Evolution in Web and Web data
Hypermedia

- Hypertext + multimedia, alternative to hypertext
- Information *objects*: text, images, animations, audio, video
- Not possible to show everything at once:
  - *Layout*
  - *Timing*
  - *Navigation*
- Design (generate) presentation for different *platforms*: WWW, phone, PDA, etc.
  - Majority of surfers will not be using (the same) PC
Evolution in hypermedia

• First: standalone special-purpose systems
• Now: Web-based
  – From *authoring* to *designing* to *generating*
  – From *static* to *dynamic* (generated from database query result)
  – From *single site* to *portals* (integrated access service)
  – From *read-only* to *interactive* and often *collaborative* (read-write)
Research (Hera)

• In WIS how to (automatically) generate hypermedia access to the information?
  – Hypermedia access: \textit{navigation}
  – Personalization, customization: \textit{adaptation}
  – Information \textit{integration}: (logistics of) information retrieval from available, heterogeneous sources
  – Interoperability \textit{w.r.t. dynamics}: for e-business, business processes and service integration
  – \textit{Querying} and \textit{transforming} of data and metadata

Ref: wwwis.win.tue.nl/~hera
Three Generations of Web-based Systems

1. HTML written by *author*
   - Easy, uniform interface
   - Large effort for maintenance
   - Not suited for changing information

2. Automatically *generating* information
   - First, using templates (and databases)
   - Later, using XML and XSLT transformations

3. Automatic *processing* of information
   - Explicit metadata (RDF)
   - Agreement on meaning (ontologies)
Future “development”

• Academic research is now at third step from previous slide.
• TBL has sketched his ideas for the next Web.
• Semantic Web research aims at even further developments in the Web and Web applications
  • From human-readable via machine-readable to machine-processable
Common Syntax: XML

• *HTML*: a fixed set of tags complicates the identification of information elements

• *XML* allows to define data structures:
  – Tags with freely chosen names
    • No predefined tags enables definition, transmission, validation and interpretation of data between applications (and organizations)
  – Freely chosen attributes
  – Simple definition: DTD
  – Extended, richer definition: XML Schema

• XPath and XQuery for querying
<skills>
  <people>
    <person>
      <name>Bob</name>
      <know-how>XML</know-how>
    </person>
    <person>
      <name>Peter</name>
      <know-how>XML</know-how>
      <know-how>RDF</know-how>
    </person>
  </people>
  <seminars>
    <seminar>
      <topic>XML</topic>
      <participant>
        <name>Karin</name>
        <name>Alice</name>
      </participant>
    </seminar>
  </seminars>
</skills>
Specification of meaning: RDF

- **Resource**: denotes an information item, e.g. via a URL
- **Property type**: name of a property of a resource
- **Value**: value for that property

Example:

```
Resource = http://www.example.org/index.html
Property type = http://purl.org/dc/elements/1.1/creator
Value = http://www.example.org/staffid/85740
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
    <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
        <contact:fullName>Eric Miller</contact:fullName>
        <contact:mailbox rdf:resource="mailto:em@w3.org"/>
        <contact:personalTitle>Dr.</contact:personalTitle>
    </contact:Person>
</rdf:RDF>
Meaning: ontologies

- **Ontology** = a vocabulary with associated meaning
  - Meaning expressed via relationships between concepts (no relationships with "real world")
- Possibility to define synonyms, specializations and other *relationships*
- Use of same ontology = *contract* on meaning of words (tags, attributes)
  - Often, industry or domain dependent
Future of the Web

1. Common syntax (XML)
2. "Specification" of meaning (RDF)
3. "Agreement" on meaning (ontologies)

4. Use logic to derive conclusions
   - Necessary in e-commerce: How does the supplier's message relate to the customer's knowledge?

5. Goal: trust in communication between Web systems, and hence the possibility to automate using agents

Ref: w3.org
“Layer Cake”
WIS & Metadata

• Whether or not this original vision becomes reality, metadata about data is very relevant and useful for information system development.

• WIS repository (back-end) typically assembled from different heterogeneous sources, e.g. databases, files, WWW

• To manage (coordinate) data from different sources, metadata helps to organize and structure the data
Metadata

- Describing the data:
  - Data about data
  - Definitional data providing information about or documentation of data managed in an application

- Sometimes provided by sources
  - Obtaining (good) metadata becomes more and more relevant, and even a goal of its own.

- Needed by IS, to specify the “logistics” of data, i.e. what needs to be done with the data
  - Meaning
  - Validity
  - Quality
After the first WWW

• The evolution of the WWW is **not** a managed process.
• Several independent developments, seen now.
Web 2.0

• O’Reilly (conference in 2004)
• Two phenomena, related to (constant) creativity/innovation and (social) collaboration
• Internet as the (software) platform
  – Note: company-oriented; TBL and W3C question the innovation
Web 2.0 Apps

- eBay
- Wikipedia
- del.icio.us
- Flickr
- Writely (later Google Docs)
- Google Maps
Web 2.0

• We see a large social aspect
• We also see a set of APIs that go beyond read-only access
• Rich User Interface, Rich Internet Applications
  – AJAX
  – started (its success) with Google (Maps)
  – different separation of concerns between server and user interface
Web 2.0: user-participation

- Main phenomenon is user-participation: from read-only to contributing content (and metadata)
- Early examples: Flickr, del.icio.us, last.fm
- Tagging (user-generated metadata)
- Later, Facebook and other Social Web apps
- Good for the people, but brings a new dimension to WIS development.
Web 3.0

- The academics have seen this development with Web 2.0, have observed that it is about programming and about social issues, and have claimed that the real next step is in the further development of the metadata-based part of the Web.
- Semantic Web = Web 3.0
- From the original vision, the initiative Linked Data is currently the most promising.
Scope

• We will consider “Semantic Web” and “Web 2.0” further in later parts of the course.
• We will concentrate on the aspects that relate to Web-based information systems.
WIS and Web Engineering
Web Engineering

• From ad-hoc development and programming to “real” engineering.
• It is about software and about data (content), and all the rest.
  – In current research tastes differ about priority.
• It is also about knowing what you do and establishing “evidence”.
• Still not a generally accepted approach.
Definition

• Web Engineering is the application of systematic, disciplined and quantifiable approaches to development, operation, and maintenance of Web-based applications.

• Note that Web Engineering is a moving target since Web technologies are constantly evolving, making new types of applications possible, which in turn may require innovations in how they are built, deployed and maintained.

Levels

- Web Page Construction
- Web Page Design
- Web Site Design
- Web Site Construction
- Web-based System
- Web project planning and management
Differences with Conventional SW

- compressed development schedules
- constant evolution with shortened revision cycles
- "content is king"
- insufficient requirement specifications
- small teams working to very short schedules
- emerging technologies/methodologies
- lack of accepted testing processes
- user satisfaction and the threat from one's competition
Different: Information

• WIS contain text and multimedia, which are difficult to structure, cannot be normalised and are very hard to sort and search.

• They mix document-orientation with database access through the hypertext metaphor.
Different: End-users

- WIS are not always confined to specific user groups within an organisation, but go world-wide.
- Difficulties in understanding the potential, unknown, and perhaps unknowable, users to establish the quality parameters of the applications (for quality, testing, security, etc.).
Characteristics

• Wide variety in applications
  – scale
  – time

• Multidisciplinary development
  – information science, graphic design, HCI, requirements engineering, network design, software engineering, legal, social, ethical
WIS Engineering issues

- Methodologies
- Requirements Elicitation
- Testing, Metrics and Quality
- Maintenance
Web Engineering

• Note that many active in the field of WIS Engineering refer to it as Web Engineering.
• Some take Web Engineering (also) as the engineering of the Web, and specially in current ideas from Web Science it is smart to check the scope and focus.
  – “The Web’s development is a mix of standard-setting, unstructured, decentralised activity and innovation, and deliberate engineering.”
  – services, distribution, P2P, personalization, multimedia, natural language processing
WIS Engineering Methodology

• Design of WIS requires careful engineering of information exchange between IS and OS
• Implies engineering of front-end (interface) and back-end (storage & retrieval)
• Professional applications: “from art to engineering”
  – well-founded (software) engineering methodologies
  – model-driven
More engineering issues

- *Personalization* (1-1)
- Multiple *output* devices
Modeling (originally)

- **Presentation level**
  - “Web” of pages

- **Storage level**
  - HTML pages, report and scripts, graphics, animation

- **Logical level**
  - E-R diagram (for storage)
  - Application diagram (geared towards presentation)
RMM

*Relationship Management Methodology*

- Entity-Relationship modeling
- Transformation from data model to data+navigation model
- RMDM: Relationship Management Data Model represents objects and navigational relationships (navigational design of application)
- Design method for Web (hypermedia) applications

RMM methodology

- Requirements analysis
- E-R diagram
- Application diagram (top-down)
- Slice design
- Application diagram (bottom-up)
- User interface design
- Implementation
RMDM

- (Application) *domain model* primitives
  - Entities
  - Attributes
  - Relationships
- *Slices*: from large objects (with many attributes) to smaller units (with coherent attributes, possibly from different objects)
- From *semantical* aspect to *navigational* (presentation) aspect (w.r.t. complexity and size)
Hera motivation

• Methodologies exist for manual hypermedia presentation design, Hera targets automated hypermedia presentation

• Automated presentation is important for databased content (the ‘deep web’) as opposed to manually crafted content (the ‘surface web’): most WIS are data-driven

• Presentations must be adaptable to different users/user platforms
Hera Methodology

- Model-driven methodology, defines design phases:
  - **Conceptual Design** that results in **Conceptual Model** (CM, describes data content used for generation of hypermedia presentations) construction
  - **Application Design** that results in **Application Model** (AM, describes the navigation structure and functionality) construction
  - **Presentation Design** that results in **Presentation Model** (PM, describes spatial layout and rendering of hypermedia presentations) construction
Hera Models

• Fully specify dynamic hypermedia applications; hence, there is *no* need of additional programming

• Are used by a generic Hera *engine* for generation of hypermedia application pages (by on-demand instantiations of model subsets)
Conceptual Model

- Provides a uniform *semantic* view over different data sources that are integrated within a given Web application.
- Consists of hierarchies of *concepts* relevant within the given domain, their properties, and relations.
Conceptual Model

- Defines the data content in terms of RDFS (concepts, attributes, properties)
Application (Navigation) Model

- *Navigation structure* of a hypermedia application on top of CM
- *Hypermedia dynamics* (navigation structure updates and application functionality) of a hypermedia application
Navigation Structure in Application Model

- *Navigation nodes* (pages) specification in terms of slices (collections of concepts’ attributes to be displayed)
- *Node composition* in terms of slice aggregation relationships
- *Navigation edges* (usually hyperlinks) in terms of slice references
Slices

- Meaningful collection of attributes of one or more related concepts
- Represent a presentation page or its part
Dynamics in Application Model

- *User input* specification in terms of Input Forms
- *Application context* (state) specification in terms of Application Context Model
- *Context manipulation* specification in terms of queries
Input Forms

- Specify *user-data entries*; contain sets of input fields with:
  - Input method (selection from offered items, text input, etc.)
  - How the offered items are created (for selections)
- Determine data *manipulation* operation associated with a form (form processing)
Application Context Model

- Extends CM with *additional* data structures needed for application functionality (to store application/navigation *state*, user inputs, user model, etc.)
- Example: storing the user selection (shopping basket)
Data Manipulations

- Update application context information
- Defined as SeRQL queries
- Used for processing forms (handle user input)

CONSTRUCT
  {P}<rdf:type>{acm:SelectedPainting>}
FROM
  {P}<rdf:type>{cm:Painting};
  <cm:aname>{Paname}
WHERE
  Paname IN SELECT Faname
  FROM    {SF}<form:aname>{Faname},
          {SF}<rdf:ID>{FormName}
WHERE FormName = “SelectForm”
Hera Architecture

Defines how the models are used for automatic generation of hypermedia presentation
Hera Implementations

• HPG 1.0 (Hera Presentation Generator, static version) implemented using XSLT
• HPG 2.0 (Hera Presentation Generator, dynamic version) implemented as a Java servlet
  – Uses Jena (RDF API from HP) for RDF data transformations based on RDFS models (CM, AM)
  – Can use XForms processor
  – Uses Sesame as main content repository and application context repository; uses SeRQL/RQL as query languages
  – Set of graphical tools for designers for CM and AM based on Visio or Java
• Hera-S: Sesame-based version

Device Adaptation

HTML

SMIL

WML
Adaptation and Personalization

• *Adaptation* (and *personalization*) is a design aspect that gained much more attention, but still is lightly supported in WE methodologies
  – difficult to specify
  – not many tools
  – difficult combination with other aspects
  – User Modeling is not easy

• Examples: My-portals, device-dependency
Adaptation Model

context independent

context dependent

pp:ImageCapable = Yes

um:ExpertiseLevel = Medium
• Object-Oriented Hypermedia Design Method
• Modeling/analysis, design, implementation, testing, and maintenance
• Conceptual Design + Navigation Design + Abstract Interface Design + Implementation
  – navigation objects are views of conceptual objects
  – abstractions to organize the navigation space, e.g. navigational contexts
  – separation of interface issues from navigation issues
  – some design decisions must only be made at implementation time
Other Methodologies

• **UWE: UML-based Web Engineering**
  – Conceptual, navigation, presentation model
  – Storyboarding and presentation flow

• **OOWS**
  – Web extension for OO software production method
  – Class diagram, navigational modeling, and presentational modeling

• **OO-H: Object Oriented Hypermedia**
  – Navigational access diagram, abstract presentation diagram

• **WebML: Web Modeling Language**
  – Known for its successful tool support and commercial applications
Assignment

Homework
Engineering

• In the context of the Web and WIS the term engineering is used to denote the structured, professional approach based on proven principles.

• Have a look at definitions of Engineering.

• Give your definition (the one you consider the best) of what Web (WIS) Engineering is (or should be). [>250w]
  – Consider your answer to the previous (small) assignment.
Model-driven

• Many approaches for Web application development (WIS) are considered to be “model-driven”. While this term has several interpretations, e.g. “model-based” and “following MDA”, it seems clear that this is a smart approach and the one advocated in academic research.

• What is model-driven (in this context)? [>250w]

• What are its pro’s and con’s, or how can it contribute to WIS development? [>500w]
WE Methods

• In conferences like ICWE you find a lot of work on methods for WE.
• Why do several methods (OOHDM, OOWS, etc.) “borrow” from Object Orientation and why does that seem smart? [>250w]
• Find out what are the main aspects (models) of WebML.
• Give WebML pro’s and con’s. [>250w]
Web services

• In Web development you also see a lot about “Web services”. It is a term that has been used for many different things, but now you see more of a consensus on it to refer to a way for software to communicate.

• Find out what is meant with Web services.

• What does it bring to Web application development? [>400w]
Individual work

• Answer these questions, in your own words, with proper references to the literature where appropriate, and integrate this into one text file, each answer starting on a new page, and create a PDF from this file.

• Send the PDF via email with subject “WIS homework 2” to g.j.houben@tue.nl by Sep. 26th (noon, strict). [30 hrs]