

LaTeX3 architecture

and current work in progress

Frank Mittelbach



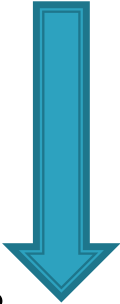
Introducing new concepts and ideas ...

I presume this is one of several dozen bugs that would arise over the years if anyone were foolish enough to try allowing "_" in command names.

Leslie Lamport



A Timeline

- ▶ 1982 TeX2
 - *4 years later ...*
 - ▶ 1986 LaTeX 2.09
 - *4 years later ...*
 - ▶ 1990 TeX3
 - *4 years later ...*
 - ▶ 1994 LaTeX2e
 - *5 years later ...*
 - ▶ 1997 LuaTeX beta
 - *5 years later ...*
 - ▶ 2012 **LaTeX3 beta** ???
- 1991 expl3 (first attempt)
• 1992 LaTeX3 architecture and kernel
- 
- ? What happened with it ?

The 1992/93 kernel

\LaTeX 3
Programmers Guide
Frank Mittelbach
93/09/30

2		CONTENTS
Contents		
I	Primitive data structures and concepts	7
1	Conventions	8
1.1	Functions	8
1.2	Parameters	9
2	Defining functions	10
2.1	Defining new functions	10
2.2	Defining internal functions (no checks)	10
3	Control sequence names	12
3.1	Functions	12
3.2	Predicates and conditionals	12
3.3	Internal functions	13
3.4	Internal variables	14
4	\LaTeX3 functions	15
4.1	Expanding arguments of functions	15
4.2	Defining new variables	16
4.3	Manipulating the first argument	17
4.4	Manipulating two arguments	17
4.5	Manipulating three arguments	18
4.6	Internal functions and variables	18
5	Tracing modules and functions	20
5.1	Functions	20
5.2	Internal variables	21
5.3	Tracing \TeX functions	22
5.4	Dumping the current definitions	22
5.4.1	Functions	22
6	Predicates and conditionals	23
6.1	Internal functions	23
6.2	Predicates	24
6.3	Variables	24
7	Checking functions and variables	25
7.1	Functions	25
7.2	Constants	25
7.3	Internal functions	26
8	Quarks	27
8.1	Functions	27
8.2	Constants	27
9	Counters	28
9.1	Functions	28
9.2	Formatting a counter value	29
9.3	Variable and constants	30
9.3.1	Internal functions	30

Reasons for Failure

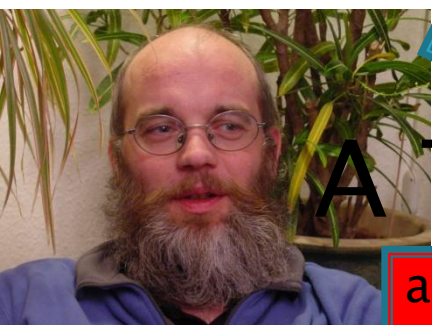
- ▶ Right Architecture (ideas) --- Wrong Time
 - Too radical
 - Too experimental
 - Too immature (and unexplained)
 - Far too slow and far too huge (when built on TeX)
- ▶ Burning Issues needed resolving first
 - “intermediate” version LaTeX2e was released

2.2.1991 pool size exceeded

- ▶ At the end of this message I attached the .log file for Frank's test file tparm4. This job crashes on 'pool size exceeded', for which I've been afraid since Frank sent his first proposal for the new kernel.
- ▶ The new font selection scheme, the new macro-naming convention, the resource database, ... : they all eat truck-loads full of pool space!
- ▶ I ***do*** like the interfaces of the modules I just mentioned, but I think this project is definitely going in the wrong direction:
it's nice but impractical!

Drivers (pains and objectives)

- ▶ No consistent design model
 - A few generic support commands, e.g., `\@startsection`
 - Limited flexibility, limited scope
- ▶ Most design changes required programming
- ▶ No (proper) management of logical or visual context
 - except for limited support of context for lists
 - Some hardwired context settings, e.g., footnotes in minipages



described by

A Typesetting Example



adjusted individually

```
documentclass[12pt]{article}
\usepackage{amsmath}
\title{\LaTeX}
\date{}

document preparation
program. It offers
extensive facilities
for desktop publishing,
including, tables and
\LaTeX() usage
```

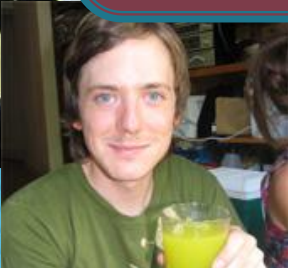
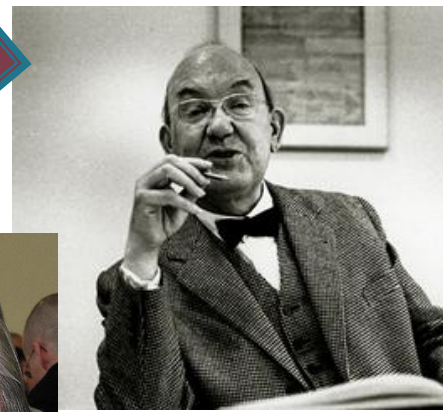
```
<?xml version="1.0"?>
<quiz>
<question>
Who was the forty-second
President of the U.S.A.?
</question>
<answer>
William Jefferson Clinton
</answer>
<!-- Note: We need to add
more questions later.-->
</quiz>
```

XML

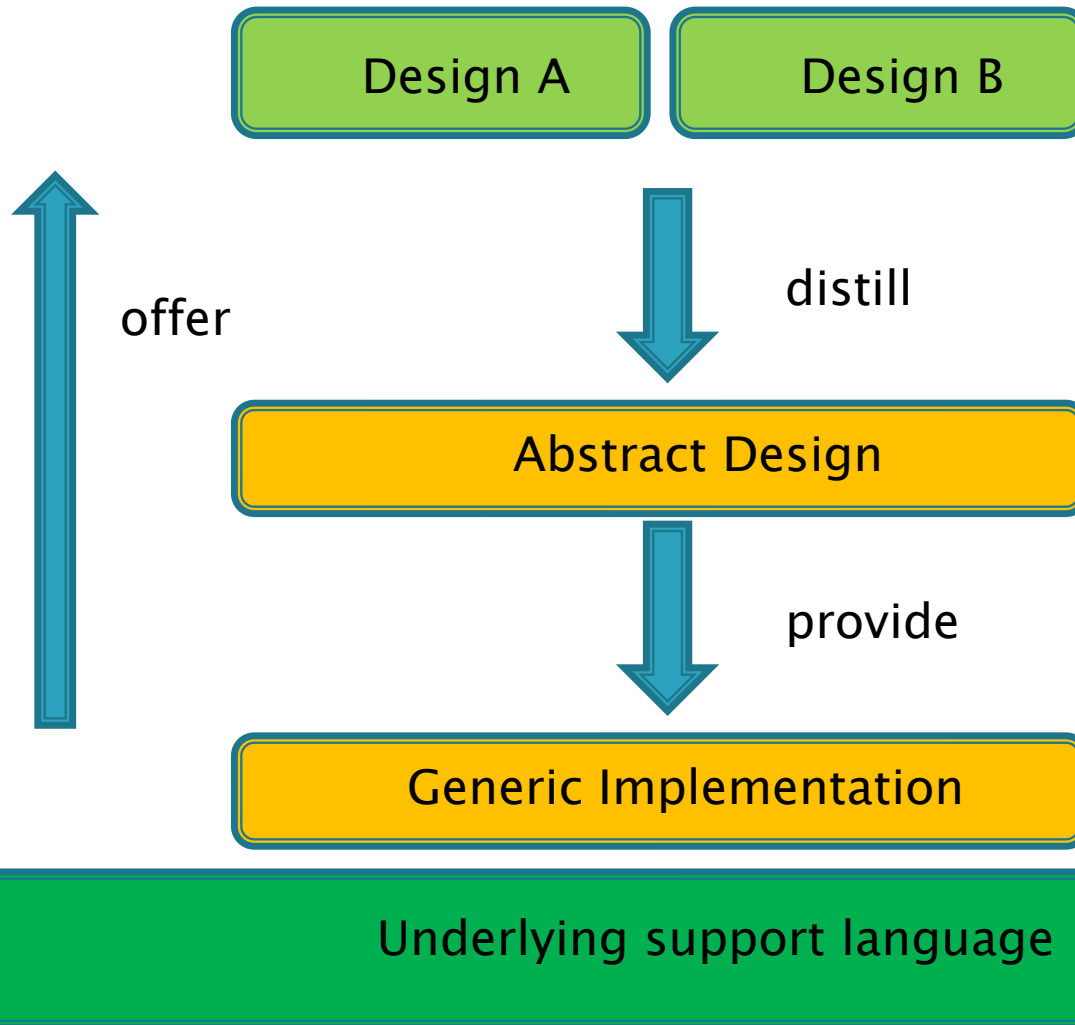


Academy Engraved LET: The quick brown fox jumped over the lazy dog.
ALGERIAN CONDENSED LET: THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.
Arrriba Arrriba LET: The quick brown fox jumped over the lazy dog.
Avant Garde Mono ITCIT: The quick brown fox jumped over the lazy dog.
PORTAGOITC TT: THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.
Blackmoor LET: The quick brown fox jumped over the lazy dog.
Bancroft TT-Heavy: The quick brown fox jumped over the lazy dog.
BRAGANZASCITC TT: THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.

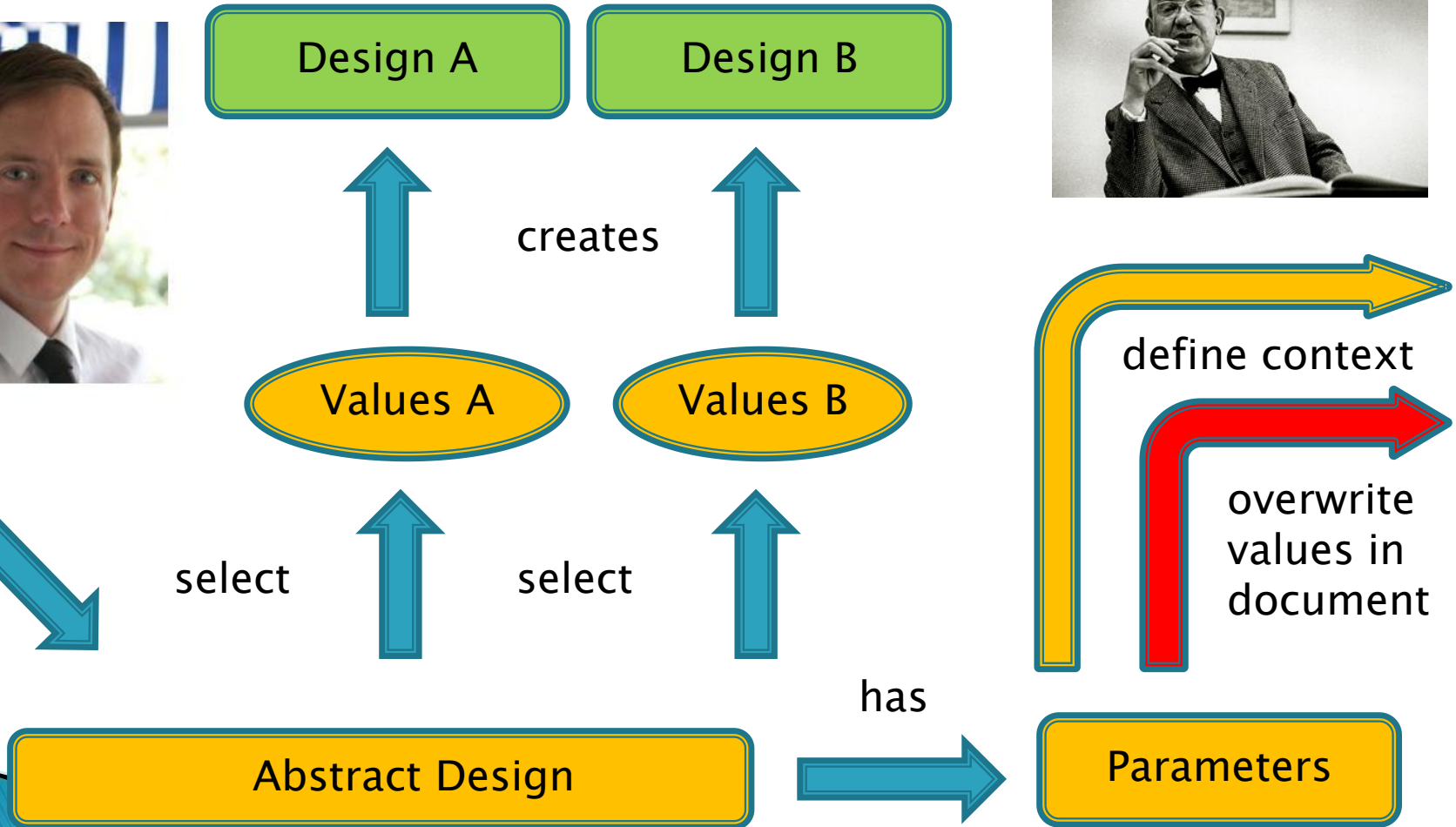
rendered as



Design Templates



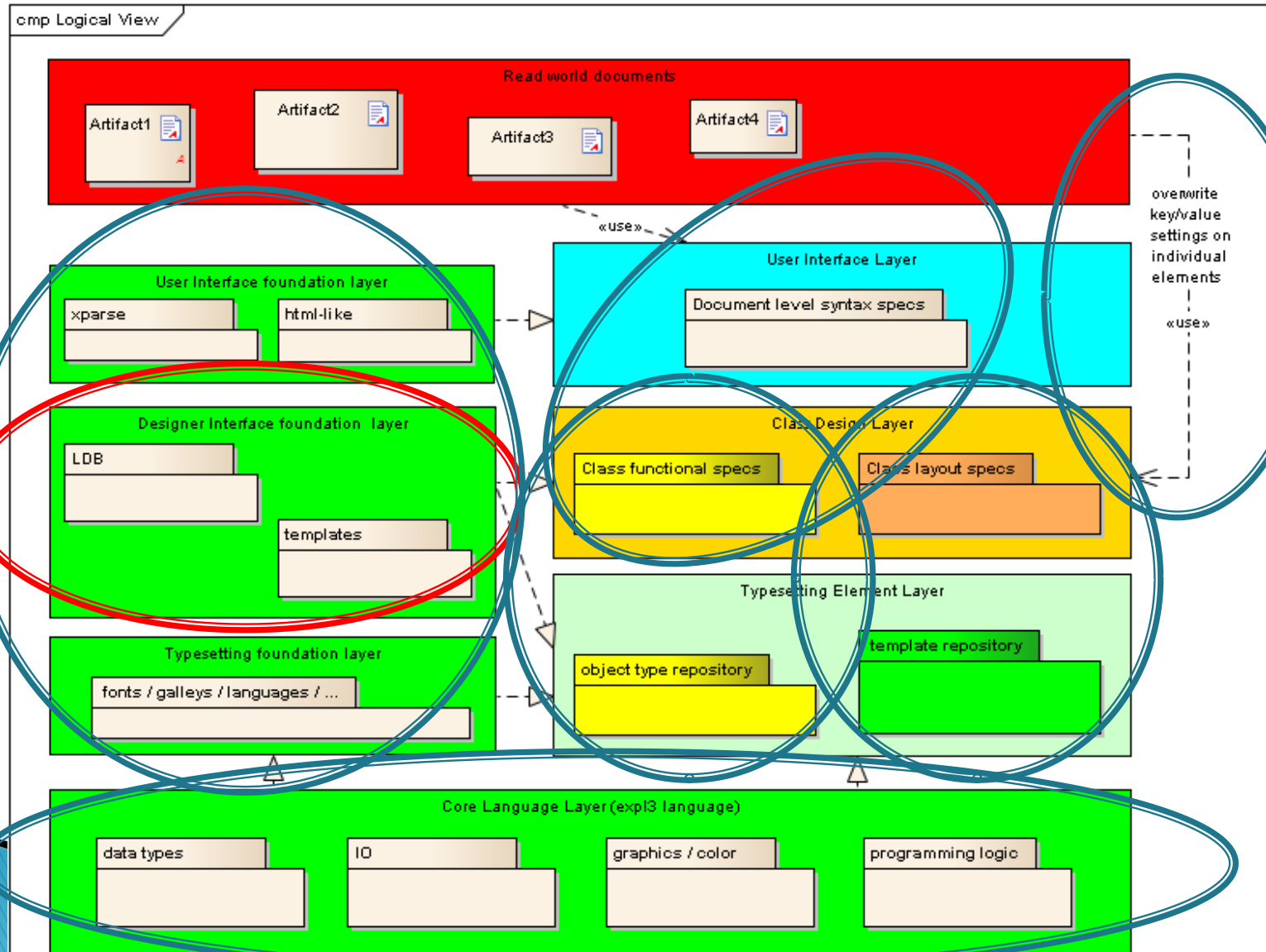
Design Templates



Functional Principles

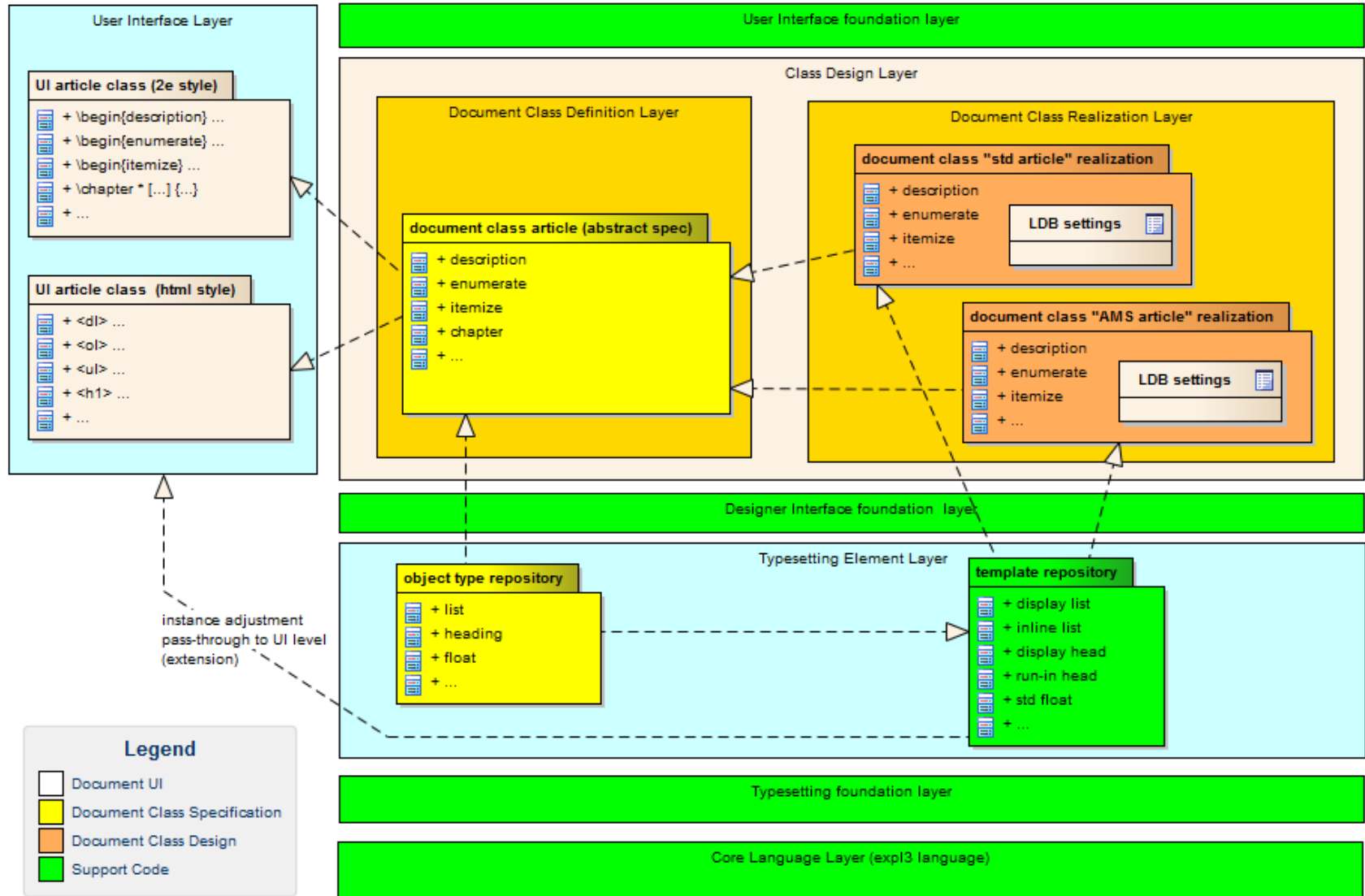
- ▶ Clear separation: UI, design, coding
 - Supports reuse and flexibility
- ▶ Logical and visual context dependencies are managed
 - Needed for high-quality results
- ▶ UI supports formatting adjustments
 - Perhaps strongest point of LaTeX
- ▶ Comprehensive, orthogonal programming language
 - Now why would you need this?

Architecture overview



Architecture details

class document class architecture (layers)



LDB – context dependencies

- ▶ Nesting and sequencing of structural elements in the document defines “context”
- ▶ Elements belong to one or more classes, e.g., “list” (generic), “itemize” (specific)
- ▶ Encountered elements update the context and applicable rules are carried out
- ▶ Notation:
 - <list start of environment of class “list”
 - itemize> end of environment of class “itemize”
 - <note> completed environment of class “note”
 - !head element of class “head”
 - * loose nesting
 - tight nesting
 - !head<list sequencing

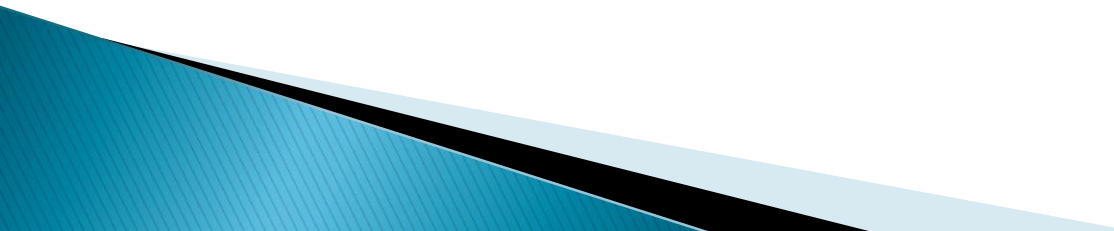
LDB - examples

Context	Explanation
<code>!head<list</code>	List immediately follows a heading
<code><list*<list</code>	List nested within list
<code><list*<itemize</code>	An “itemize” nested within some list
<code>list><itemize</code>	An “itemize” starts immediately after a list has ended
<code><float*<caption>*<caption</code>	Second “caption” environment within a float

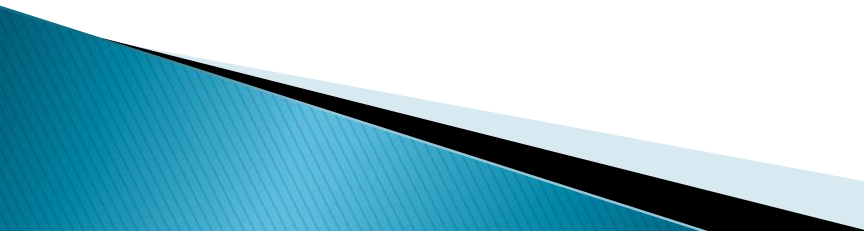


Live demo

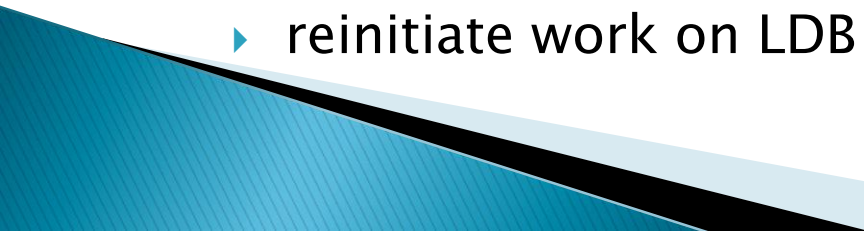
Summary

- ▶ Nothing in this architecture really requires TeX as a formatting engine.
 - ▶ But ...
 - ▶ With today's advances in the processing power of the underlying engine the ideas **now** appear to be feasible (in "x"TeX).
- 

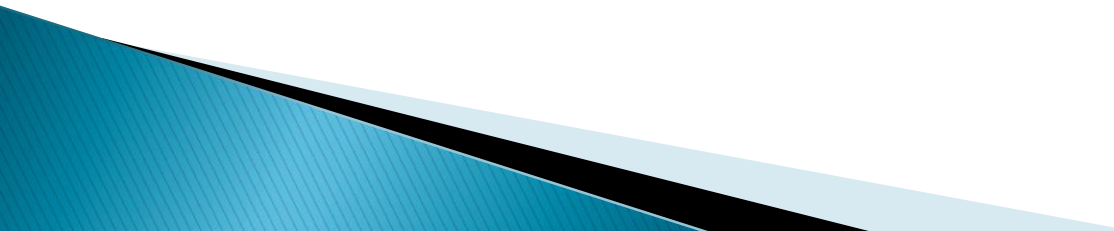
2011 Activities

- ▶ Big Bang (documentation and code cleanup)
 - much more consistent documentation
 - clarifying which functions are expandable or not
 - restructuring code into l3kernel, l3packages, l3experimental and l3trial
 - ▶ significant speed improvements in the kernel
 - faster `\prg_return_...`: conditional code, faster seq and prop
 - by a factor at least 3 on sample documents (e.g., siunitx)
 - ▶ expl3 now mostly stable
 - for those parts that have been moved to l3kernel
 - ▶ GitHub mirror
 - ▶ work on stand-alone kernel
- 

2011 Activities (cont.)

- ▶ modules l3str and l3regex currently written
 - possible extensions to code high-lighting
 - ▶ module l3fp being reimplemented
 - fast expandable IEEE-854 compliant decimal floating point arithmetic and expression parsing
 - ▶ module xcoffins / l3coffins
 - ▶ module xgalley
 - ▶ initial work on font support (xfss)
 - first task converting the highly optimized NFSS to less optimal but more readable code
 - ▶ reinitiate work on LDB
- 

Outlook

- ▶ Properly integrate template and LDB
 - ▶ Define a mechanism to overwrite template instance values on document level
 - ▶ Define standard environment management
 - ▶ Finish galley mechanism
 - ▶ Rework output routine concepts
- 

Introducing new concepts and ideas ...

There is nothing more difficult to take in hand, more perilous to conduct or more uncertain in its success than to take the lead in the introduction of a new order of things.

Machiavelli

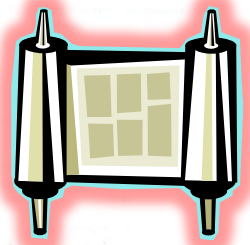




\LaTeX 3
 Programmers Guide
 Frank Mittelbach
 93/09/30



Questions?



2

Contents

I Primitive data types

1 Conventions

1.1 Functions

1.2 Parameters

2 Defining functions

2.1 Defining new functions

2.2 Defining internal functions

3 Control sequence

3.1 Functions

3.2 Predicates and control sequences

3.3 Internal functions

3.4 Internal variables

4 \LaTeX 3 functions 15

4.1 Expanding arguments of functions 15

4.2 Defining new variants 16

4.3 Manipulating the first argument 17

4.4 Manipulating two arguments 17

4.5 Manipulating three arguments 18

4.6 Internal functions and variables 18

5 Tracing modules and functions

5.1 Functions

5.2 Internal variables

5.3 Tracing \TeX functions

5.4 Dumping the current module

5.4.1 Functions

6 Predicates and conditions

6.1 Internal functions

6.2 Predicates

6.3 Variables

7 Checking functions and macros

7.1 Functions

7.2 Constants

7.3 Internal functions

8 Quarks

8.1 Functions

8.2 Constants 27

9 Counters 28

9.1 Functions 28

9.2 Formatting a counter value 29

9.3 Variable and constants 30

9.3.1 Internal functions 30

