Intelligent Tagger for E-Learning

Ubayeed Syed
EECS 891 Final Project
January 21st, 2004

Committee:
Dr. Susan Gauch (chair)
Dr. Costas Tsatsoulis
Dr. Arvin Agah
Presentation Outline

- Introduction
- Objective
- Technical Background
- Design Considerations
- Implementation
- Evaluation
- Conclusion and Future Work
Introduction ..

§ Popularity of XML
- Data storage, Information Exchange
- Describe the data in a structured manner
- Separation of content and presentation
- Used in Educational Technology

§ Basic issues in Education Technology
- Produce and deliver quality of content
- Mix and match content from other resources
- Develop interchangeable content
- Basic unit is called ‘Learning Object’

§ Learning Objects
- Entities used, re-used or referenced during technology supported learning
Introduction

Intelligent Knowledge Management Environment (IKME)

- Reusability & Extensibility.
- Faster delivery of content.
- Using XML as data format for publishing.
Objective

System to Learn the users preferences from the users history
Help decrease the effort required to create new Learning Objects.
Technical Background ..

Reusable Learning Objects

- Problems with traditional content development techniques.
- Chunks of information.
- Reusability.
- Reduction of development time.
Native XML databases

- Databases designed specially to store XML documents.
- Defines a (logical) model for an XML document and stores and retrieves documents according to that model.
- Has an XML document as its fundamental unit of storage.
- Provides support features commonly found in common RDBMS.
Technical Background

- **XML-RPC**

  - specification and set of implementations that allow remote procedure calls.
  - Uses HTTP as transport and XML as encoding.
  - Allows integration of disparate systems running in different environments.
  - Serializing & Un-Serializing of data (XML format).
### Design Considerations

#### Process of Creating Learning Objects

- Initially Sample Template initialized with blank values.
- Sample template is loaded and displayed on the screen.
- Store the Learning Object into database.
- For each enumerated field determine the default values.
- Update Sample Template (XML file) with the default values of enumerated fields.
Design Considerations..

Why eXist?

Limitations of flat-file mechanism

- Data is isolated and separated
- file structure has to be defined
- Incompatible file formats
- Change in file structure may need a change in the code
- wastage of space.
Design Considerations ..

- Limitations of Cookies
  - Can identify only the last preference of the user but not the longer user activity history trend.
Design Considerations

- Database approach chosen because
  - No wastage of space
  - No redundancy in storing and defining data
  - Flexibility to choose the history size
  - Documents are stored in chronological order, hence recent user activity can be known.
Design Considerations

n Determining the default value from the users history
  n From the Schema, identify the enumerated fields.
  n For each enumerated field, Query the eXist database for recent $N$ Learning Objects.
  n Extract the enumerated field content from the results.
  n Choose the most common value as the default value.
  n Data structures used are one-dimensional arrays.
Implementation

- Perl Version 5.0 and later
- Apache Web Server
- eXist V0.9 and later
- Additional Perl Modules
  - RPC::XML
  - RPC::XML::Client
  - XML::Twig
  - XML::Sablotron
  - Data::Dumper
  - CGI
Implementation ..

- Major sections:
  - Parse_Schema.pl
  - Create.Temp.pl
  - Generate_Default_Values.pl
Implementation ..

- System Design

```
Sample Template  CGI Query String

CREATE_TEMP.PL

Temp.xml  Query String

GENERATE_DEFAULT_VALUES.PL

Updated Sample Template
```
Implementation ..

- System Inputs

  - Sample Template file: XML file, that has all the required fields for creating Learning Object.

  - CGI Query String: The array of attribute values obtained from the users Web Page as a cgi query string.
Example of Sample Template

```xml
<object>
  <metadata>
    <portionmark></portionmark>
    <environment></environment>
    ....
  </metadata>
  ....
  <tracking> ........
  </tracking> ........
</object>
```

Empty enumerated fields
Implementation

- System Output - *Updated Sample Template*

- The Sample template now has default values assigned for each enumerated field.
Example of Updated Sample Template

<object>
  <metadata>
    <portionmark> T </portionmark>
    <environment> NBC </environment>
  </metadata>
  ....
</object>

Updated enumerated fields
Implementation

- Intermediate Output - *Temp.xml*

  - Temporary XML file that has all the sub-fields of a highest level field in the XML Schema.
  - Its overwritten each time when user creates new Learning Object.
Implementation

Example of *Temp.xml*:

```xml
<metadata>
  <security>
    <classification> Confidential </classification>
    <portionmark/> T </portionmark>
  </security>
  .......... 
  <description>
    <object_category> preparation </object_category>
    <object_type/>
    <proponent/>
    <vital_record/>
  </description>
  .......... 
</metadata>
```
Evaluation

- Login Page - Restricts access to DB
Evaluation ..

Name the Learning Object
Evaluation ..

- Highest level fields of Learning Object
Evaluation ..

- Metadata and its sub-fields
Evaluation ..

- Metadata and its sub-fields
Evaluation ..

- Metadata – next logins, enumerated fields with default values
Evaluation

- Metadata – next logins, enumerated fields with default values
Benefits and Costs

Benefits

- Nothing has to be put on clients machine, user can thus work from any machine.
- No redundancy in storing and defining data.
- User has the flexibility to choose the total Learning Objects for his history.
Benefits and Costs continued

Costs

- User must login each time.
- Cost involved in accessing and updating the database.
  - Database has to be queried $M$ times, $M = \text{Size of history.}$
Future work

- Extend the functionality of login system to support multiple users with individual logins.
- Also, learn the preferences on optional enumerated fields.
- Investigate the size of history to get best default values.
- Investigate the methods of weighting recent values higher than older values.
Questions ?