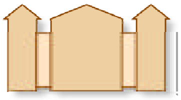
Hypermedia Presentation Generation on the Web

Lynda Hardman
Jacco van Ossenbruggen
CWI Amsterdam



Talk overview

- Introduction and historical background
- Multiple delivery publishing (MDP)
- MDP on the Web: Style sheets
- 2nd generation multimedia - Cuypers
- Towards 3rd generation multimedia

What is a “Document”?

Examples:
- Book, poem
- Article, paper, report
- Memo, e-mail, letter, etc

Definition:

> A document is a self-contained unit of information, intended to be communicated to a human interpreter

What isn t a document?

All data that is:
- Fragmentary
- Intended solely for further machine processing

Examples:
- Database records
- HTTP requests
- Software source code
- RDF metadata ...

Electronic Documents (then)

- Goal (authoring/production):
  - More efficient/effective production by using WYSIWYG authoring interfaces (WP,DTP)
- Goal (final-form):
  - Obtain same typographic quality as traditional print
- Production electronic, dissemination and final-form still on paper
- Authoring & storage format:
  - Mimics final-form presentation format

Electronic Documents (now)

- Goal (authoring/production):
  - Efficient, industrial scale, full document life cycle
- Goal (final-form):
  - Improve communication by exploiting presentation potential of new media
    - Use of audio, video, animation, etc
    - Interactivity (hyperlinks, forms, etc.)
    - Dissemination over internet (WWW)
    - Use of document technology to access (legacy) information
- Both production & dissemination is electronic
- Authoring & storage format:
  - Differs radically from presentation format

Electronic Documents: Issues

Problem: many document formats cannot cope with changing environment (c.f. issues in software engineering)

- –Hardware dependencies (use of printer/typesetter specific control sequences)
- –Software dependencies (use of proprietary formats)
- –Presentation dependencies (layout and style)

Related issues:

- –Longevity (many documents need to last >30 years)
- –Maintenance & reuse
- –Flexibility & tailorability

“Solution”

(Semi-automatically) convert all documents to new format or new layout

- –Expensive
- –Time consuming
- –Error prone (& pretty boring too!)
- –Loss of (implicit) information

Real solution

Multiple delivery publishing model

Multiple delivery publishing (MDP)

- •MDP distinguishes two formats
  - –One for authoring and long term storage
  - –Another one for final-form presentation
- •Mappings from source to target format
- •Source format can now abstract from all details that are likely to change in the target
- •Sounds pretty straightforward eh?
- •But it actually meant…

Revolution!

Software developers

No longer control their application’s own file format

Document authors

No longer control style and layout of their documents

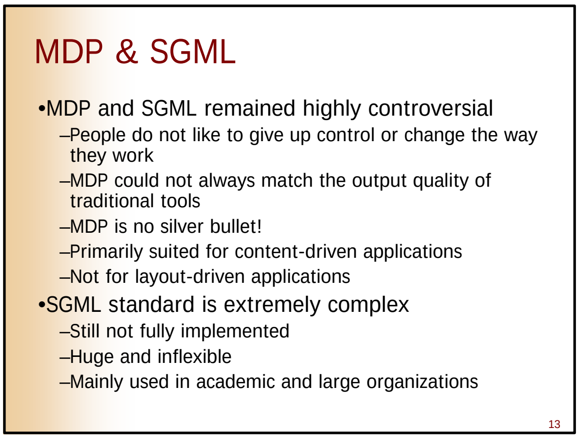
Tools

No longer used the “sacred” WYSIWYG paradigm

Multiple delivery publishing was not obvious at all!

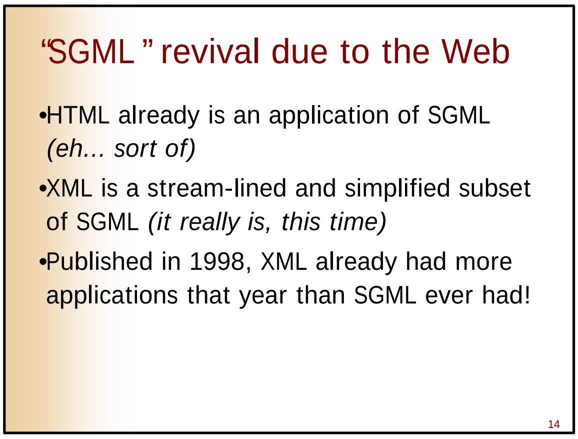
MDP: Nothing new …

- •This approach was already advocated by Goldfarb et al. in the 70’s!
- •Source documents encoded using IBM’s Generic Markup Language (GML)
- •GML was standardized by ISO in 1986 as SGML
- •First publicly available parser developed at the VU
  - –Amsterdam SGML Parser by Warmer, Van Egmond and Van Vliet (late 80’s)



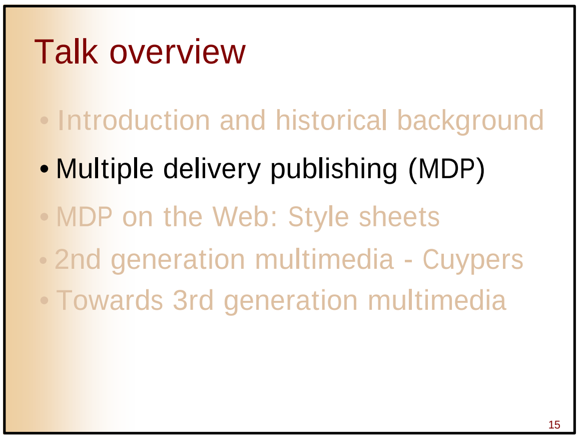
MDP & SGML

- MDP and SGML remained highly controversial
  - People do not like to give up control or change the way they work
  - MDP could not always match the output quality of traditional tools
  - MDP is no silver bullet!
  - Primarily suited for content-driven applications
  - Not for layout-driven applications
- SGML standard is extremely complex
  - Still not fully implemented
  - Huge and inflexible
  - Mainly used in academic and large organizations



"SGML" revival due to the Web

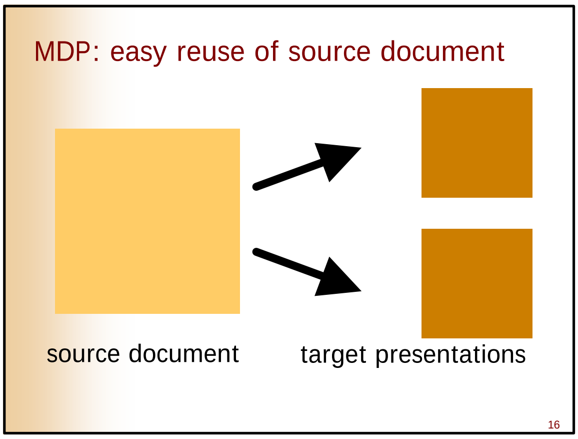
- HTML already is an application of SGML *(eh... sort of)*
- XML is a stream-lined and simplified subset of SGML *(it really is, this time)*
- Published in 1998, XML already had more applications that year than SGML ever had!



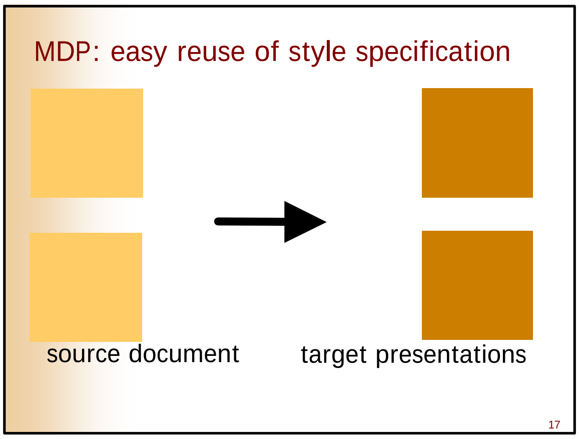
Talk overview

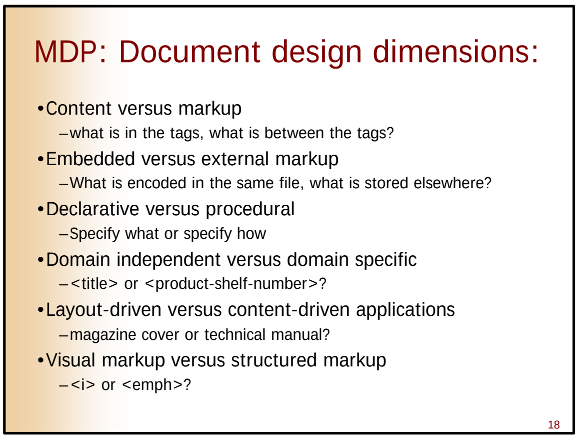
- Introduction and historical background
- Multiple delivery publishing (MDP)
- MDP on the Web: Style sheets
- 2nd generation multimedia - Cuypers
- Towards 3rd generation multimedia

MDP: easy reuse of source document



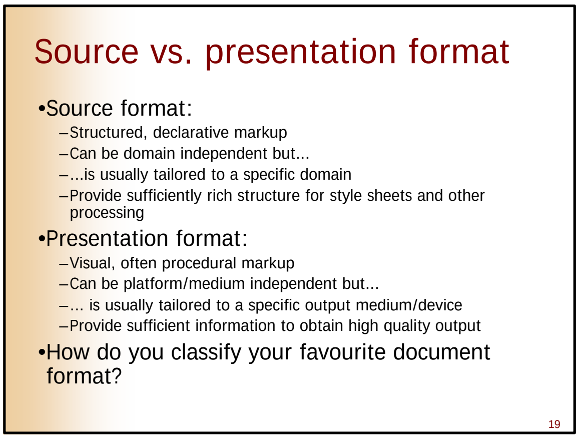
MDP: easy reuse of style specification





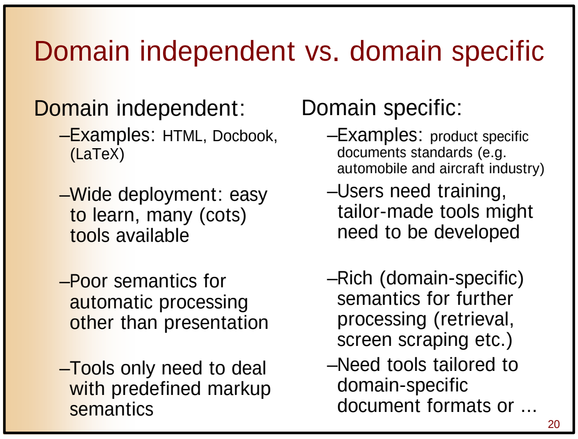
MDP: Document design dimensions:

- Content versus markup
  - what is in the tags, what is between the tags?
- Embedded versus external markup
  - What is encoded in the same file, what is stored elsewhere?
- Declarative versus procedural
  - Specify what or specify how
- Domain independent versus domain specific
  - <title> or <product-shelf-number>?
- Layout-driven versus content-driven applications
  - magazine cover or technical manual?
- Visual markup versus structured markup
  - <i> or <emph>?



Source vs. presentation format

- Source format:
  - Structured, declarative markup
  - Can be domain independent but…
  - …is usually tailored to a specific domain
  - Provide sufficiently rich structure for style sheets and other processing
- Presentation format:
  - Visual, often procedural markup
  - Can be platform/medium independent but…
  - … is usually tailored to a specific output medium/device
  - Provide sufficient information to obtain high quality output
- How do you classify your favourite document format?



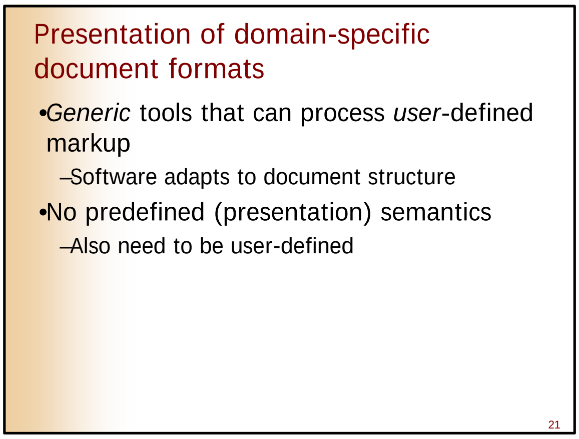
Domain independent vs. domain specific

Domain independent:

- Examples: HTML, Docbook, (LaTeX)
- Wide deployment: easy to learn, many (cots) tools available
- Poor semantics for automatic processing other than presentation
- Tools only need to deal with predefined markup semantics

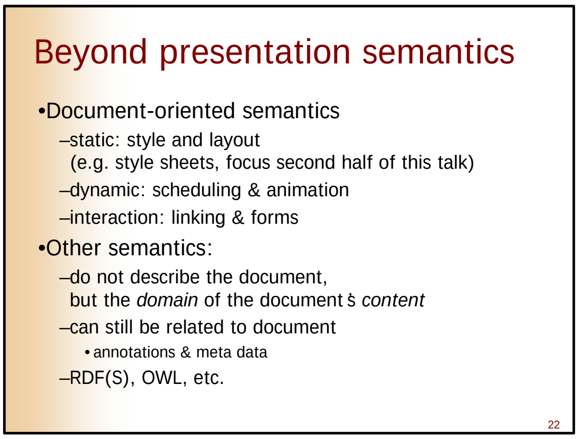
Domain specific:

- Examples: product specific documents standards (e.g. automobile and aircraft industry)
- Users need training, tailor-made tools might need to be developed
- Rich (domain-specific) semantics for further processing (retrieval, screen scraping etc.)
- Need tools tailored to domain-specific document formats or …



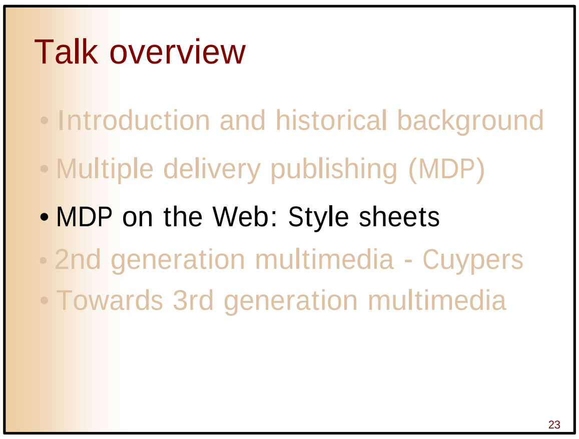
Presentation of domain-specific document formats

- *Generic* tools that can process *user*-defined markup
  - Software adapts to document structure
- No predefined (presentation) semantics
  - Also need to be user-defined



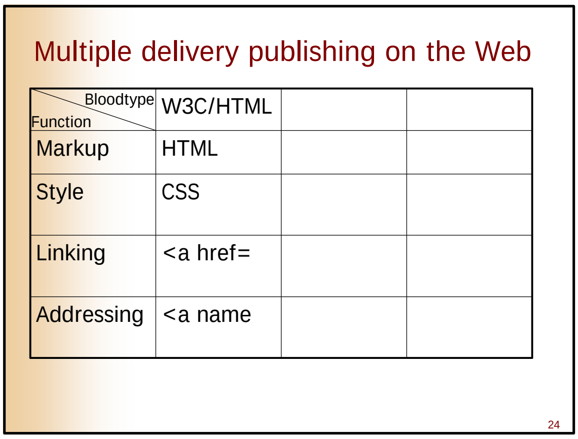
Beyond presentation semantics

- Document-oriented semantics
  - static: style and layout
    (e.g. style sheets, focus second half of this talk)
  - dynamic: scheduling & animation
  - interaction: linking & forms
- Other semantics:
  - do not describe the document,
    but the *domain* of the document's *content*
  - can still be related to document
    - annotations & meta data
  - RDF(S), OWL, etc.



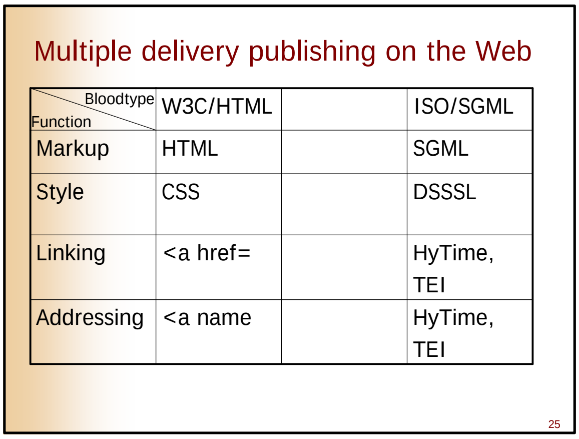
Talk overview

- Introduction and historical background
- Multiple delivery publishing (MDP)
- MDP on the Web: Style sheets
- 2nd generation multimedia - Cuypers
- Towards 3rd generation multimedia



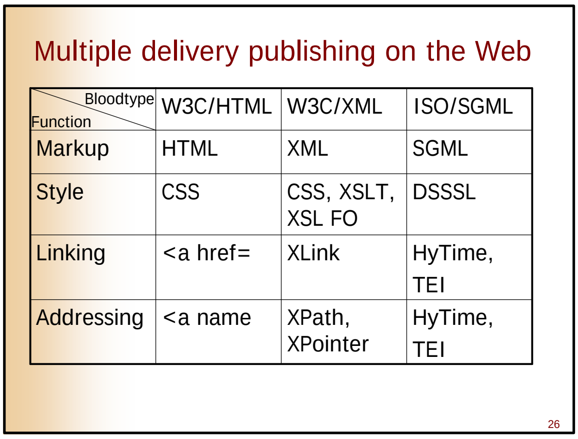
Multiple delivery publishing on the Web

Bloodtype Function	W3C/HTML		
Markup	HTML		
Style	CSS		
Linking	<a href=		
Addressing	<a name		



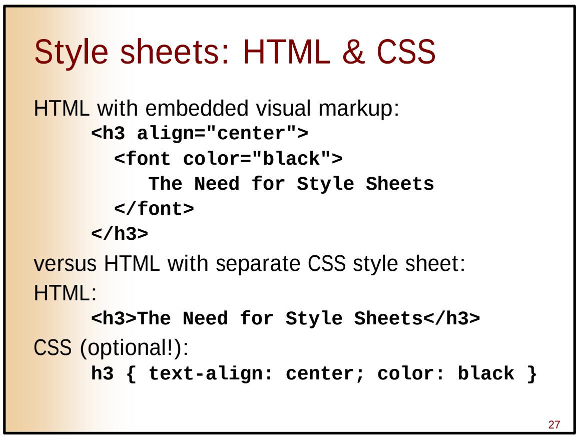
Multiple delivery publishing on the Web

Bloodtype Function	W3C/HTML		ISO/SGML
Markup	HTML		SGML
Style	CSS		DSSSL
Linking	<a href=		HyTime, TEI
Addressing	<a name		HyTime, TEI



Multiple delivery publishing on the Web

Bloodtype Function	W3C/HTML	W3C/XML	ISO/SGML
Markup	HTML	XML	SGML
Style	CSS	CSS, XSLT, XSL FO	DSSSL
Linking	<a href=	XLink	HyTime, TEI
Addressing	<a name	XPath, XPointer	HyTime, TEI



Style sheets: HTML & CSS

HTML with embedded visual markup:

```
<h3 align="center">
  <font color="black">
     The Need for Style Sheets
  </font>
</h3>
```

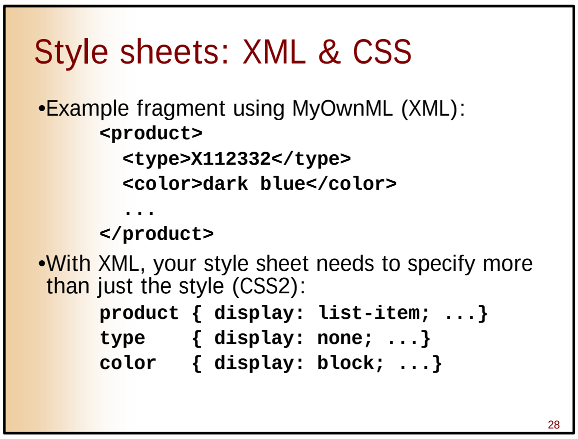
versus HTML with separate CSS style sheet:

HTML:

```
<h3>The Need for Style Sheets</h3>
```

CSS (optional!):

```
h3 { text-align: center; color: black }
```



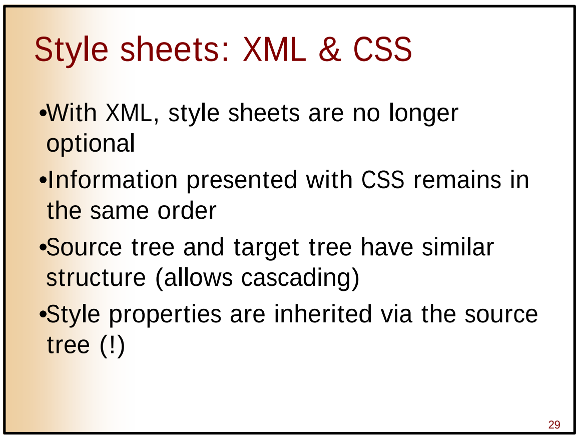
Style sheets: XML & CSS

- Example fragment using MyOwnML (XML):

```
<product>
  <type>X112332</type>
  <color>dark blue</color>
  ...
</product>
```

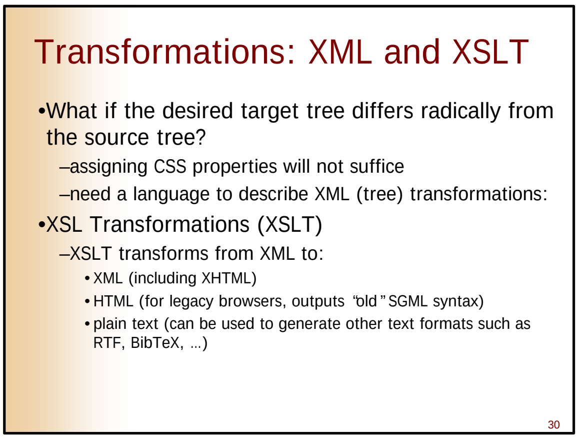
- With XML, your style sheet needs to specify more than just the style (CSS2):

```
product { display: list-item; ...}
type    { display: none; ...}
color   { display: block; ...}
```



Style sheets: XML & CSS

- With XML, style sheets are no longer optional
- Information presented with CSS remains in the same order
- Source tree and target tree have similar structure (allows cascading)
- Style properties are inherited via the source tree (!)



Transformations: XML and XSLT

- What if the desired target tree differs radically from the source tree?
  - –assigning CSS properties will not suffice
  - –need a language to describe XML (tree) transformations:
- XSL Transformations (XSLT)
  - –XSLT transforms from XML to:
    - XML (including XHTML)
    - HTML (for legacy browsers, outputs “old ” SGML syntax)
    - plain text (can be used to generate other text formats such as RTF, BibTeX, ...)

Transformations: XML and XSLT

- XSLT itself also uses XML syntax (unlike CSS …)
  - so you can transform XSLT using XSLT…
  - … but it doesn t look really human friendly!
- The structure of the target tree and source tree can differ (unlike CSS):
  - XSLT style sheets can be chained, not cascaded

XSLT template rules

- Transformations are described as a set of one or more template rules
- Each template rule consists of two parts:
  - A pattern that is matched against the source tree: the selector
  - A template to be filled in and added to the result tree
- XSLT selectors are based on XPath, e.g:

```
–product           /product
–color|type        product/color
–catalog//product  text()
–id("W11")         product[1]
–@class            /   *   @*
```

XSLT: Example (I)

A single template rule may be sufficient…

```
<xsl:template match="/">
  <html>
   <head>
    <title>Product Report Summary</title>
    </head><body>
    <p>...<table>
     <tr><td>
      <xsl:value-of select="product/type"/>
     </td><td>
      <xsl:value-of select="product/color"/>
     <td> ... </tr></table>...</body></html>
</xsl:template>
```

XSLT: Example (II)

… or a style sheet can contain many (smaller) template rules

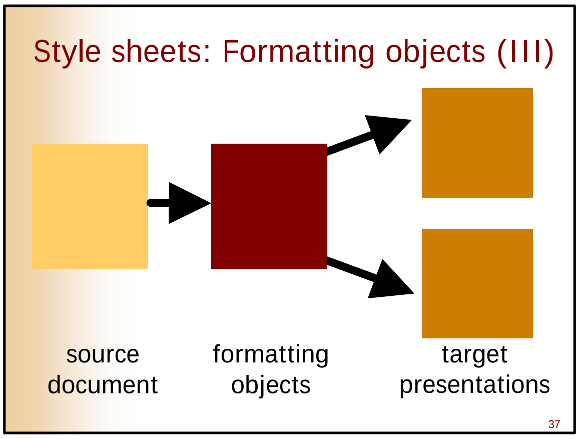
```
<xsl:template match="/">
 <table>
  <xsl:apply-templates/>
 </table>
</xsl:template>

<xsl:template match="product">
 <tr>
  <xsl:apply-templates/>
 </tr>

</xsl:template>
 <xsl:template match="color|type">
  <td>
   <xsl:apply-templates/>
  </td>
 </xsl:template>
...
```

Style sheets: Formatting objects (I)

- All these style sheet examples actually do two things:
  - specify how an XML document should be presented
  - specify how that presentation should be encoded in HTML
- Drawbacks:
  - need to start all over again for target formats other than HTML
  - limited by the presentation capabilities of HTML & CSS

Style sheets: Formatting objects (II)

- Solution:
  - design new target language (argh!)
  - a language that is designed to describe formatting semantics
  - such a language is called a formatting vocabulary
  - elements in the language are called formatting objects (FO)
- Example: the formatting vocabulary defined by XSL
  - fo:block, fo:flow, fo:footnote, fo:external-graphic, fo:page-sequence
- XSL well suited for on-line and paper-based formatting beyond HTML

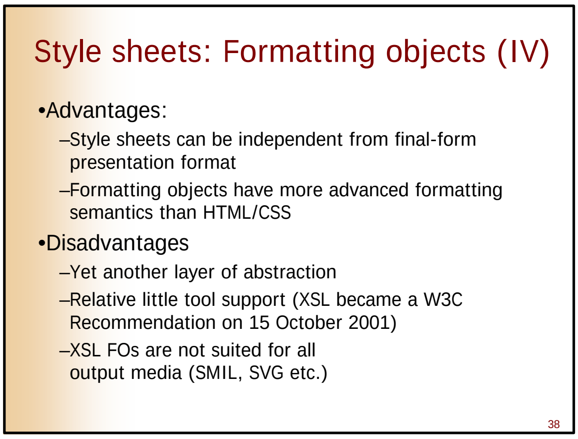


Style sheets: Formatting objects (III)

source document

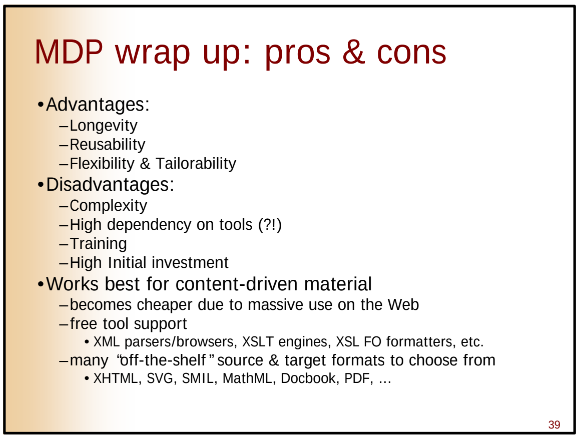
formatting objects

target presentations



Style sheets: Formatting objects (IV)

- Advantages:
  - Style sheets can be independent from final-form presentation format
  - Formatting objects have more advanced formatting semantics than HTML/CSS
- Disadvantages
  - Yet another layer of abstraction
  - Relative little tool support (XSL became a W3C Recommendation on 15 October 2001)
  - XSL FOs are not suited for all output media (SMIL, SVG etc.)

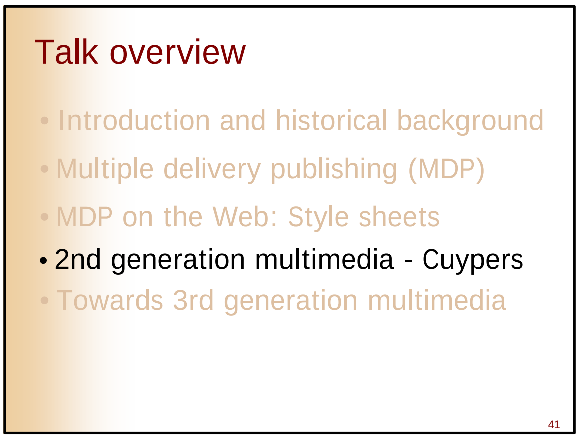


MDP wrap up: pros & cons

- Advantages:
  - Longevity
  - Reusability
  - Flexibility & Tailorability
- Disadvantages:
  - Complexity
  - High dependency on tools (?!)
  - Training
  - High Initial investment
- Works best for content-driven material
  - becomes cheaper due to massive use on the Web
  - free tool support
    - XML parsers/browsers, XSLT engines, XSL FO formatters, etc.
  - many "off-the-shelf" source & target formats to choose from
    - XHTML, SVG, SMIL, MathML, Docbook, PDF, ...

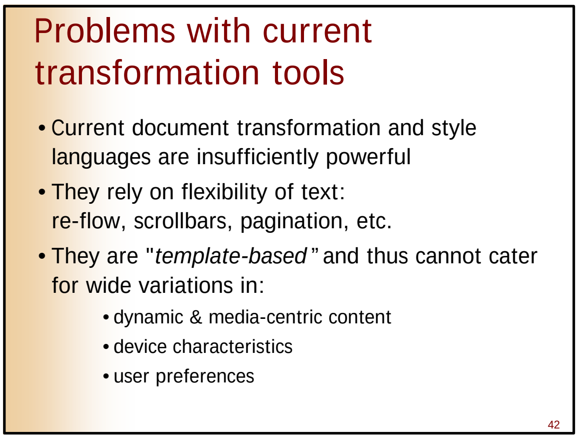
Further reading

- Overview pages at www.w3.org:
  - http://www.w3.org/XML/
  - http://www.w3.org/Style/XSL/
  - http://www.w3.org/Style/CSS/
- Recommendations (+ drafts) at www.w3.org/TR/:
  - http://www.w3.org/TR/xsl
  - http://www.w3.org/TR/xslt
  - http://www.w3.org/TR/REC-xml
  - http://www.w3.org/TR/REC-CSS2
- Tutorials and more
  - http://www.xml.com
  - http://www.mulberrytech.com/
  - http://www.mulberrytech.com/quickref/ (Jacco's personal favorite)



Talk overview

- Introduction and historical background
- Multiple delivery publishing (MDP)
- MDP on the Web: Style sheets
- 2nd generation multimedia - Cuypers
- Towards 3rd generation multimedia



Problems with current transformation tools

- Current document transformation and style languages are insufficiently powerful
- They rely on flexibility of text: re-flow, scrollbars, pagination, etc.
- They are "*template-based*" and thus cannot cater for wide variations in:
  - dynamic & media-centric content
  - device characteristics
  - user preferences

Multimedia on the Web

- Real multimedia Web content is still rare
  - Mostly bells & whistles to enhance HTML text …
  - … or mono-media AV-streams
- Virtually all presentations are hand-authored
  - proprietary formats that are hard to generate
  - limited support for dynamic content and multichanneling
  - most Web technology is text/page-oriented …
  - … with SMIL as one of the few exceptions

✍Conclusion:

Multimedia has hardly caught up with the 1st generation Web!

2nd generation multimedia

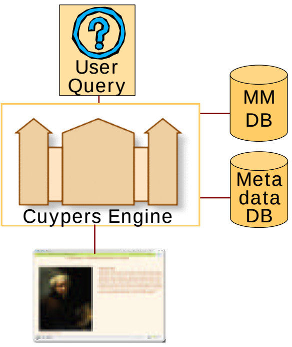
- Adapt to end-user’s platform capabilities
  - PC, PDA, mobile, voice-only, …
- Adapt to the network resources available
  - bandwidth and other quality of service parameters
- Personalization
  - language, abilities, level of expertise, ..

•Problem: current 2nd generation Web tools **do not work for multimedia**

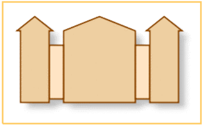
Multimedia differs from text

- Different document and presentation abstractions
  - hard to separate style from structure
- Formatting is not based on text flow
  - no pages or scrollbars, no line-breaking or hyphenation
  - templates often do not work well either
- Feedback from the formatting back-end required
  - need to check whether proposed layout is feasible
  - layout of media items is less flexible than text layout
- Transformations are hard in a functional language
  - need to try out designs and backtrack when necessary

Cuypers Engine



Cuypers multimedia generation engine



Demo time

Cuypers – the bad news

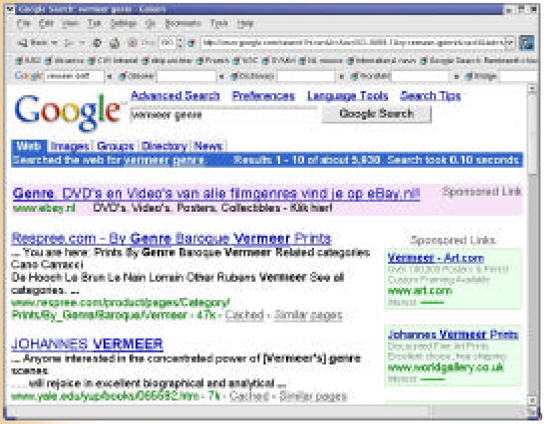
Currently all our design knowledge is:

- implicit and hidden in the generation rules
- lost in the generated Web presentation
- not reusable for other Web applications/sites

We need the Semantic Web

Talk overview

- Introduction and historical background
- Multiple delivery publishing (MDP)
- MDP on the Web: Style sheets
- 2nd generation multimedia - Cuypers
- Towards 3rd generation multimedia



Searched the web for vermeer genre. Results 1 - 10 of about 5,830. Search took 0.10 seconds.

Genre. DVD's en Video's van alle filmgenres vind je op eBay.nl! Sponsored Link
www.ebay.nl DVD's, Video's, Posters, Collectibles - Klik hier!

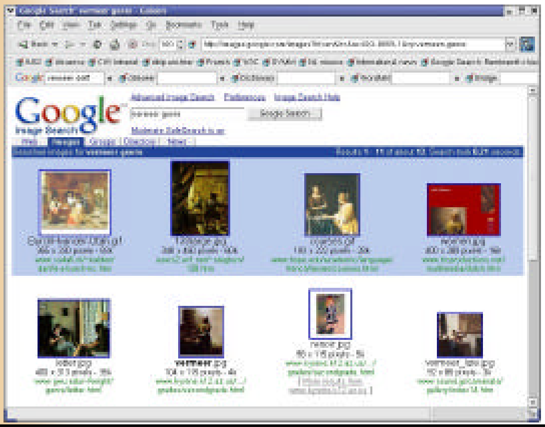
Respree.com - By Genre Baroque Vermeer Prints
... You are here: Prints By Genre Baroque Vermeer Related categories: Cano Carracci
De Hooch Le Brun Le Nain Lorrain Other Rubens Vermeer See all categories. ...
www.respree.com/productpages/Category/Prints/By_Genre/Baroque/Vermeer - 47k - Cached - Similar pages

JOHANNES VERMEER
... Anyone interested in the concentrated power of [Vermeer's] genre scenes
... will rejoice in excellent biographical and analytical ...
www.yale.edu/yup/books/066582.htm - 7k - Cached - Similar pages

Sponsored Links

Vermeer - Art.com
Over 100,000 Posters & Prints
Custom Framing Available
www.art.com

Johannes Vermeer Prints
Discounted Fine Art Prints
Excellent choice, free shipping
www.worldgallery.co.uk

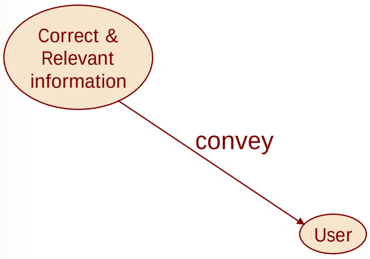


letter.jpg

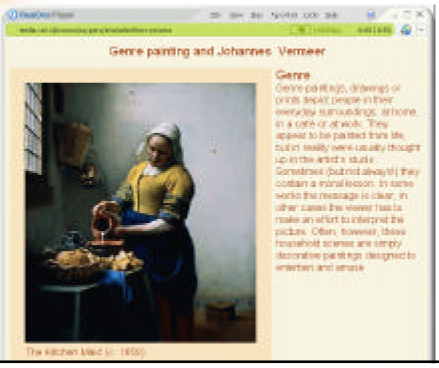
vermeer.jpg

vermeer_late.jpg

Presentation Generation



Tailored presentation



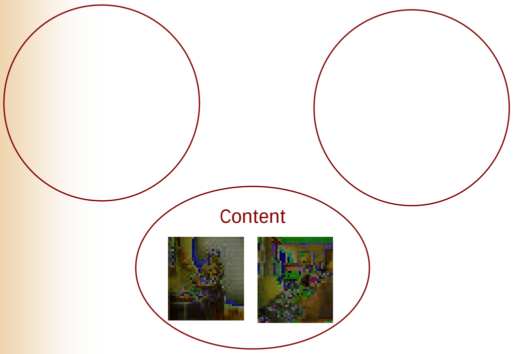
Genre painting and Johannes Vermeer

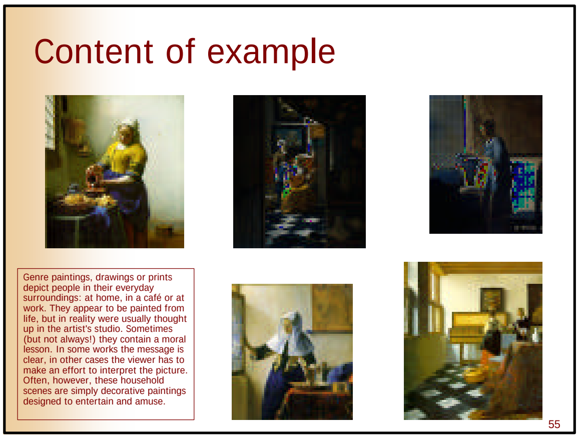
The Kitchen Maid (c. 1658)

Genre

Genre paintings, drawings or prints depict people in their everyday surroundings, at home, in a cafe or at work. They appear to be painted from life, but in reality were usually thought up in the artist's studio. Sometimes (but not always!) they contain a moral lesson. In some works the message is clear, in other cases the viewer has to make an effort to interpret the picture. Often, however, these household scenes are simply decorative paintings designed to entertain and amuse.

Three ingredients

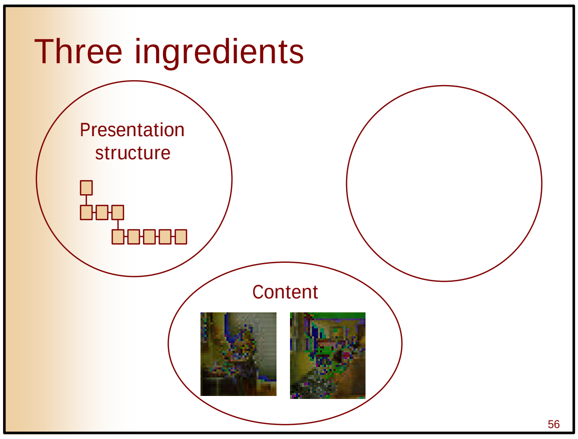




Content of example

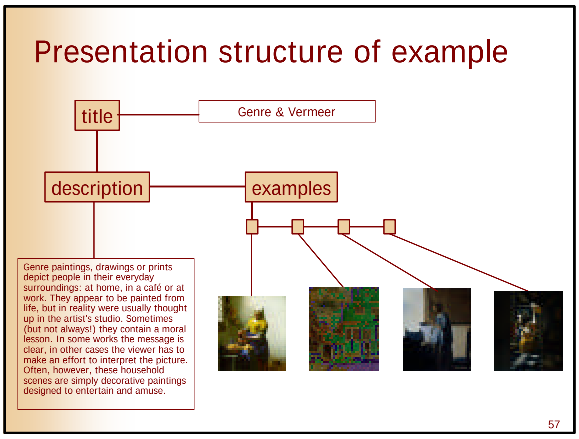
Genre paintings, drawings or prints depict people in their everyday surroundings: at home, in a café or at work. They appear to be painted from life, but in reality were usually thought up in the artist's studio. Sometimes (but not always!) they contain a moral lesson. In some works the message is clear, in other cases the viewer has to make an effort to interpret the picture. Often, however, these household scenes are simply decorative paintings designed to entertain and amuse.

55



Three ingredients

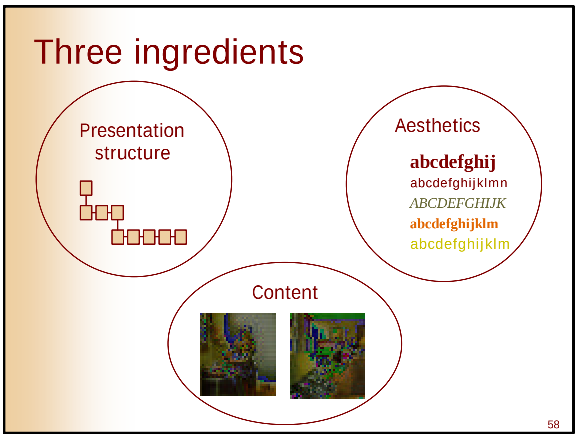
56



Presentation structure of example

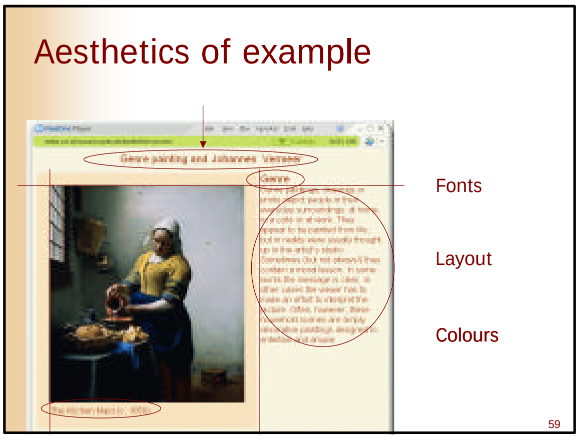
Genre paintings, drawings or prints depict people in their everyday surroundings: at home, in a café or at work. They appear to be painted from life, but in reality were usually thought up in the artist's studio. Sometimes (but not always!) they contain a moral lesson. In some works the message is clear, in other cases the viewer has to make an effort to interpret the picture. Often, however, these household scenes are simply decorative paintings designed to entertain and amuse.

57



Three ingredients

58



Aesthetics of example

Genre painting and Johannes Vermeer

Genre

Genre paintings, drawings or prints depict people in their everyday surroundings: at home, in a café or at work. They appear to be painted from life, but in reality were usually thought up in the artist's studio. Sometimes (but not always!) they contain a moral lesson. In some works the message is clear, in other cases the viewer has to make an effort to interpret the picture. Often, however, these household scenes are simply decorative paintings designed to entertain and amuse.

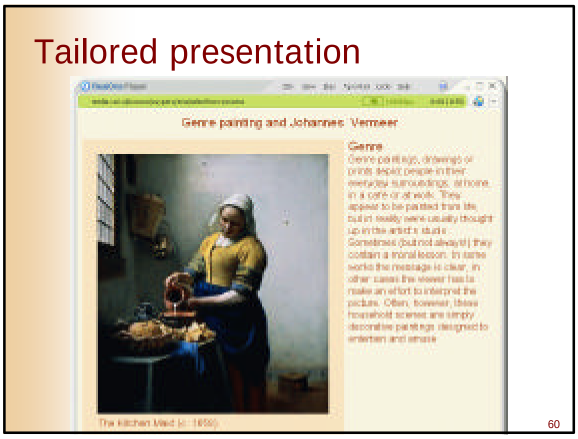
The Kitchen Maid (c. 1658)

Fonts

Layout

Colours

59



Tailored presentation

Genre painting and Johannes Vermeer

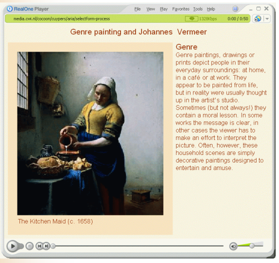
Genre

Genre paintings, drawings or prints depict people in their everyday surroundings: at home, in a café or at work. They appear to be painted from life, but in reality were usually thought up in the artist's studio. Sometimes (but not always!) they contain a moral lesson. In some works the message is clear, in other cases the viewer has to make an effort to interpret the picture. Often, however, these household scenes are simply decorative paintings designed to entertain and amuse.

The Kitchen Maid (c. 1658)

60

One size *doesn t* fit all



RealOne Player

media.cwi.nl/cocoon/cuypers/aria/selectform-process

Genre painting and Johannes Vermeer

The Kitchen Maid (c. 1658)

Genre

Genre paintings, drawings or prints depict people in their everyday surroundings: at home, in a café or at work. They appear to be painted from life, but in reality were usually thought up in the artist's studio. Sometimes (but not always!) they contain a moral lesson. In some works the message is clear, in other cases the viewer has to make an effort to interpret the picture. Often, however, these household scenes are simply decorative paintings designed to entertain and amuse.

What we need is…

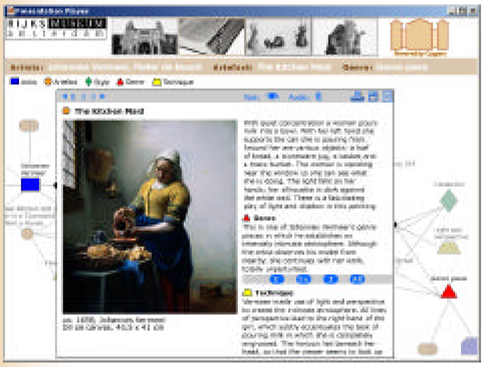
Correct and relevant information presented to the user

- conveying underlying relations in the subject matter
- at a suitable level of detail
- in the time available to the user
- using appropriate media
- in an appropriate style
- making optimal use of the delivery context

Create models…

Need	Knowledge of…
Convey underlying relations	Domain
Suitable level of detail	Discourse
Time available to user	User task
Appropriate media	Media characteristics
Appropriate style	Graphic design
Device characteristics	Device capabilities

Encyclopaedia



The Kitchen Maid

[illegible]

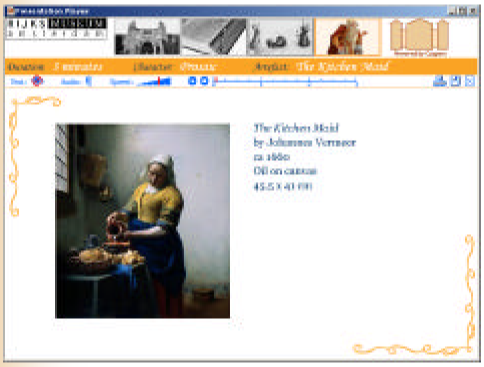
Genre

[illegible]

Technique

[illegible]

Storyteller



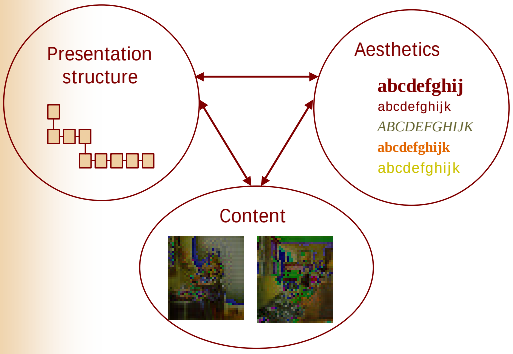
The Kitchen Maid
by Johannes Vermeer
ca 1660
Oil on canvas
45.5 x 41 cm

Create models…

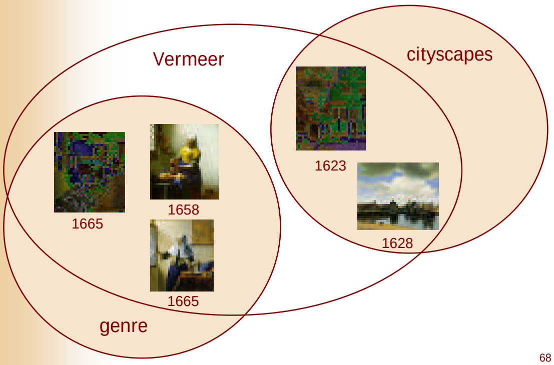
Need	Knowledge of…
Convey underlying relations	Domain
Suitable level of detail	Discourse
Time available to user	User task
Appropriate media	Media characteristics
Appropriate style	Graphic design
Device characteristics	Device capabilities

…to drive the creation of presentations

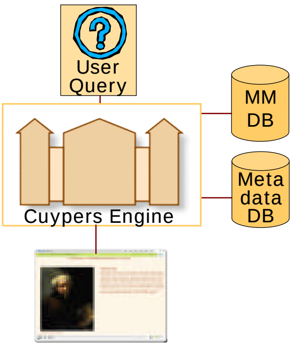
Design dependencies



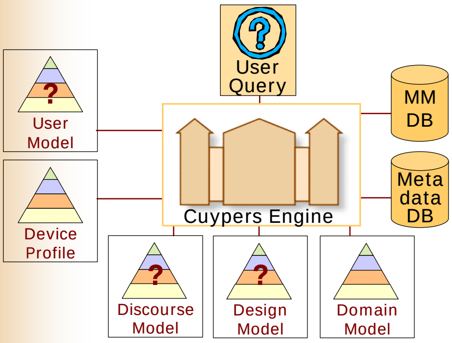
Presentation structure depends on content



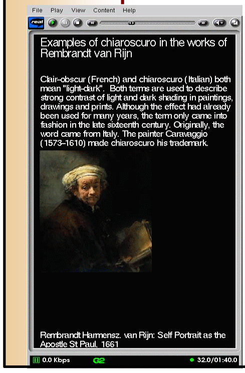
Cuypers Engine



Cuypers Knowledge Sources



Example scenario



Examples of chiaroscuro in the works of Rembrandt van Rijn

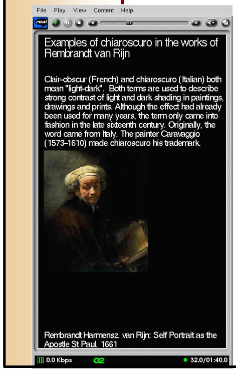
Clair-obscur (French) and chiaroscuro (Italian) both mean "light-dark". Both terms are used to describe strong contrast of light and dark shading in paintings, drawings and prints. Although the effect had already been used for many years, the term only came into fashion in the late sixteenth century. Originally, the word came from Italy. The painter Caravaggio (1573–1610) made chiaroscuro his trademark.

Rembrandt Harmensz. van Rijn: Self Portrait as the Apostle St Paul, 1661

•User is interested in Rembrandt and wants to know about about the "chiaroscuro" technique

•System responds with textual explanation of the technique and a number of example images of its application in Rembrandt's paintings

Example - Annotated SMIL



Examples of chiaroscuro in the works of Rembrandt van Rijn

Clair-obscur (French) and chiaroscuro (Italian) both mean "light-dark". Both terms are used to describe strong contrast of light and dark shading in paintings, drawings and prints. Although the effect had already been used for many years, the term only came into fashion in the late sixteenth century. Originally, the word came from Italy. The painter Caravaggio (1573–1610) made chiaroscuro his trademark.

Rembrandt Harmensz. van Rijn: Self Portrait as the Apostle St Paul, 1661

Combine

Content
    text, images
Document structure
    SMIL: timing, layout, links
"Message"
    Rembrandt is-a painter

Using an existing ontology

See http://www.cwi.nl/~media/semantics/

```
<?xml version="1.0"?>
<!-- taken from
 http://www.ics.forth.gr/proj/isst/RDF/RQL/rql.html
 -->
<rdf:RDF xml:lang="en"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/TR/2000/CR-rdf-schema-20000327#"
 xmlns="">

   <rdfs:Class rdf:ID="Artist"/>
   <rdfs:Class rdf:ID="Artifact"/>
   <rdfs:Class rdf:ID="Museum"/>
   <rdfs:Class rdf:ID="Painter">
      <rdfs:subClassOf rdf:resource="#Artist"/>
   </rdfs:Class>
   <rdfs:Class rdf:ID="Painting">
      <rdfs:subClassOf rdf:resource="#Artifact"/>
   </rdfs:Class>
...
</rdf:RDF>
```

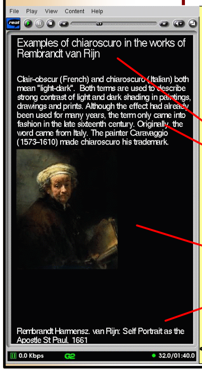
Embedding RDF in SMIL - I

```
<smil xmlns="http://www.w3.org/2000/SMIL20/CR">
  <head>
    <meta name="generator" content="CWI/Cuypers 1.0"/>
    <metadata>
      <rdf:RDF xml:lang="en"
               xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
               xmlns:oil="http://www.ontoknowledge.org/oil/…"
               xmlns:museum="http://ics.forth.gr/.../museum.rdf"

        <museum:Museum rdf:ID="Rijksmuseum" />

        <museum:Painter rdf:ID="Rembrandt">
          <museum:fname>Rembrandt</museum:fname>
          <museum:lname>Harmenszoon van Rijn</museum:lname>
          <museum:paints rdf:resource="#apostlePaul" />
        </museum:Painter>

        <museum:Painting rdf:about="#apostlePaul">
          <museum:exhibited rdf:resource="#Rijksmuseum" />
          <museum:technique>chiaroscuro</museum:technique>
        </museum:Painting>
      </rdf:RDF>
    </metadata>
    . . .
```

Embedding RDF in SMIL - II

```
<museum:Painting rdf:about="#apostlePaul">
          <museum:exhibited rdf:resource="#Rijksmuseum" />
          <museum:technique>chiaroscuro</museum:technique>
          <token:painted-by rdf:resource="#Rembrandt" />
        </museum:Painting>
      </rdf:RDF>
    </metadata>
...
  </head>
  <body>
    <par>
      <text region="title" src="...query to MM DBMS..."/>
      <text region="descr" src="..."/>
      <seq>
        <par dur="10"> ... 1st painting+title ... </par>
        <par dur="10"> ... 2nd painting+title ... </par>
        <par dur="10"> ... 3rd painting+title ... </par>
        <par dur="10"> ... 4th painting+title ... </par>
        <par dur="10" id="apostlePaul">
          <img  region="img" src="..."/>
          <text region="ptitle" src=".."/>
        </par>
      </seq>
    </par>
  </body>
</smil>
```

Marked-up presentation



Examples of chiaroscuro in the works of Rembrandt van Rijn

Clair-obscur (French) and chiaroscuro (Italian) both mean "light-dark". Both terms are used to describe strong contrast of light and dark shading in paintings, drawings and prints. Although the effect had already been used for many years, the term only came into fashion in the late sixteenth century. Originally, the word came from Italy. The painter Caravaggio (1573–1610) made chiaroscuro his trademark.

Rembrandt Harmensz. van Rijn: Self Portrait as the Apostle St Paul, 1661

```
 <museum:Painting rdf:about="#apostlePaul">
        <museum:exhibited rdf:resource="#Rijksmuseum"/>
        <museum:technique>chiaroscuro</museum:technique
 >
 </museum:Painting>
  </rdf:RDF>
  </metadata>
...
  </head>
  <body>
    <par>
      <text region="title" src="..."/>
      <text region="descr" src="..."/>
      <seq>
        <par dur="10"> ... </par>
        <par dur="10"> ... </par>
        <par dur="10"> ... </par>
        <par dur="10"> ... </par>
        <par dur="10" id="apostlePaul">
          <img  region="img" src="..."/>
          <text region="ptitle" src=".."/>
        </par>
      </seq>
    </par>
  </body>
</smil>
```

Conclusions

- Generating multimedia presentations requires
  - making design dependencies explicit
  - taking these dependencies into account
- Semantic Web has the potential to
  - encode knowledge we use
  - reuse knowledge already available
- Cuypers system
  - provides an experimentation platform
  - using standard tools and languages as much as possible