



Website META Language

Fritz Zaucker <zaucker@ee.ethz.ch>
Tobi Oetiker <tobi@ee.ethz.ch>

Contents

- Separating Contents and Design
- XML the promised Land / Go there today!
- WML Structure / The 9 Pass Compiler
- A Real Site

Website Requirements

Producing a Website requires at least 3 people:

- Information Provider
- Designer
- Programmer

But what about maintaining a website ?

Ex 1, Ex 2, Ex 3,

A look at HTML history

The original Idea: Humans provide contents with mark-up, browser does the layout.

```
<address>Tobi Oetiker</address>
```

```
<address>La Jolla</address>
```

HTML Today

Designers did not like this: We want full control!

Table, Images, Layers, Font tags, Absolute Positioning.

```
<table border=0 bgcolor=#000000 cellpadding=1
      cellspacing=0><tr><td>
<table border=0 bgcolor=#ffffff cellpadding=1><tr>
<td><font face="Arial,Helvetica,San-Serif" size=2>
Tobias Oetiker<br>La Jolla
</font></td></tr></table></td></tr></table>
```

The Promise: XML

Back to the roots ...

```
<address>  
<name>Tobias Oetiker</name>  
<city>La Jolla</city>  
</address>
```

the only problem: this is not supported yet ...

The Fulfilling

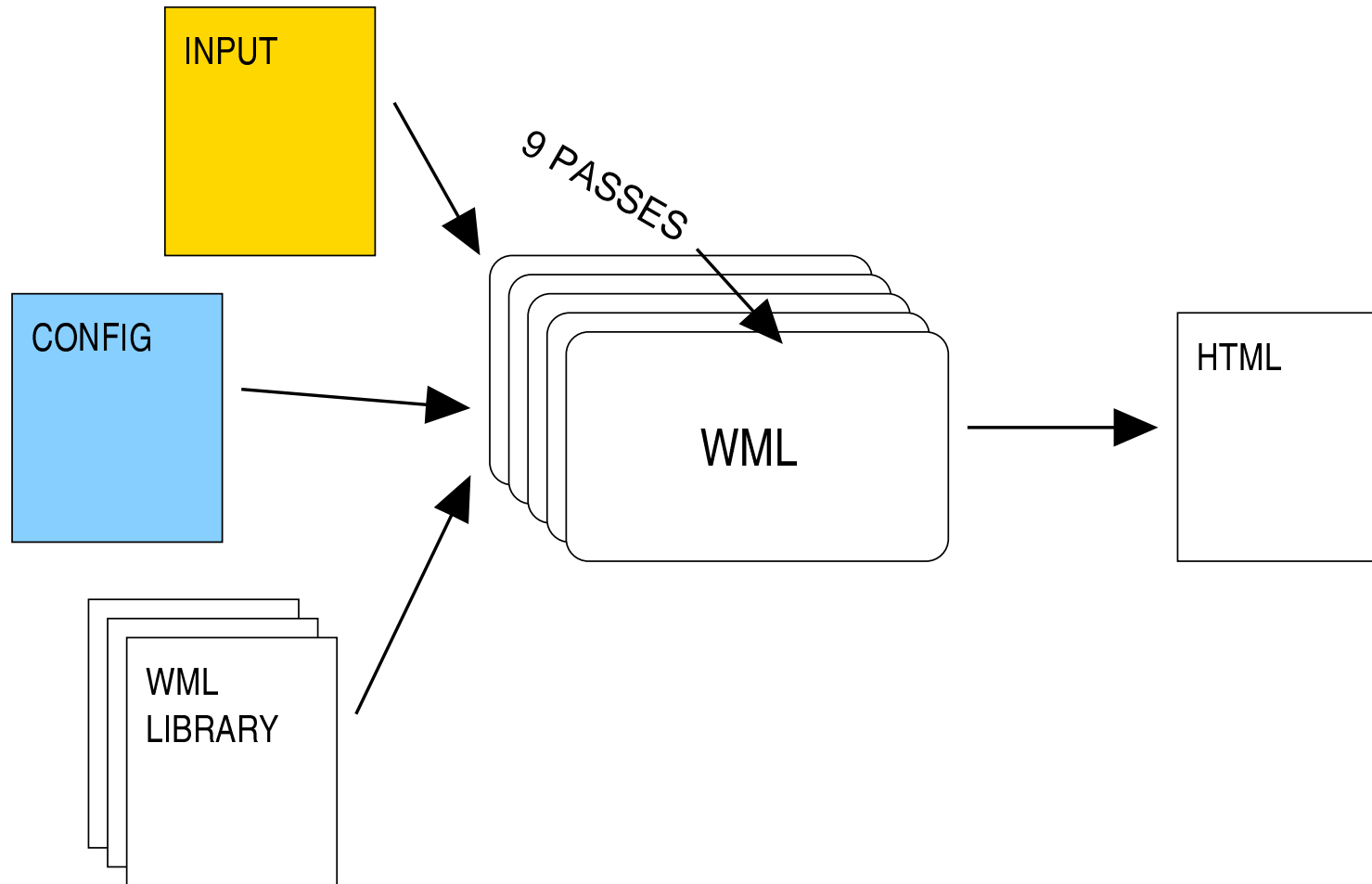
The Idea:

- Write input in a simple XML like language
- Process input into HTML
- View with normal browser

The Implementation:

Ralf Engelschalls: Website Meta Language (WML)

WML Structure



Nine Languages in one Tool

P1 Include Pre-Processor (Somewhat like CPP). Supports
`#include "file.inc"`

P2 Meta-HTML (Add HTML tags using HTML like Language)

P3 Embedded Perl – Use arbitrary Perl code in your webpages.

P4 GNU m4 – macro preprocessor

P5 Diversion Filter (Redirect Output within the Document)

Nine Languages in one Tool (cont.)

P6 Area Substitution (Regex Search and Replace on parts of the document)

P7 HTML Fix (Size Images . . .)

P8 HTML Strip (Remove Excess Space)

P9 Slice output into several files. (Multilingual Sites, Frame Support)

P1 Include Pre-Processor

Source Reading and Include File Expansion

```
#include "file.wml"  
#include "file2.inc" variable1=value1 ...  
#use wml::category::file  
#use wml::category::file2 variable=value ...
```

a line with an image `\`
`` which is continued at this line with `variable=$(variable)`

```
# lines starting with a hash are comments
```

P2 Meta-HTML Compiler

High-Level Macro Construct Expansion

```
<define-tag foo>  
Fully Open Object  
</define-tag>
```

```
<define-container bar>  
%body is a <foo>  
</define-container>
```

```
<bar>Grolok</bar>
```

Grolok is a Fully Open Object

P3 Embedded Perl 5 Language

Programming Construct Expansion

```
<perl> print "local time is ".localtime(time) </perl>  
<: print 22*15+340 :>  
<:= 22*15+340 :>
```

P4 GNU m4

Low-Level Macro Construct Expansion

```
#use wml::std::tags  
<symbol bar BAZ>  
foo bar quux
```

P5 Diversion Filter

```
<H1><<TITLE>></H1>  
<hr>  
<<BODY>>  
..TITLE>>Title of my Document<<..  
..BODY>>What I have to say<<..
```

P6 Area Subst

Character and String Substitution

```
{: [[s|ä|&auml;|]]  
Foo Bar Baz Quux [[s|ü|&uuml;|]]  
with Umlauts ä and ü  
:}
```

P7 HTMLfix

Markup Code Fixup

- add missing WIDTH and HEIGHT attributes
- add missing '#' in color definitions
- add quotes to `<numeric attributes="1">`
- replace obsolete or proprietary tags like `<center>` by `<div align="center">bla</div>`
- ...

P8 HTMLstrip

Markup Code Stripping

- remove newlines
- recognizes `<pre>`

P9 Slice

Markup Code Splitting and Output Generation

```
[EN>Welcome to:] [DE:Willkommen zu:] Foo Bar Baz Quux!
```

So you must be a Programmer?

- Setup is demanding, but there is good documentation.
- Translation from WML to HTML happens off-line. Complex operations are not a problem.
- Authoring Pages is trivial. Once the templates are done.
- Everybody (should) know(s) Perl and META-HTML is simple to learn.

WML Input

```
<PAGE AUTHOR = "Tobi Oetiker <tobi@ee.ethz.ch>"  
    TYPE = "head" PAGE = "txt_latex">
```

```
<H1>LaTeX</H1>
```

```
<LINK HREF="http://ee-staff.ethz.ch/lshort/lshort.pdf"  
    TITLE="The not so Short Introduction to LaTeX2e">
```

After reading this booklet, you should be fit to write your term paper or thesis with LaTeX. There are also a [Postsript version](/lshort/) of this document available.

```
</LINK>
```

...

META-HTML definition

```
<define-container LINK>
<preserve TITLE><preserve HREF>
<set-var %attributes>
<TABLE><TR><TD COLSPAN=2>
  <if <get-var HREF> <prog <A HREF="<get-var HREF">">>>
  <B><get-var TITLE></B>
  <if <get-var HREF> "</A">
  <hspace 10>
  <if <match <get-var HREF> "^(ht|f)tp://"> "(off-site)">
</TD></TR>
<TR><TD width=30><hspace 30></TD>
  <TD width=100%><vspace 3><BR>%body</TD>
</TR></TABLE>
<restore HREF><restore TITLE>
</define-container LINK>
```

WML Output Source

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML lang="en">
<!-- These pages have been designed by Tobias Oetiker oetiker@ee.ethz.ch
      using WML, The Gimp, ImageMagick, FreeType, Tidy and Perl -->
<HEAD>
<TITLE>LaTeX</TITLE>
<STYLE type="text/css">
<!--
HR, H1, H2, DIV, P, LI,
UL, OL, TABLE, TD, TR {
      font-family: lucida-sans, helvetica, arial, sans-serif}
EM {font-style: italic}
TT {font-family: lucida-typewriter, courier, courier new}
-->
</STYLE>

<SCRIPT type="text/javascript" language="JavaScript">

<!--//Deframe Me
      if(top.frames.length> 0)
      top.location.href=self.location;
//-->

</SCRIPT>
</HEAD>
<BODY background="../../../img/backdrop.gif" bgcolor="#ffffff" text="#000000" link="#4f94ff" vlink="#4f9
<!-- title area ***** -->
<TABLE summary="Logo and title area" width="625" cellpadding="0" cellspacing="0" border="0">
<TR>
<TD valign="top" width="155" align="left"><IMG src="../../../img/imgdot-1x1-transp.gif" alt="" width="1
<IMG src="../../../img/imgdot-1x1-transp.gif" alt="" width="1" height="10" align="bottom" border="0"><B
<IMG alt="Student Computing Logo" src="../../../pics/logo-en.gif" width="130" height="100"><BR>
<IMG src="../../../img/imgdot-1x1-transp.gif" alt="" width="1" height="10" align="bottom" border="0"><B
</TD>
<TD width="$NavBorder"><IMG src="../../../img/imgdot-1x1-transp.gif" alt="" width="20" height="1" align
<TD valign="top" width="450"><IMG src="../../../img/imgdot-1x1-transp.gif" alt="" width="1" height="5"
</TABLE summary="Title area" width="100" height="75" cellpadding="0" cellspacing="0" border="0">
```

LINK

More Information

<http://www.engelschall.com/sw/wml> The WML Homepage has full online documentation and many examples.

<http://webfactory.ee.ethz.ch> This site gives background information about web design with WML. There is also the on-line version of an article about WML in the ETH-Input magazine with some examples.

Exercises

```
wget http://ee-staff.ethz.ch/~zaucker/talks/wml/wml-course.tgz
tar xzf wml-course.tgz
```

There are 4 subdirectories below `wml` named `part1` to `part4` with several example files `*.wml`, WML configuration files (`.wmlrc` and `.wmkrc`), a README file and template and include files below the `.wml/` directory.

The example files can be translated with one of the following commands:

```
wmk ex0.wml
wmk -f ex0.wml
wmk -a .
wmk -fa .
```

The resulting output files will be in the same directory as the input files and have the ending `.html`

Exercise 0

- Load ex0.wml in your browser.
- Examine ex0.wml
- Translate ex0.wml and look at ex0.html

Exercise 1

Most stuff was moved from the *.wml file into .wml/template_1.inc

- Translate ex1.wml and look at ex1.html with your browser
- Change template_1.inc to template_1a.inc, translate and view in browser.
- Same with template_1b.inc

Examine ex1.html after each translation. Look at the template files in .wml/

Exercise 2

Replaced `#include` statement with PAGE tag. Added parameters to BODY tag.

Added list environment for later.

- Look at `.wmlrc` and `template_2.inc`
- Play with settings for `MarginWidth` and `MarginHeight` in `.wmlrc` file.

When modifying `.wmlrc` or `template` files you must run `wmk -f` to force translation.

Exercise 3

Redefined UL and LI tags in `.wml/html3.inc` (included from `template_3.inc`).

- Examine `.wml/html3.inc`
- Play with definition of LI tag

Exercise 4

Added OL list. Redefined LI tag breaks OL numbering.

- Think about how this could be fixed.
- Try `TYPE=4a`
- Examine `.wml/html4a.inc`

How could LI tag be improved for UL lists?

Exercise 5

Adds a footer (TYPE=5) and header (TYPE=5a).

- What's wrong with the ETH logo?
- Try TYPE=5b and examine the difference between template_5a.inc and template_5b.inc
- Try TYPE=5c

Exercise 6

Change to part2 directory.

Added a spacer below H1 heading and language tags in .wmlrc, .wmkrc, and template_6.inc.

- Translate ex6.wml
- What files were created?
- View output files.

Exercise 7

Completely bi-lingual.

- Translate ex7.wml
- View output files (see language selector)

Exercise 8

Change to part3 directory.

GIF-support.

- Translate ex8.wml
- View output files
- Where are the .gif files (right mouse button on image)?

What are the advantages/disadvantages of images?

Exercise 9

Change to part4 directory.

Added navigation bar.

- Translate ex9.wml and view output.
- Translate site: wmk -a .
- Change TYPE=9 to TYPE=9a and re-compile:

```
perl -i -p -e 's/TYPE=10/TYPE=10a/' *.wml */*.wml */**/*.wml
wmk -fa
```

- Replace ColBasSec in .wmlrc: #bfb9f7 and re-compile:

```
rm .img/*
wmk -fa
```

Exercise 10

Start with one of the examples and create your own template.