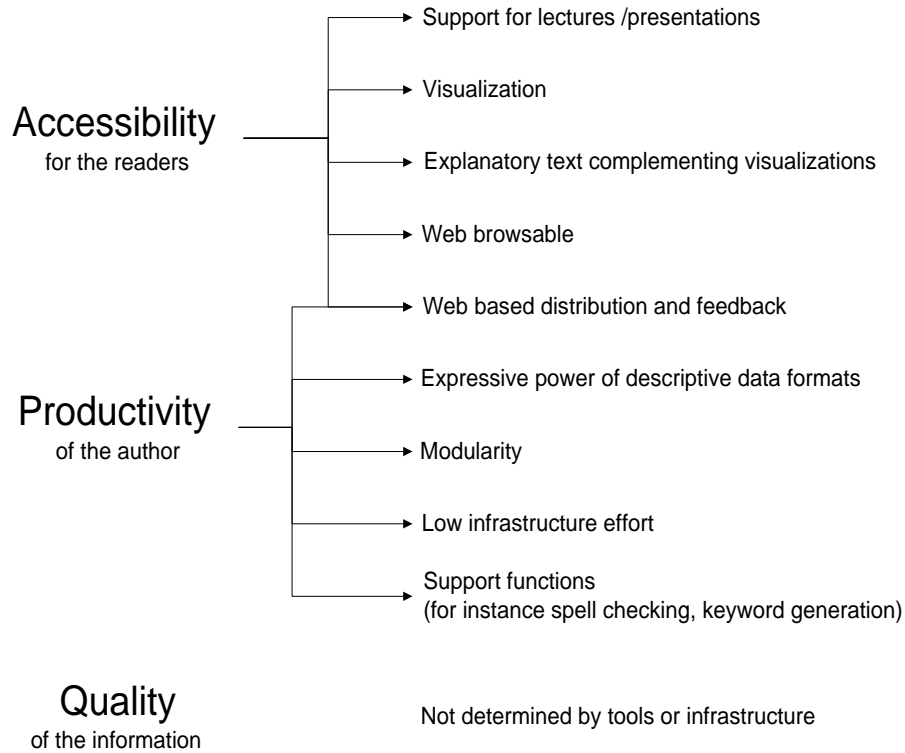


Figure 1: The keydrivers of the author using the documentationtools

2.1 Requirements w.r.t. the output of the tools

- Platform independent (1)
- Web enabled (2)
- High quality on paper (3)
- Fast scrolling enabled if viewed on monitor (4)
- Presentations enabled for electronic projection (5)

2.2 Requirements w.r.t. managing the documentation and the tools

- Modularity of information (6)
- Minimal redundancy, limited use of copy paste (7)
- Separation of information and presentation (8)
- Low effort to create and maintain the tools (9)

2.3 Requirements w.r.t. source data

- Standard Format (10)
- Enabling version management, a.o. by supporting differencing (11)
- Platform Independence (12)
- Support for Rich Graphics, i.e. maintaining the structure of the information (13)
- Bibliography support, including symbolic citation support (14)
- Automatic generation of figure, table and page numbering with symbolic reference support (15)

2.4 Requirements w.r.t. Use of the tools

- Easy (16)
- User actions should have an intuitive and predictable result (17)
- Customizable, f.i. minimize user actions for frequent operations (18)
- Support for Graphics (19)

2.5 Requirements w.r.t. Design and Implementation

- Platform Independent (20)
- Extendability (21)
- Scalability (22)

2.6 Future Requirements

- Spell Checking (23)
- Keyword Generation to support web search engines (24)

3 Design

The design is based on a collection of tools and fileformats which are integrated by a lightweight tool and a limited number of guidelines and conventions. The tools selected at this moments are:

- LaTeX, text source files .tex
- Visio, Visio proprietary source files .vsd
- NikNak as postscript to pdf convertor, .pdf as standard output format
- Tex4ht as html generator, this does not work properly yet

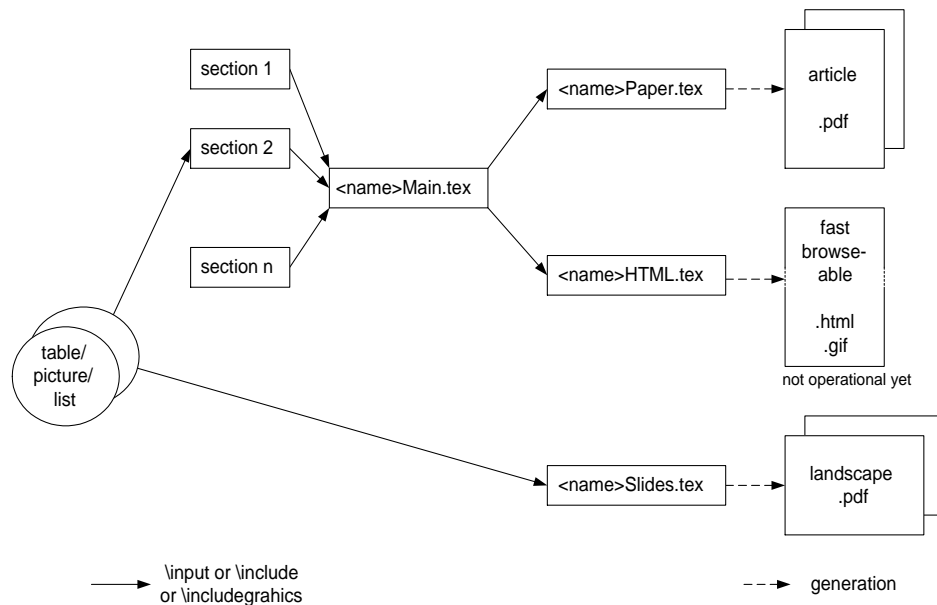


Figure 2: Structure of the source information in relation to the generated output

Figure2 shows the structure of the source information and the translation of the source information to the generated output formats. It shows a number of guidelines being used:

- Modularization of the source information
- 1 (One) image, table, list or cohesive text module per file
- Separation of information and presentation
- Use a rootfile per output type (Slides, paper article or html article)

- Structure of an article following a standard layout, see figure 3

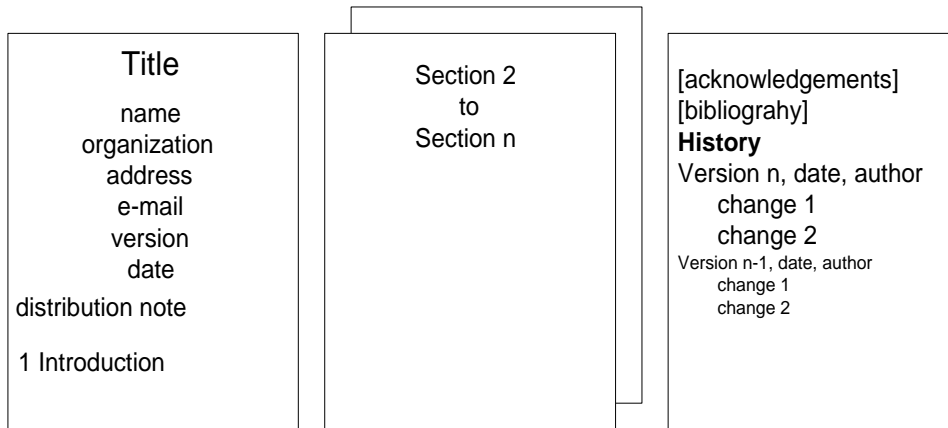


Figure 3: Typical layout of an article on paper

The typical layout of an article, as shown in figure 3, is supported by a number of self made LaTeX commands described in section ??, and follows of course the predefined conventions captured in LaTeX style files.

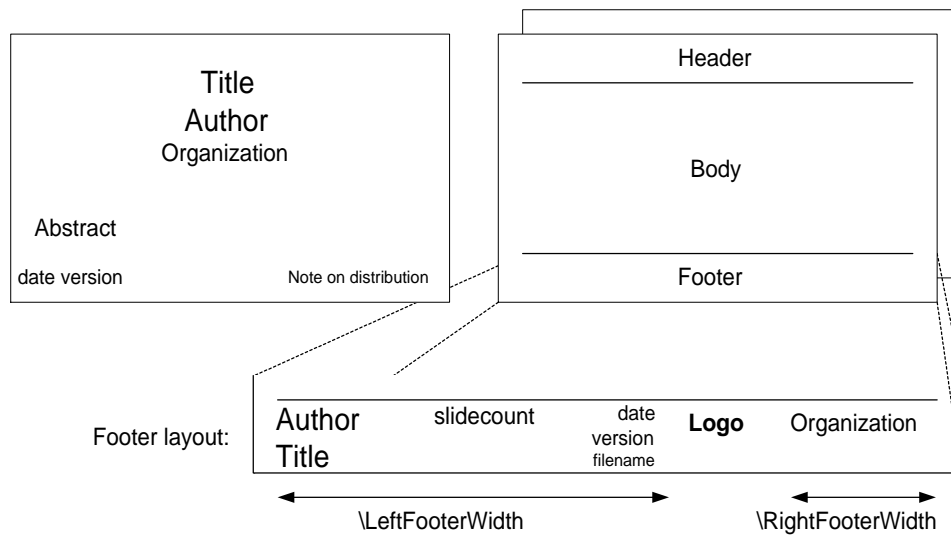


Figure 4: Typical layout of a presentation on slides

The same holds for the typical slide layout, which builds on the standard slide environment of LaTeX.

Figure 5 show the entire chain of transformations from source file to output file, including the intermediate filetypes. Note that LaTeX generates much more

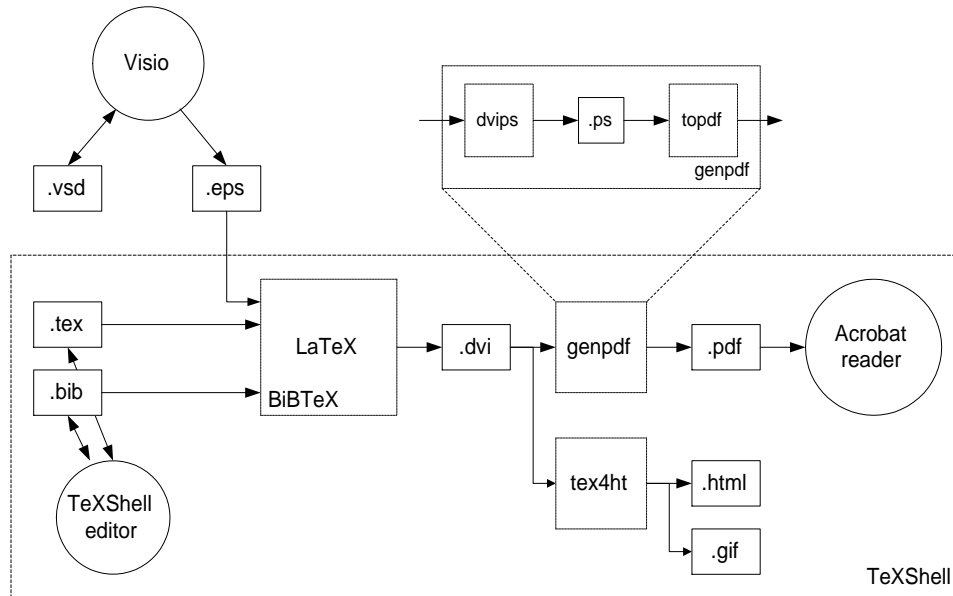


Figure 5: The different filetypes and the transformation chain from source to output

intermediate files which have been left out of the diagram for simplicity reasons.

Today the intermediate steps to dvi and ps files is not the most logical design. New LaTeX tools exist, such as pdfTeX which transform LaTeX input directly into pdf files. Unfortunately the graphics support does not seem to be at the same level as the chain LaTeX→dvi→ps→pdf. This part of the design might change in the near future.

4 Justification

The tool selection process was started by means of a micro-enquete: several colleagues were asked which tools they would recommend to produce a significant amount of articles to be integrated in a book, with many pictures, together with the rationale of this advice. This resulted in 3 main alternatives, which were studied during the feasibility phase:

- Combination of Word and Powerpoint
- Framemaker
- LaTeX

The combined use of Word and Powerpoint is today's default choice, due to the penetration of PC's loaded with these tools. The main disadvantages of these tools are:

- Poor support for picture generation and maintenance
- Poor support for modularity
- Not platform independent
- Source format not published
- Many non trivial automatisms
- the lack of separation of information and presentation

The poor picture support can be overcome by using a specialized tool like Visio for drawing pictures. However the next disadvantage pops up: the clumsy embedding mechanism.

Framemaker improves on nearly all the disadvantages of Word and Powerpoint. The drawing UI used to have a number of inconveniences. Major remaining disadvantages are:

- the marginal support for modular work
- the low level of automation support.
- the lack of separation of information and presentation

This alternative was not studied in depth, so some of the disadvantages might have less impact than expected by me, while others such as the source format might appear as a problem.

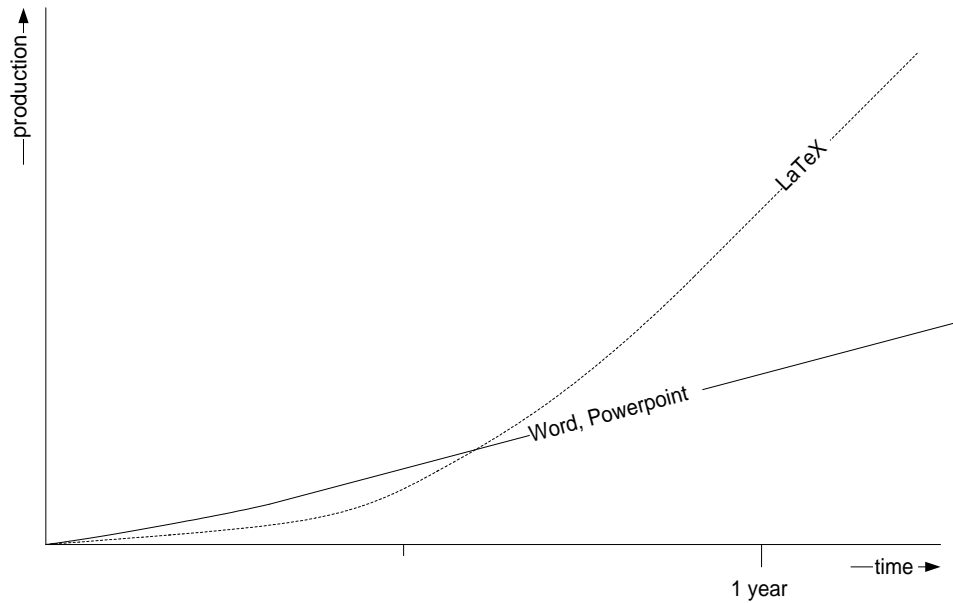


Figure 6: Comparison of expected productivity of LaTeX versus Word

Figure 6 shows the expected difference in productivity between the chosen solution of Latex in combination with Visio versus the Word and Powerpoint alternative.

Note that the Latex plus Visio alternative does not fulfil all requirements, especially the Visio tool is violating both the platform independence of the tools as well as the openness and manageability of the source format. On top of that I am not yet capable of automating Visio tasks, although according to collegeas this ought to be possible. The online Visio help does not say a single word about this functionality.

History

Version: 1.0, date: december 22 1999 changed by: Gerrit Muller

- update of frontpage and footer layout
- added description of micro-enquete

Version: 0, date: october 13 1999 changed by: Gerrit Muller

- Initial Version.