XML Query Forms (XQForms): Declarative Specification of XML Query Interfaces

Michalis Petropoulos
Computer Science Dept.
University of California, San Diego / Enosys Markets, Inc.
mpetropo@cs.ucsd.edu

Vasilis Vassalos
Information Systems Dept.
New York University / Enosys Markets, Inc.
avasilis@enosysmarkets.com

Yannis Papakonstantinou
Computer Science Dept.
University of California, San Diego / Enosys Markets, Inc.
yannis@cs.ucsd.edu
Overview

Motives

• An increasing amount of sources export XML views of themselves
  – View = XML Schema + data
• XML Query will allow querying these views
• XQForms generates query forms and reports for querying these XML views
Basic Architecture

XML Schema → XQForms Editor → XQForm Annotation → XQForms

Annotations
Author

Data Server

XML queries → XML data
Basic Architecture

Example XML Schema
Basic Architecture

Query Control Annotation

```xml
<select>
   <label>Manufacturer</label>
   <values>
      <value default="true">
         <label>No preference</label>
         <data_value>any</data_value>
         <binding_path></binding_path>
         <condition>EQ</condition>
      </value>
      ...
      <value>
         <label>Turck</label>
         <binding_path>
            proximity_sensors/product/manufacturer
         </binding_path>
         <condition>EQ</condition>
      </value>
   </values>
   <multiple>false</multiple>
</select>
```
Basic Architecture

XML Schema → XQForms Editor → XQForm Annotation → XQForms

Annotations Author

Data Server

XML queries → XML data
Detailed Architecture

Steps in a Query Control lifecycle

XML Schema

XMAS Query

1 WHERE
2 proxSource proximity_sensors.product $P
3 AND $P manufacturer._ $M
4 AND $M = "Turck"

Annotation

1 <select>
2  <label>Manufacturer</label>
3  ...
4  <value>
5   <label>Turck</label>
6   <binding_path>
7    proximity_sensors/product/manufacturer
8   </binding_path>
9   <condition>EQ</condition>
10  </value>
11  ...
12 </select>

Form Elements
Detailed Architecture

Steps in an XQForm lifecycle

- Separation of content, query semantics and presentation
Detailed Architecture

Dependencies among Query Controls

- Ability to express CASE statements
- Different query controls contribute in querying
- The corresponding form elements are shown on the query form page
Reports

Annotation

<report>
  <main_result_unit>
    <binding_path>
      proximity_sensors/product
    </binding_path>
  </main_result_unit>
  <result>
    <label>Manufacturer</label>
    <binding_path>manufacturer</binding_path>
    <pattern>concat(., ' Manufacturer')</pattern>
    <summarize order="ASC">true</summarize>
  </result>
  <result>
    <label>Part Number</label>
    <binding_path>base_part_number</binding_path>
    <template>base_part_number.xsl</template>
    <summarize>false</summarize>
  </result>
  ...
</report>

- Specifies the data that appear on the report page
- Mechanism for easy customization of the report page
- The complexity of the XSL script for rendering the XML data is hidden
Reports

Architecture
• Engine applies system XSL scripts, called *metaXSL* scripts

• *metaXSL* scripts follow the structure of the XML query and construct the visual structure of the report page

• *metaXSL* scripts attach XSL templates to the simple and complex elements
Conclusions

Functionality Provided to Developers

- Declarative definition and generation of powerful query form and report pages for XML data modeled by XML Schemas
- Automated query construction and report formatting
- Separation of querying functionality and presentation
- High degree of customizability
Conclusions

Functionality Provided to End Users

- Ability to express arbitrary selection and projection conditions on the query form page
- Ability to express arbitrary sorting conditions
- Control on the number of elements that are returned from the data server
- Advanced navigation capabilities in the returned reports
- Ability to dynamically summarize and filter the query results on the reports pages
- On Line Demo: http://www10.enosysmarkets.com
Related Work

- **XForms**
  - Based on XML Schemas
  - Focus on end user interaction and XML instance creation

- **Strudel**
  - Graph model
  - Site graph
  - Functions on edges/links

- **Commercial systems**
  - Visual InterDev
  - Relational databases
Future Directions

- Expanded XML schema
  - Disjunction of repeatable elements
- Decision trees
  - Multi-step forms
  - More powerful dependencies
- Comparison reports
- Report customization
- Updates
Outline

• Overview
• Basic Architecture
• Detailed Architecture
  – The four steps of the XQForms’ lifecycle
• Reports
  – Automatic report construction
• Summary
  – Functionality provided
• Related work
• Future directions
Annotation Scheme

- Captures the general attributes of an XQForm
- Expresses the structure and the semantics of the query form page
- Specifies the syntax of an easily extensible collection of query controls
- Specifies which data elements will appear on the report page
Overview

XQForms Editor