# Web Site Modelling and Temporal Issues

**Paolo Atzeni**
**Università Roma Tre**
`http://www.dia.uniroma3.it/~atzeni/`

**Redmond, January 2003**

## Focus

- **Preliminary thoughts based on my experience in models and schemes for Web sites and on the needs for effective management of temporal issues**

## Contents

- Data models for data-intensive Web sites
- Temporal databases
- Time in data-intensive Web sites
- Coordinates



## Contents

- Data models for data-intensive Web sites
- Temporal databases
- Time in data-intensive Web sites
- Coordinates

## Web-based information systems: a database point of view

- **Data-Intensive Web Sites**:
  - large amount of data
  - significance the hypertext structure



## Models and schemes in databases

- Almost forty years ago people realized that we often have records with the same structure; files with a rather fixed structure were introduced
- The notion of scheme of the database was later invented as an overall description of the content of a database
- Models were introduced to specify which schemes are allowed

## Models and schemes for hypertexts

- In data-intensive Web sites (and often in general) there are (many) pages with a similar (or even the same) structure



## Benefits of a model-based approach

- Synthetic description:
  - in reverse engineering (a posteriori modeling of existing Web sites):
    - support to data extraction, integration and querying (formulation and optimization)
  - in Web site development
    - separation of concerns: data, hypertext, presentation

# Example

- The **Integrated Web Museum**:
  a site integrating data coming from the Uffizi, Louvre and Capodimonte Web sites
- Developed within the **Araneus project**
  (Università Roma Tre and Università della Basilicata since 1996)

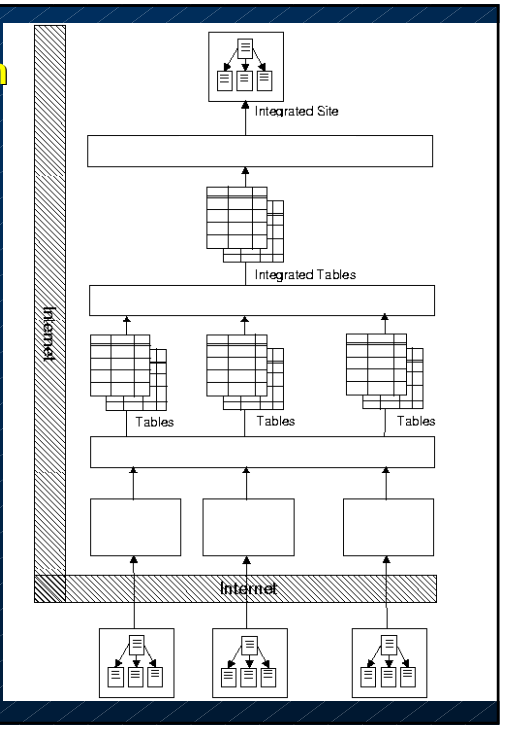
# The Integrated Web Museum

# Integration of Web Sites: The Integrated Web Museum

- Data are re-organized:
  - Uffizi, paintings organized by rooms
  - Louvre, Capodimonte, works organized by collections
  - Integrated Museum, organized by author

## The Araneus Approach

- Identification of sites of interest
- Wrapping of sites to extract information
- Navigation of site ad extraction of data
- Integration of data
- Generation of new sites



## Building Applications in Araneus

- Phase A:
  Reverse Engineering Existing Sites
- Phase B:
  Data Integration
- Phase C:
  Developing New Integrated Sites

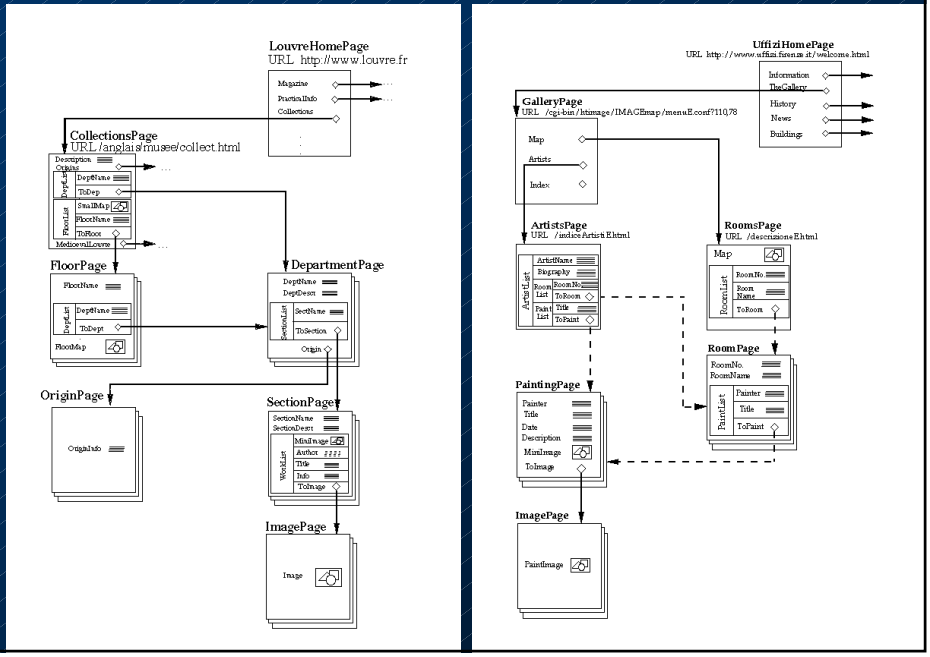
## Building Applications in Araneus
## Phase A: Reverse Engineering

- First Step: Deriving the logical structure of data in the site → ADM Scheme
- Second Step: Wrapping pages in order to map physical HTML sources to database objects
- Third Step: Extracting Data from the Site by Queries and Navigation

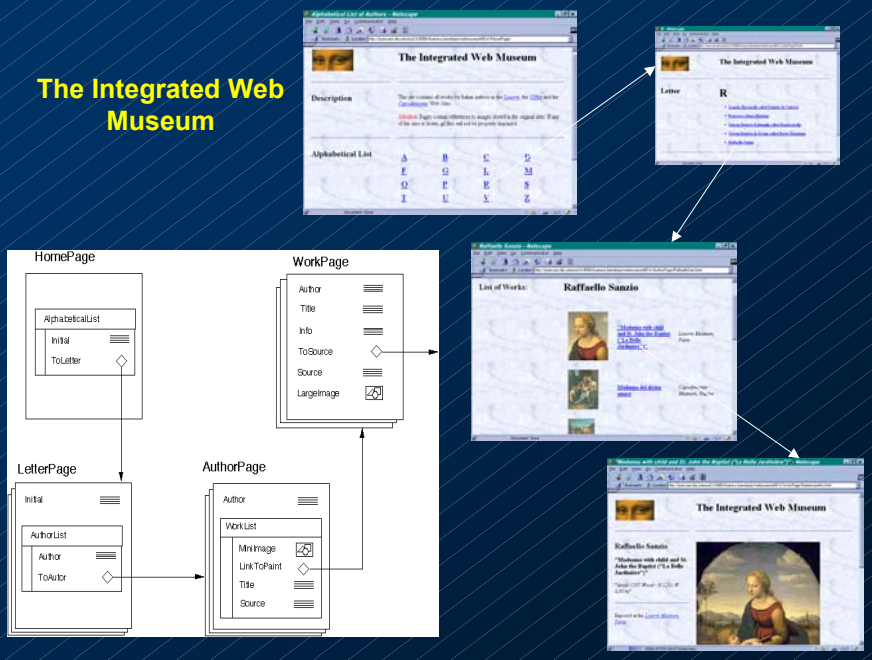
# Modeling Web Sites:
## The ARANEUS Data Model

- **ADM**
  - ODMG-like model
  - Pages are URL-identified nested objects
  - Heterogeneous Union-Types
- **Page-type: describes a set of homogeneous pages**
- **Site-scheme: set of page-types connected by links**

## The Integrated Web Museum



## The Integrated Web Museum




## Heavyweight or lightweight model?

- How much rigid should the structure be?
  - An open issue
- Extremes:
  - complex object database models
  - semistructured models

## Middleweight models

- ADM
  - heavyweight features:
    - page types, attributes, site scheme;
  - lightweight features:
    - union types, untyped links.

## Building Applications in Araneus
## Phase A: Reverse Engineering

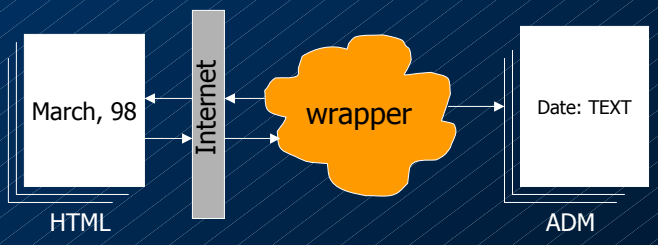
- First Step: Deriving the logical structure of data in the site → ADM Scheme
- Second Step: Wrapping pages in order to map physical HTML sources to database objects
- Third Step: Extracting Data from the Site by Queries and Navigation

## Information Extraction Task

- Information extraction task
  - source format: plain text with HTML markup (no semantics)
  - target format: database table or XML file (adding structure, i.e., "semantics")
  - extraction step: parse the HTML and return data items in the target format
- "Wrapper"
  - piece of software designed to perform the extraction step

## Wrapping Web Sites:
### The Araneus Wrapper Toolkit

- The need for wrappers



## Building Applications in Araneus
## Phase A: Reverse Engineering

- First Step: Deriving the logical structure of data in the site → ADM Scheme
- Second Step: Wrapping pages in order to map physical HTML sources to database objects
- Third Step: Extracting Data from the Site by Queries and Navigation

## Queries over Web Sites:
### Query Interfaces: Ulixes

Example of SQL Query: "*Titles and Years of Papers Published by Codd at SIGMOD* "

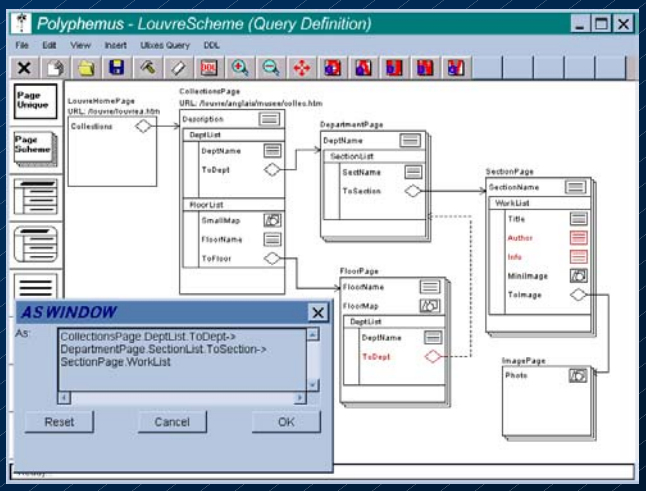
```
CREATE VIEW TitlesOfCoddPapers (Title, Year)
  OVER www.acm.org/sigmod AS

SELECT ProceedingsPage.SectionList.Articles.Title,
       HomePage.YearList.Year

FROM HomePage.YearList.NumberList.ToIsssues ->
     ProceedingsPage.SectionList.Articles

WHERE ProceedingsPage.SectionList.Articles.Authors
        LIKE '%Codd%'
```

# Queries over Web Sites:
## **Query Interfaces:** Polyphemus




# Building Applications in Araneus

- **Phase A:**
  **Reverse Engineering Existing Sites**
- **Phase B: Data Integration**
- **Phase C:**
  **Developing New Integrated Sites**

## Building Applications in Araneus

- Phase A:
  Reverse Engineering Existing Sites
- Phase B: Data Integration
- Phase C:
  Developing New Integrated Sites

## Building Applications in Araneus
## Phase C: Web Site Development

- Model-Based Development in Araneus
- CASE-Tool Approach (Homer)

## Model-Based Development in Araneus

- Distinction between design and implementation:
  - high-level models for site design
  - tools for site implementation
- Overall Goal:
  - users should design, not write code
- Flexibility in Site Implementation:
  - HTML, XML, WML
  - Independence from the actual page-generation tool

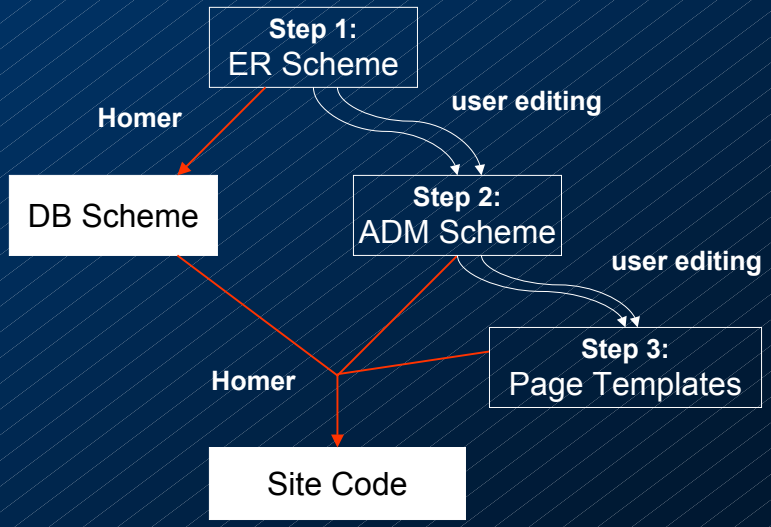
## Levels of representation in Web sites

- Data:
  - the information content
- Hypertext structure
  - how the data in arranged to form pages
- Presentation
  - layout, graphics, etc

## Model-Based Development in Araneus

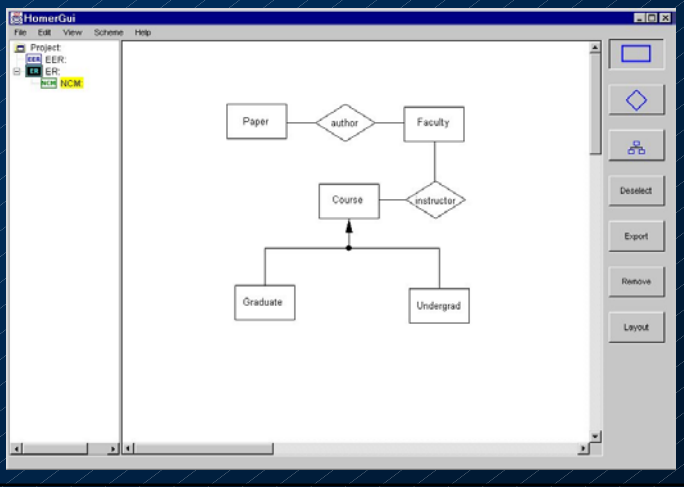
- High-Level Models:
  - data: relational (or object-relational)
  - hypertext: ADM
  - presentation: Telemachus Styles
- A design methodology
  - data design
  - hypertext design
  - presentation design

## Homer: A Case Tool for Web Sites

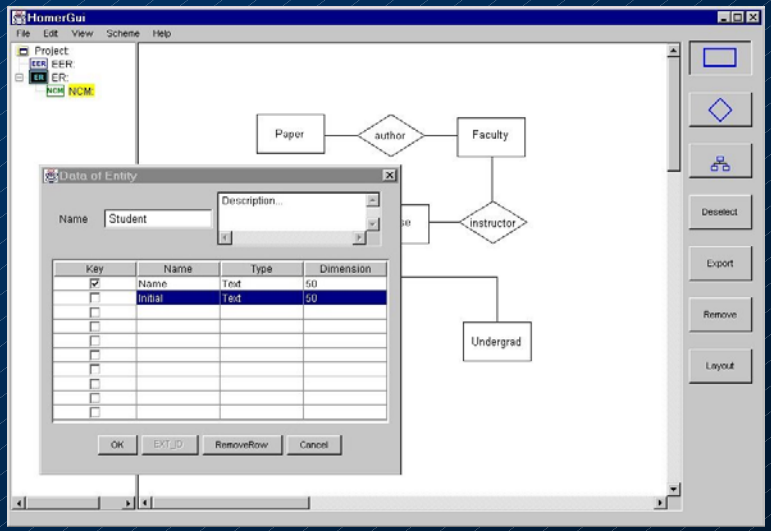


# Homer: A Case Tool for Web Sites

- **Step 1: User Draws an Entity-Relationship Scheme**

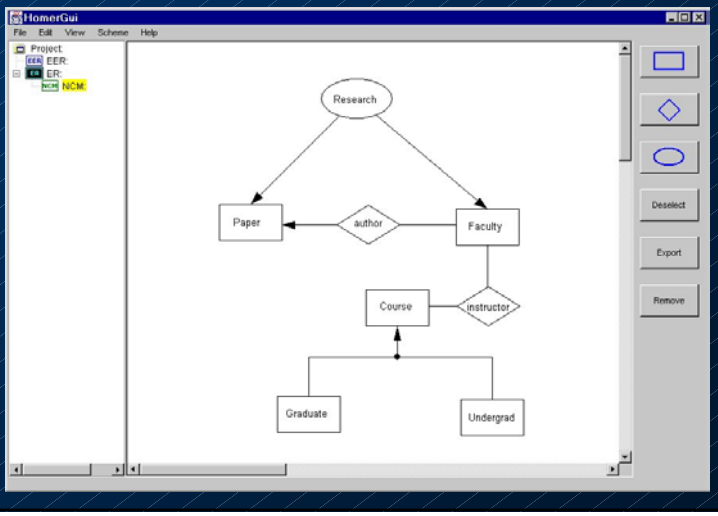



# Homer: A Case Tool for Web Sites



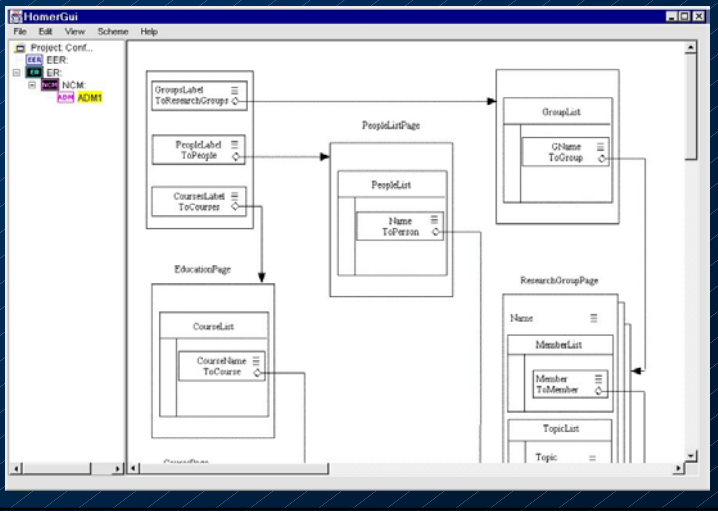
# Homer: A Case Tool for Web Sites

- **Step 2: ER Scheme ==> ADM Scheme;**




# Homer: A Case Tool for Web Sites

- **Step 2: ER Scheme ==> ADM Scheme;**



## Homer: A Case Tool for Web Sites

- Homer Engine:
  - intermediate models (ER, Relational ...) are seen as subsets of ADM
  - design steps are transformations (views)
  - nested-relational algebra with URL-invention
  - view composition
- Implementation:
  - HTML, XML with XSL stylesheets (WML)
  - Penelope, JSP, ...

## Presentation Modeling: Telemachus

- Requirements:
  - precise notion of style for page (schemes)
  - platform independence
  - rapid prototyping and flexible maintenance
  - working with “sample” pages (“templates”)
- Telemachus Styles:
  - Attribute Styles: formatting directives for attributes in pages
  - Page Styles:collections of attribute styles plus header and footer

## Contents

- Data models for data-intensive Web sites
- Temporal databases
- Time in data-intensive Web sites
- Coordinates



## Contents

- Data models for data-intensive Web sites
- Temporal databases
- Time in data-intensive Web sites
- Coordinates

## Temporal databases

- A database that records time-varying information (Jensen & Snodgrass 1999)
- Most database applications are intrinsically temporal, but traditional systems do not provide much explicit support to the management of time (usually only basic types for instants and intervals)
- Research on temporal databases aims at offering solutions



## Notions of time

- Valid time of a fact (represented by some data in a database): the time (instants or intervals) when the fact is true in the world
- Transaction time: the time when a fact is current in the database (the database knows about the fact)
- User-defined: any other of interest (with the semantics known only to the user/developer)

## Models with "identity"

- Conceptual models (such as ER) and object-based models are not "value-based" but "identity-based"
- For them time can profitably be associated with both
  - objects (entity instances), to denote the life-span of the object
  - attributes, to denote their time-varying values



## Professor as an entity type

- We could associate time information to
  - the instances of the entity (to denote the interval this person was/is a professor in our school)
  - the values of some of the attributes:
    - probably not the name
    - definitely the office hours

## Contents

- Data models for data-intensive Web sites
- Temporal databases
- Time in data-intensive Web sites
- Coordinates



## Contents

- Data models for data-intensive Web sites
- Temporal databases
- Time in data-intensive Web sites
- Coordinates

## The goal

- Reflection on requirements for modelling issues (on temporal aspects of Web sites)
- Essentially: how should we augment a logical model for Web sites to capture (the aspects of interest of) history of pages and their components and publish them in a suitable way



## Time dimensions for Web sites

- Valid time
- Transaction time

## Valid time for Web sites

- Essentially the same notion as for temporal databases; a difference:
  - in databases the interest is in representing sequences and querying them
  - here the challenge is in understanding what aspects of histories are of interest to visitors: it is a design issue



## Transaction time for Web sites

- Some people say that the Web is archival:
  - once a piece of information is published, it should not be retracted
- For sure, at least some changes should be documented:
  - if wrong exam dates are published, somebody would complain, and we should keep track
- Again, a design issue: what do we timestamp?

## Transaction time, additional issues

- It can refer to the future (in order to plan publication); in databases, instead, it is bounded by the current time
- How one keeps track of events? There are no transactions on the Web!
- So, transaction time on the Web need not coincide with the transaction time of the underlying database



## Modelling time in Web sites

- Incorporation of time in the data model, with:
  - distinction between temporal and static page schemes
  - distinction between temporal and static attributes (at the needed level of nesting)
  - suitable (and varying) time granularity

## Organization of temporal information

- Various forms:
  - snapshots:
    - a page for a course in a specific year
  - histories:
    - the list of instructors for a course over the years
  - combined:
    - the list and snapshots
  - a list of changes



## Coherence of information

- If various page schemes are temporal, then links should be coordinated, but this has also to be a design choice:
  - last year course should point to last year's instructor (unless the current instructor has responsibility, for example for delayed exams)

## Additional issues

- Version management
- Documenting the degree of currency of information
- Temporal aspects and content management systems



## Version management

- Many aspects can be versioned:
  - the values of data
  - the presentation
  - the hypertext structure
  - the database structure
- Could we be interested in seeing last years information with today's presentation (or may be the converse)?

## Documenting the degree of currency of information

- We often see "last changed on d-m-y"
- What does this mean?
  - the last time we changed something was d-m-y
    - if so, what was changed?
  - the last time we verified everything was d-m-y
- what is the appropriate granularity for tagging info?



## Temporal aspects and content management systems

- If the site content is managed in a structured way (and so the possibile changes are known), then we could have hints on what changes should be monitored and documented

## Contents

- Data models for data-intensive Web sites
- Temporal databases
- Time in data-intensive Web sites
- Coordinates



## Contents

- Data models for data-intensive Web sites
- Temporal databases
- Time in data-intensive Web sites
- Coordinates

## The same data and views for all?

- Adaptivity:
  - Personalization: content adapted to the user
    - upon system's decision
    - upon user's request
  - Customization: structure adapted to the user
    - according to the user's role
    - upon user's request
  - Context dependence

## Context

- Environment
  - User
  - Device
  - Network
  - Place
  - Time
  - Rate

## Coordinates

- A page could be the result of applying "parameters" (along coordinated) to a page template:
  - time
  - user
  - device
  - context
  - language
  - …



## Conclusions

- Preliminary thoughts on the needs for effective management of temporal issues
- Will experiment them in implementations, extending our tools