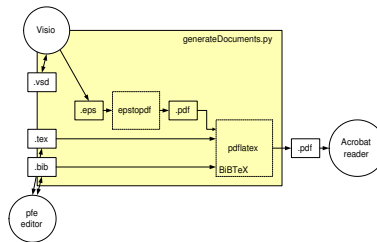


Gaudí Documentation Tools to produce Articles and Presentations

-



Gerrit Muller

Embedded Systems Institute

Den Dolech 2 (Laplace Building 0.10) P.O. Box 513, 5600 MB Eindhoven The Netherlands

gerrit.muller@embeddedsystems.nl

Abstract

This presentation shows the requirements, the choices and the underlying design of the tools to fulfil the goal of the Gaudí project to make the art of system architecting more accessible and understandable.

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.

All Gaudí documents are available at:
<http://www.gaudisite.nl/>

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.

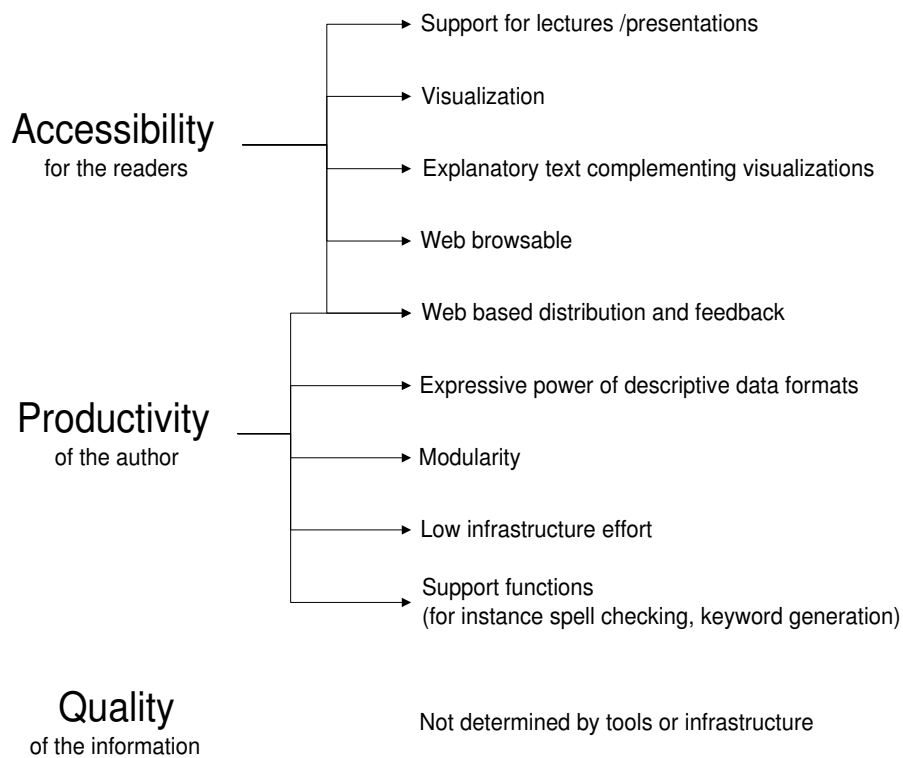


Figure 1: The keydrivers of the author using the documentationtools

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)
- Automated creation of aggregated documents, such as books and course modules and binders (8)
- Separation of information and presentation (9)
- Low effort to create and maintain the tools (10)

2.3 Requirements w.r.t. source data

- Standard Format (11)
- Enabling version management, a.o. by supporting differencing (12)
- Platform Independence (13)
- Support for Rich Graphics, i.e. maintaining the structure of the information (14)
- Bibliography support, including symbolic citation support (15)
- Automatic generation of figure, table and page numbering with symbolic reference support (16)

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

- Easy (17)

- User actions should have an intuitive and predictable result(18)
- Customizable, f.i. minimize user actions for frequent operations (19)
- Support for Graphics (20)

2.5 Requirements w.r.t. Design and Implementation

- Platform Independent (21)
- Extendability (22)
- Scalability (23)

2.6 Future Requirements

- Spell Checking (24)
- Keyword Generation to support web search engines (25)

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:

- pdflatex, text source files .tex, .pdf as standard output format
- Visio, Visio proprietary source files .vsd, .eps as output
- epstopdf as postscript to pdf convertor

Figure 2 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)

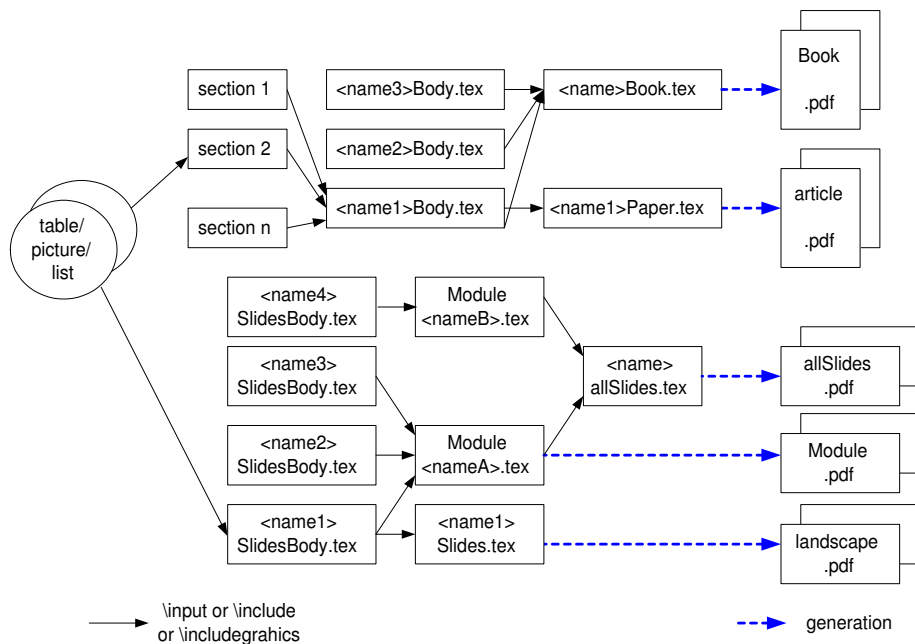


Figure 2: Structure of the source information in relation to the generated output

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

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

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

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

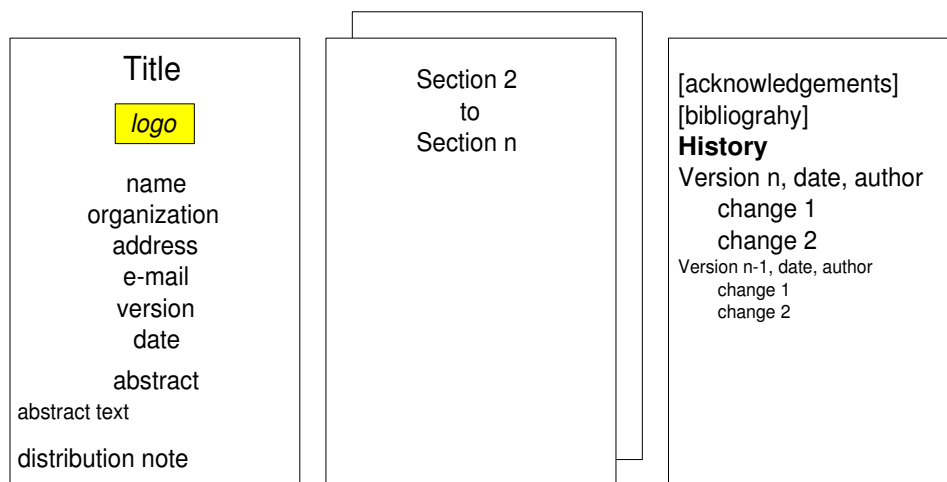


Figure 3: Typical layout of an article on paper

- 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.

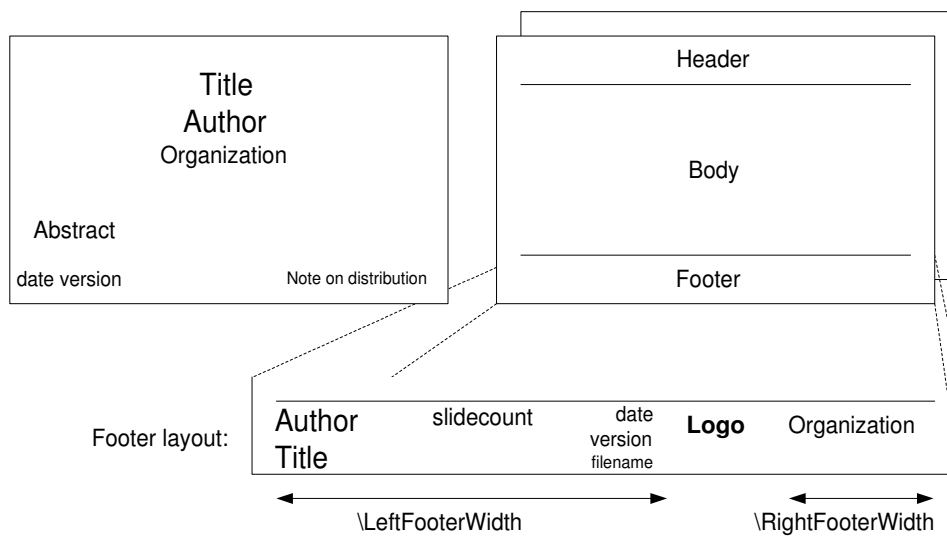


Figure 4: Typical layout of a presentation on slides

- 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 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.

References

- [1] Gerrit Muller. The system architecture homepage. <http://www.gaudisite.nl/index.html>, 1999.

History

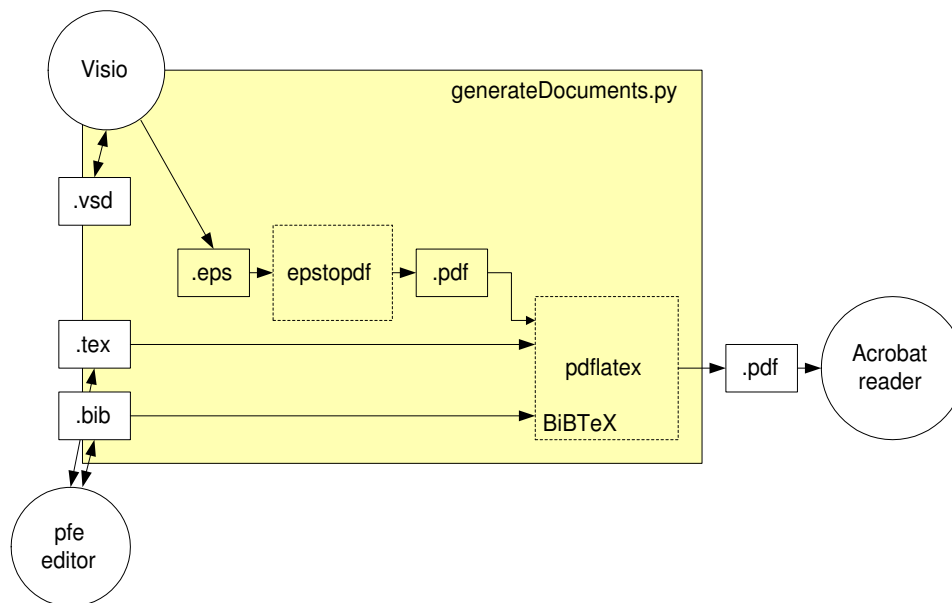


Figure 5: The different filetypes and the transformation chain from source to output

Version: 1.2, date: May 28, 2004 changed by: Gerrit Muller

- repaired figure 1
- repaired requirements section
- defined the logo

Version: 1.1, date: July 10, 2002 changed by: Gerrit Muller

- updated to reflect the use of pdflatex and epstopdf
- added management requirement about aggregation of documents

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.

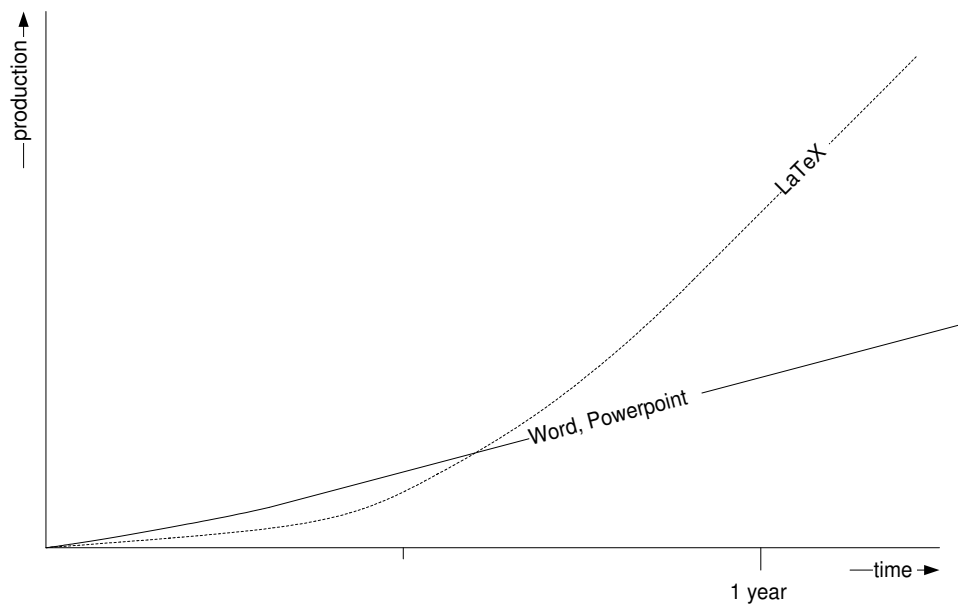


Figure 6: Comparison of expected productivity of LaTeX versus Word