

# Semantics in HTML

Petr Nálevka



University of Economics, Prague

*Dept. of Information and Knowledge*

*Engineering*

[petr@nalevka.com](mailto:petr@nalevka.com)

<http://nalevka.com>



This presentation is available at:

<http://nalevka.com/semantic.pdf>

# Semantics in HTML

- Table of Contents
  - What Tim did, first HTML versions
  - How did the world screw things up, presentational pollution
  - How did W3C try to rescue HTML, HTML 4.01
  - Where is semantics hiding in current HTML
  - Return to semantics
  - RDF vs. semantic tagging



# The Beginning...

- Tim Berners-Lee at CERN (1980-1989)
- Invented WWW (URL, HTTP and HTML)
- First HTML was purely **SEMANTIC**
  - Non-structured text → Hierarchical tagging
    - Assigning special additional meaning to parts of the text
    - Such tagging makes text more understandable for humans (different presentation for differently tagged text) and also for computers (semantic queries, indexers etc...)

# First HTML

- HTML tags (1991)
  - 22 elements, SGML
  - Purely semantic
  - RFC (until HTML 2.0)
- Specifically designed for tagging articles (scientific)
- Presentation depends on browser
- Machines can easily extract information (parser)
  - Indexing engine

```
<HTML>
  <TITLE>A sample HTML
instance</TITLE>
  <H1>An Example of Structure</H1>
  Here's a typical paragraph.
  <P>
    <UL>
      <LI>Item one has an <A
NAME="anchor">anchor</A>
      <LI>Here's item two.
    </UL>
</HTML>
```



# Semantics or presentation?

- Articles → Commercial presentations
- Documents → Web applications
- Polluting HTML with presentation
  - Browser vendors, commercial sector (pixel precise design)

# HTML 3.2

- Awful code
- Presentation directly in HTML
  - Colors, fonts, aligning, line breaks ...
- Frames
- Tables misused for layouting
- Many of them still used in current Web

```
<FONT COLOR="#446677">  
<FONT SIZE="15">
```

```
<P ALIGN="right">  
<P VALIGN="right">
```

```
<BR>
```

```
<CENTER>Centered text
```

```
<SMALL>little text
```

```
<BODY BGCOLOR="#FF0000">
```

and many more...



# Consequences

- HTML become more multimedial and colorful, but was loosing semantics
  - Difficult to be processed by machines, but also less accessible to humans
  - Web is accessible only for privileged users
    - Healthy (no disabled), big color screen, right OS, right browser, right version
- The language became a mess
  - Trying to solve all different problems but not solving any of them in a satisfactory manner



# Main Problems of HTML

## 1. Presentation mixed with semantics

- Machine readability
- Accessibility
- Web development

## 2. One language (HTML) is not enough to semantically express the whole Web in it's complexity

- Internet shops, company presentations, newspapers, encyclopedias, chats, blogs, online community portals, webmail, online PIM with collaboration, porn-sites etc...
- W3C is trying to solve both problems



# First Problem

## Presentation mixed with semantics

# HTML 4.01

- Solving partly the first problem
- 3 flavours
  - Frameset
  - Transitional } Backward compatibility
- Strict (only strict did significantly reduce presentation)
- Even in strict, many **presentational** aspects
  - Forms (radio button), line breaks...
- HTML 4.01 Strict reduced presentation significantly but did **NOT** add any semantics
  - The way it is used now it did even reduce semantics



# Breakthrough – Styling Languages

- In these days Web needs presentation, can't just depend on the browser
  - Styling languages: CSS, XSL...
- Device specific presentation in one place, separate from the document
- Helped web developers
- Same HTML document, completely different appearance
- Can style not only HTML but any SGML/XML languages



# HTML of Today Web

```
<div class="weather">
  <span class="title">Weather
Forecast</span>
  <span class="city">Prague</span>
  <span
class="temperature">25&deg;C</span>
  <div class="cloudy"/>
</div>
```



```
.weather {
background-color:#CCDDFF;
border:1px solid black;
padding:5px;
width:200px;
padding:10px;
}
.weather .title {
font-weight:bold;
margin-bottom:10px;
}
.weather .city {
float:left;
font-style:italic;
padding-top:5px;
}
.weather .temperature {
float:right;
padding-top:5px;
}
.weather .cloudy {
background:transparent
url(cloudy.png)
no-repeat scroll 80px 0%;
height:50px;
width:250px;
}
```



# Where did Semantics go?

- HTML of today reduced to `div span div span`  
WooDoo
  - Semantically empty
- Semantics hides in the `class` attribute
- Disadvantages
  - Div, span brings no information, difficult to read, error-prone, makes code ugly
  - Class-based semantics is difficult to standardize (microformats) and to validate



## vCard

```
BEGIN:VCARD  
VERSION:3.0  
N:Nalevka;Petr  
FN:Petr Nalevka  
URL:http://nalevka  
.com  
END:VCARD
```

# Microformats



[Petr Nalevka](http://nalevka.com/)

## hCard

```
<div class="vcard">  
  <a class="url fn" href="http://nalevka.com/">  
    Petr Nalevka</a>  
</div>
```

- Just a hack, because of Internet Explorer 6 and below
- Allows to attach semantics and keep ill semantically empty HTML model

# Isn't that much nicer?

```
<weather>
  <title>Weather
Forecast</title>
  <city>Prague</city>
<temperature>25&deg;C</tempera
ture>
  <cloudy/>
</weather>
```



```
weather {
background-color:#CCDDFF;
border:1px solid black;
display:block;
padding:10px;
width:200px;
}
weather title {
display:block;
font-weight:bold;
margin-bottom:10px;
}
weather city {
float:left;
font-style:italic;
padding-top:5px;
}
weather temperature {
float:right;
padding-top:5px;
}
weather cloudy {
background:transparent url(cloudy.png)
no-repeat scroll 80px 0%;
display:block;
height:50px;
width:250px;
}
```



## Second Problem

**One language (HTML) is not enough to express the whole Web in it's complexity**

# Add semantics by naming your tags

- Do not map **YOUR RICH** domain specific semantics to the very **POOR HTML** semantic categories
  - Do express your full semantics directly in your Web documents
  - Much easier to process with XML tools
    - XSLT, XPath, XQuery, Schema languages and validation
- Majority of today's browsers already able to render

# Make your semantics understandable

- When ever some of your semantics matches a standardized language **USE IT** to make your document understandable for others
- You need to judge disadvantages of loosing semantics against advantages of being **UNDERSTANDED = VISIBLE, ACCESSIBLE**

# Ontologies

- ONTOLOGY = syntax and semantics of a language
- Syntax – defined by Schema (Relax NG, NVDL)
  - makes document automatically validable
  - Ensures interoperability of documents
- Semantics
  - Described verbally in language specification

# One language is not enough

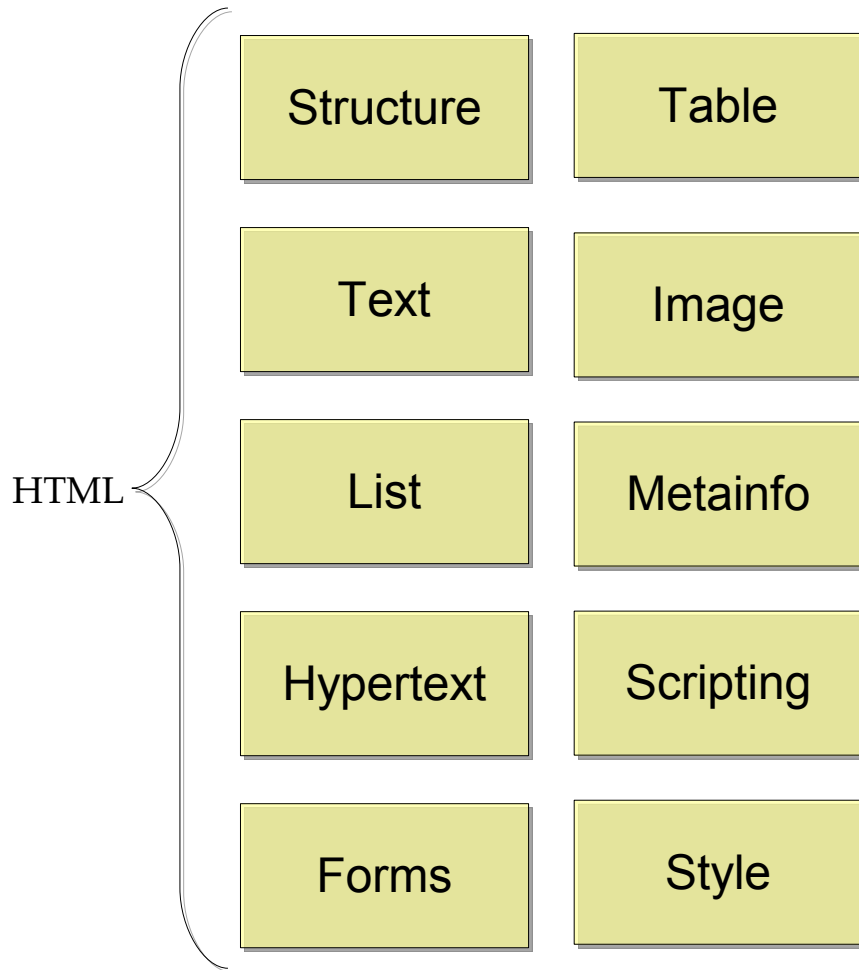
- Combine more languages within one document to express maximum of the semantics
  - Find the most suitable combination of languages for your particular task
  - Use your own semantics for highly specific tasks
- W3C is moving this direction
  - XHTML – XML brings **NAMESPACES**
  - Modularization of XHTML



# Namespaces and Compound Documents

```
<?xml version="1.0" standalone="yes"?>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:svg="http://www.w3.org/2000/svg">
  <head>
    <title xml:lang="en">Sample XHTML + SVG
document</title>
  </head>
  <body>
    <svg:svg width="4cm" height="8cm" version="1.1"
baseProfile="tiny" >
      <svg:ellipse cx="2" cy="4" rx="2" ry="1" />
    </svg:svg>
  </body>
</html>
```

# Modularization of XHTML



- Module – logically separated functionality
- Each module can be replaced by a full-featured language
  - Hybertext → XLink
  - Forms → XForms
  - Image → SVG
  - Metainfo → RDF
  - Table → Other table model
- Advantages
  - Tailored for specific task
  - Semantical
  - Declarative
  - Rich

# XML-based Semantic Web

- Example – Selling immobilities
  - Specific language, standardized – agreed upon semantics, syntax specified through a schema – ensures interoperability

```
...
<r:flat xmlns:r="http://www.reality.com/offer" xmlns:r="http://www.reality.com/offer">
  <r:address>
    <r:street>Raichlova<r:num>2816</r:num><r:secNum>8</secNum></r:street>
    <r:postcode>15500</r:postcode><r:city>Praha<r:city><r:cityPart>Praha
13<r:cityPart>
    <r:country>Czech republic</r:country>
  </r:address>
  <r:rooms>
    <r:room type="bedroom" sq="18"/>
    <r:room type="kitchen" number="25"/>
    ...
  </r:rooms>
  <r:price currency="CZK" vat="19">7250000</r:price>
  <r:finishDate>03/04/2008</r:finishDate>
  <r:floor>4</r:floor>
  <r:kmToCenter>25</r:kmToCenter>
  
  <r:contact>...</r:contact>
  <r:orientation><r:side>north</r:side>...<r:orientation>
  ...
</r:flat>
```

# Semantic Queries

- Queries

- All flats in Prague less than 30 kilometres from the centre with 5 rooms and at least 100 square meters for less than 6000000 CZK
- All flats with a terrace at least 40 meters big
- All flats in Prague with a garage and windows oriented on the east side which will be finished this year
- All flats without annoying neighbours :)



# XML tagging vs. RDF

- Why to use RDF, semantics may be attached using XML tagging
- XML tagging
  - (+) Only one source of information for machines and humans (no cheating)
  - (+) Very easy to write and understand even with non-visual editors
  - (+) Covers most requirements for Semantic Web
  - (+) Supported by most current browsers
  - (–) Doesn't support some advanced features like reasoning directly



Thank you for attention!

For more info visit:

<http://nalevka.com>

Ask questions at:

[petr@nalevka.com](mailto:petr@nalevka.com)

