

# XML-Enabled Database System

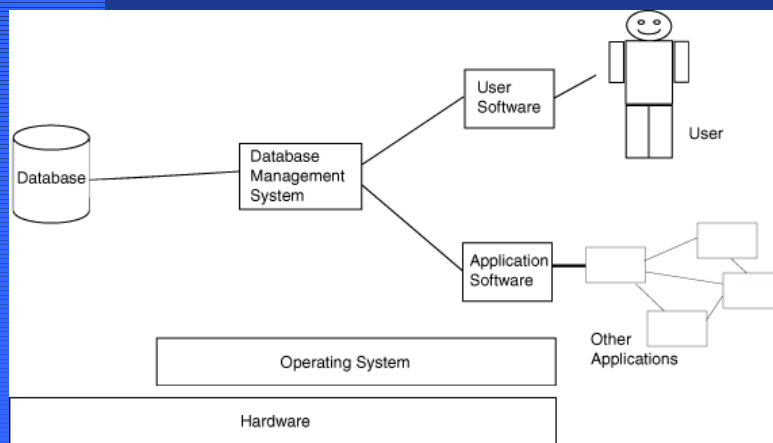
Mark Graves

*This presentation is Copyright 2001, 2002 by Mark Graves and contains material Copyright 2002 by Prentice Hall PTR. All rights reserved.*

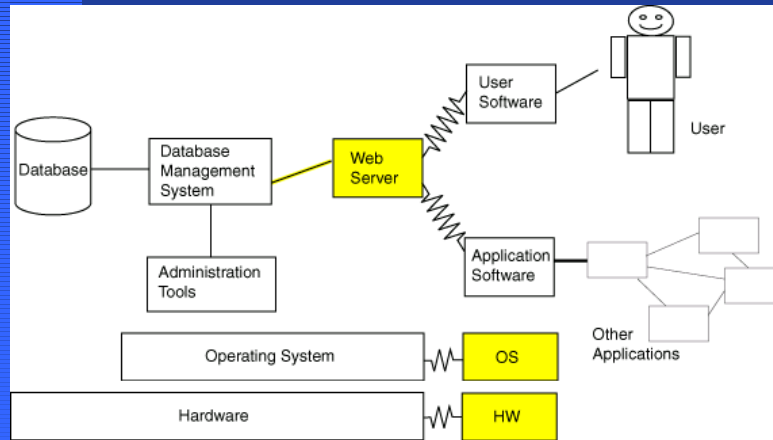
# Agenda

- Database System Overview
- Flat file database
  - Access via XSL
  - Browsing using Java visualizations
- Gene Notebook Example

# Database System



# Web-enabled Database System



## Web Server

- **Server which responds to HTTP (Hypertext Transport Protocol)**
- **Receives a URL request**
- **Returns files: static pages, CGI, servlet, JSP**
- **Examples: apache, tomcat**

[www.apache.org](http://www.apache.org)

## Web-enabled XML Database

- Flat-file database -- individual documents
- Relational database -- relational data rendered as XML
- XML database -- XML data stored in a DBMS

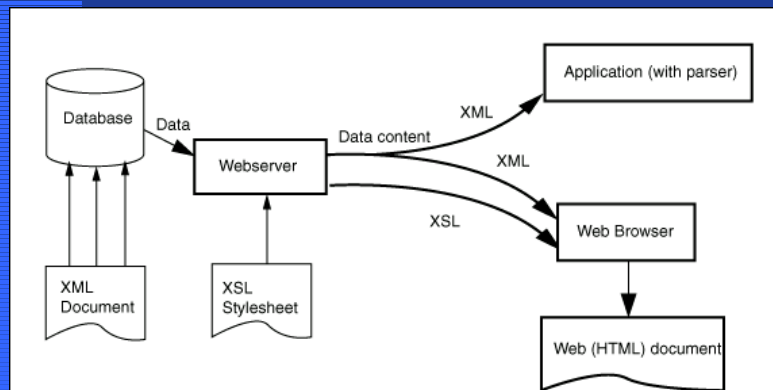
## Flat File XML Document

```
http://localhost/db/genes/14680.xml
<?xml version="1.0"?>
<gene id="14680">
  <name>BRCA1</name>
  <organism>Homo sapiens</organism>
  <chromosome_loc chr="17">17q21</chromosome_loc>
  <protein id="U37574"/>
  <DNA_sequence>atggattta</DNA_sequence>
  <db_xref gi="555931"/>
</gene>
```

## XML+XSL⇒HTML

- Use web server to organize XML documents
- Provide XSL style sheets to present data as HTML
- Use XSL-enabled web browser to navigate XML documents (e.g. Microsoft Internet Explorer 5+).

## Web-enabled XML Flat File DB



# Gene XSL Stylesheet

## XSLT

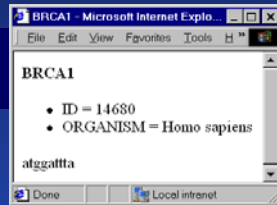
```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html"/>
  <xsl:template match="gene">
    <B><xsl:value-of select="name"/></B>
    <ul>
      <li>ID = <xsl:value-of select="@id"/></li>
      <li>ORGANISM =
        <xsl:value-of
          select="organism"/></li>
    </ul>
    <xsl:value-of select="DNA_sequence"/>
  </xsl:template>
</xsl:stylesheet>
```

## XML

```
<?xml version="1.0"?>
<?xml:stylesheet type="text/xsl" href="gene.xsl"?>
<genes>
  <gene id="14680">
    <name>BRCA1</name>
    <organism>Homo sapiens</organism>
    <chr_loc chr="17">17q21</chr_loc>
    <protein id="U37574"/>
    <DNA_sequence>atggatta</DNA_sequence>
    <db_xref gi="555931"/>
  </gene>
</genes>
```

## HTML

```
<B>BRCA1</B>
<ul>
  <li>ID = 14680</li>
  <li>ORGANISM = Homo sapiens</li>
</ul>
atggatta
```



# Generic Stylesheet

## XSLT

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html"/>
  <xsl:template match="/*[1]">
    <h2><xsl:value-of select="name()"/></h2>
    <TABLE>
      <xsl:for-each select="*">
        <TR>
          <TD><b><xsl:value-of select="name()"/></b>:
            <b></b>
          <TD><xsl:value-of select="."/></TD>
        </TR>
      </xsl:for-each>
    </TABLE>
  </xsl:template>
</xsl:stylesheet>
```

## XML

```
<?xml version="1.0"?>
<?xml:stylesheet type="text/xsl"
  href="record.xsl"?>
<gene id="14680">
  <name>BRCA1</name>
  <organism>Homo sapiens</organism>
  <chr_loc chr="17">17q21</chr_loc>
  <protein id="U37574"/>
  <DNA_sequence>atggatta</DNA_sequence>
  <db_xref gi="555931"/>
</gene>
```

## HTML

```
<h2>BRCA1</h2>
<TABLE>
  <TR><TD>name:</TD><TD>BRCA1</TD></TR>
  <TR><TD>organism:</TD><TD>Homo sapiens
    </TD></TR>
  ...
</TABLE>
```



## XML+Java⇒GUI

- **Use SAX parser framework**
  - start document
  - start element: name, AttributeList
  - character: char[], start, length
  - end element: name
  - end document
- **Use Java (Swing) components**
- **Create components as document is parsed**
- **May need intermediate data structure**

## Simple API XML (SAX)

- **Event-driven parsing --- Method is called for each parsing event.**
- **Events:**
  - start document
  - start element: name, AttributeList
  - character: char[], start, length
  - end element: name
  - end document

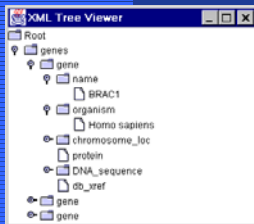
# SAX Trace

## XML

```
<?xml version="1.0"?>
<genes>
  <gene id="14680">
    <name>
      BRCA1
    </name>
    <organism>
      Homo sapiens
    </organism>
    <chr_loc chr="17">
      17q21
    </chr_loc>
    <protein id="U37574"/>
    <DNA_sequence>
      atggattta
    </DNA_sequence>
    <db_xref gi="555931"/>
  </gene>
</genes>
```

```
document start
start: genes
start: gene {id="14680"}
start: name
  chars: BRCA1
end: name
start: organism
  chars: Homo sapiens
end: organism
start: chr_loc {chr="17"}
  chars: 17q21
end: chr_loc
start: protein {id="U37574"}
end: protein
start: DNA_sequence
  chars: atggattta
end: DNA_sequence
start: db_xref {gi="555931"}
end: db_xref
end: gene
end: genes
document end
```

# Tree Viewer Application



```
public void startElement(String tag, AttributeList attrList) {
    MutableTreeNode currentParent = getCurrentParent();
    MutableTreeNode node = new DefaultMutableTreeNode(tag);
    treeModel.insertNodeInto(node, currentParent,
        currentParent.getChildCount());
    getParentStack().push(node);
}

public void endElement(String tag) {
    MutableTreeNode currentParent = getCurrentParent();
    getParentStack().pop();
}

public void characters(char[] chars, int start, int length) {
    String string = new String(chars, start, length);
    string = string.trim();
    if (string.length() == 0) return; //skip whitespace
    MutableTreeNode currentParent = getCurrentParent();
    MutableTreeNode node = new DefaultMutableTreeNode(string);
    treeModel.insertNodeInto(node, currentParent,
        currentParent.getChildCount());
}
```



## Graphical Visualization

- **Genome Sequence Viewer**
- **Pathway Visualization**
- **Micro-array Expression Results**

## Gene Notebook Example

How does this fit together?

## Agenda

- **System Requirements**
- **XSL/HTML Implementation**
- **Details**

## Gene Notebook

- **Web-based database to organize information about genes**
- **Should be able to add and update information on genes, gene annotations, and gene relationships**

## Data Types

- **Gene**
  - Descriptive Information
  - Sequence
  - Sequence Annotations
  - Literature References
- **Pairwise Relationships**
  - Similarity
  - Interaction
- **Relationship Collections**
- **Experimental Data**

## Operations

- **Create a gene**
- **Associate information with a gene**
- **Create pairwise relationships and collections of genes**
- **Associate information with those**
- **Search information in database to find its associated gene(s)**

# Requirements

- Web-based
- Persistent
- Can be updated

# User Interface

The screenshot shows a web browser window with the address bar displaying `http://127.0.0.1:8080/servlets/com.weave.smlib.demo.SMLServlet?doc=6&includeflagid=1&head=1&size=127`. The main content area is titled "Gene Notebook" and displays information for "BRCA1 Gene 14680".

**Genes**  
BRCA1 Gene 14680

**Properties**

- organism: Homo sapiens ([Change value](#), [Delete Property](#))
- chromosome\_loc: 17q21 ([Change value](#), [Delete Property](#))

**Annotations**

- DNA\_sequence: atggattta ([Change value](#), [Delete Annotation](#))
- protein\_sequence: PRKVFQENEMDVNGG ([Change value](#), [Delete Annotation](#))

**Commands**

[Change name of gene](#)

Add property type: [name] value:

Add annotation type:  value:

**Additional Information**

[View Microarray Data](#)

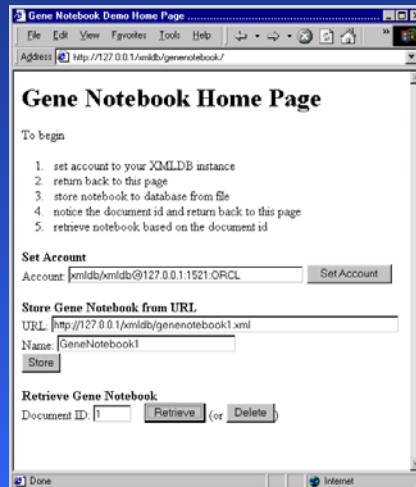
## Storage Locations

- **Gene**
  - Descriptive Information (XML DB) ➤
  - Sequence (FASTA) ➤
  - Sequence Annotations (XML DB)
  - Literature References (XML files) ➤
- **Pairwise Relationships (XML Link)**
- **Relationship Collections (XML DB)**
- **Experimental Data (RDB) ➤**

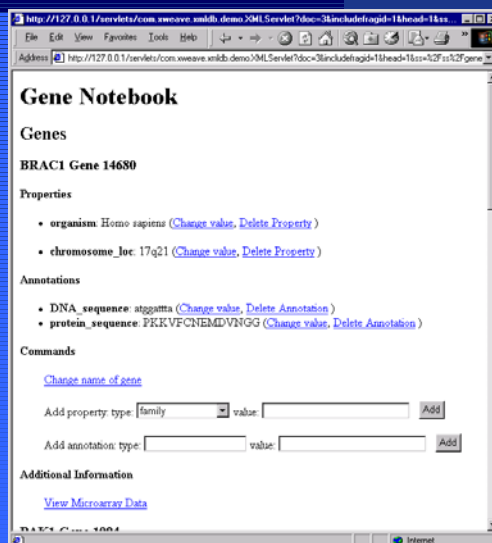
## Gene Notebook System

- **HTML Home Page to organize multiple Gene Notebooks**
- **XML Data Server (URL Java Servlet)**
- **XSL Stylesheet (11 templates; 167 lines)**
  - Default XML Templates (node, CData, etc)
  - Main Template (GeneNotebook)
  - Domain-specific Templates (genes, gene, interactions, relationships)

# Home Page



# Gene Notebook



# Gene Notebook



## XML

```
<?xml version="1.0"?>
<GeneNotebook>
<genes>
<gene id="14680">
  <name>BRCA1</name>
  <organism>Homo sapiens</organism>
  <annotations>
  <DNA_sequence>atggattta</DNA_sequence>
  <chr_loc chr="17">17q21</chr_loc>
  <protein_sequence>
    PKKVFCNEMDVNGG</protein_sequence>
  <db_xref gi="555931"/>
  </annotations>
</gene>
<gene id="1094">
  ...
</GeneNotebook>
```



## XSL

```
<?xml version="1.0"?>
<!-- Stylesheet for viewing GeneNotebook document -->
<xsl:stylesheet xmlns:xsl="...">
<xsl:template match="/">
  <xsl:apply-templates />
</xsl:template>
<xsl:template match="gene">
<h3><xsl:value-of select="name"/> Gene <xsl:value-of
  select="@id"/></h3>
<h4>Properties</h4>
<xsl:apply-templates select="*">
  <!-- apply templates for sub-elements of gene -->
  <xsl:template match="*">
    <ul>
  <li><b><xsl:node-name/></b>: <xsl:value-of
    select="."/>
  ...
</xsl:template>
```



# Aspects of Gene XSL Template

- Access Gene Properties
- Browse Microarray Data in a RDBMS by Building HREFs in XSL
- Add Property to a Gene
  - Edit XML Fragments
  - Build HTML Forms

# Accessing Gene Properties

```
<h4>Properties</h4>
<xsl:apply-templates select="">
  <!-- apply templates for sub-elements of gene -->
  <xsl:template match="">
    <ul>
      <li><b><xsl:node-name/></b>: <xsl:value-of select="" />
      (<xsl:element name="a"><xsl:attribute
name="href">com.xweave.xmldb.demo.XMLServlet?cmd=store&frag=<xsl:value-of
select="@FRAGID"/>&txttag=<xsl:node-name/></xsl:attribute>Change
value</xsl:element>,
      <xsl:element name="a"><xsl:attribute
name="href">com.xweave.xmldb.demo.XMLServlet?cmd=delete&frag=<xsl:value-of
select="@FRAGID"/>&txttag=</xsl:attribute>Delete Property</xsl:element>
      )</li>
    </ul>
  </xsl:template>
  <xsl:template match="name|annotations">
    <!-- already grapped name, annotations so ignore it this time -->
  </xsl:template>
</xsl:apply-templates>
```

## Properties

- organism: Homo sapiens ([Change value](#), [Delete Property](#))

# Browsing Relational Data

The screenshot shows a web browser window with two tabs. The active tab displays the 'PAK1 Gene 1094' page, which includes a 'Properties' section with 'organism: Yeast' and 'DNA\_sequence: ggatttta'. Below this is a 'Commands' section with a 'Change name of gene' link and a form to add properties or annotations. A second tab is open, displaying a table titled 'EXPER\_RESULT\_V LISTING' with columns for ID, GENE\_ID, NAME, EXPERIMENT, VARIANT, and VALUE. A line connects the 'View Microarray Data' link in the first tab to the table in the second tab.

**PAK1 Gene 1094**

**Properties**

- organism: Yeast ([Change value](#), [Delete Property](#))

**Annotations**

- DNA\_sequence: ggatttta ([Change value](#), [Delete Annotation](#))

**Commands**

[Change name of gene](#)

Add property: type:  value:

Add annotation: type:  value:

**Additional Information**

[View Microarray Data](#)

**EXPER\_RESULT\_V LISTING**

ID	GENE_ID	NAME	EXPERIMENT	VARIANT	VALUE
3299	1094	PAK1	102	240	0.21
3300	1094	PAK1	102	270	0.18
3301	1094	PAK1	102	300	-0.15
3302	1094	PAK1	102	330	0.19
3303	1094	PAK1	102	360	0.04
3304	1094	PAK1	102	390	0.07
3305	1094	PAK1	103	10	-0.23
3306	1094	PAK1	103	30	-0.25
3307	1094	PAK1	103	50	-0.32
3308	1094	PAK1	103	70	-0.45
3309	1094	PAK1	103	90	-0.67
3310	1094	PAK1	103	110	-0.58
3311	1094	PAK1	103	130	-0.74
3312	1094	PAK1	103	150	-0.64



# Building HREFs in XSL

```
<h4>Additional Information</h4>
<div style="margin-left:2em">
<!-- Link to Microarray Data -->
<xsl:element name="a">
  <xsl:attribute name="href">
    com.xweave.xmldb.demo.XMLServlet?
    cmd=rdb&amp;
    name=<xsl:value-of select="name"/>&amp;
    tablename=madb.exper_result_v&amp;
    stylesheet=/ss/generic1.xsl
  </xsl:attribute>
  View Microarray Data
</xsl:element>
</div>
```

Additional Information

[View Microarray Data](#)

```
<a href="com.xweave.xmldb.demo.XMLServlet?
cmd=rdb&name=PAK1&tablename=madb.exper_result_v&stylesheet=/ss/generic1.xsl">
```

# Adding Gene Property

http://127.0.0.1/servlets/com.xweave.xmldb.demo.XMLServlet?doc=2&includefragid=1&head=1&...  
Address: http://127.0.0.1/servlets/com.xweave.xmldb.demo.XMLServlet?doc=2&includefragid=1&head=1&ss=%2Fss%2Fgenenotebook.xsl&cmd=Retrieve

**ANGPT1 Gene 531**

**Properties**

- organism: Human ([Change value](#), [Delete Property](#))

**Annotations**

- protein\_sequence: MPEPKKVFNCNMDVNGG ([Change value](#), [Delete Annotation](#))

**Commands**

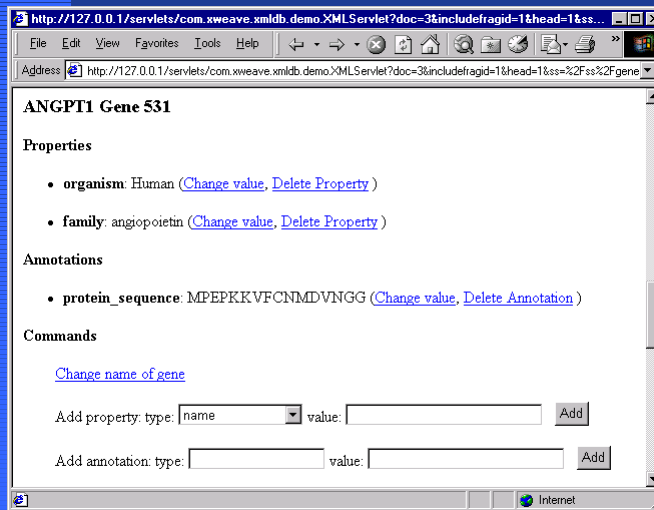
[Change name of gene](#)

Add property: type:  value:

Add annotation: type:  value:

```
http://127.0.0.1/servlets/com.xweave.xmldb.demo.XMLServlet?
texttag=name&text=angiotensin&frag=3.15&cmd=Append
```

# Protein Family Property Added



# XML Data with Fragment Ids

```
<gene FRAGID="3.15" id="531">
  <name FRAGID="3.16">
    ANGPT1
  </name>
  <organism FRAGID="3.17">
    Human
  </organism>
  <annotations FRAGID="3.18">
    <protein_sequence FRAGID="3.19">
      MPEPKKVFCNMDVNGG
    </protein_sequence>
  </annotations>
</gene>
```

```
<gene FRAGID="3.15" id="531">
  <name FRAGID="3.16">
    ANGPT1
  </name>
  <organism FRAGID="3.17">
    Human
  </organism>
  <annotations FRAGID="3.18">
    <protein_sequence FRAGID="3.19">
      MPEPKKVFCNMDVNGG
    </protein_sequence>
  </annotations>
  <family FRAGID="3.30">
    angiotensinogen converting enzyme 1
  </family>
</gene>
```

# Building HTML FORMs in XSL

Add property: type: family value: Add

```
<form
  action="com.xweave.xmldb.demo.XMLS
  ervlet" method="GET">
  Add property:
  type:
  <select name="texttag">
    <xsl:for-each
      select="/GeneNotebook/genes/gene/*">
      <option><xsl:node-name/></option>
    </xsl:for-each>
  </select>
  value:
  <input type="input" name="text"
    size="30"/>
    <xsl:element name="input">
      <xsl:attribute name="type">
        hidden</xsl:attribute>
      <xsl:attribute name="name">
        frag</xsl:attribute>
      <xsl:attribute name="value"><xsl:value-of
        select="@FRAGID"/></xsl:attribute>
    </xsl:element>
    <input type="hidden" name="cmd"
      value="Append"/>
    <input type="submit" name="ignore"
      value="Add"/>
  </form>
```

http://127.0.0.1/servlets/com.xweave.xmldb.demo.XMLServlet?  
texttag=name&text=angiopoietin&frag=3.15&cmd=Append&ignore=Add