Chapter 17: The iDiary Database: A Case Study in Database Organization

Fluency with Information Technology
Third Edition

by
Lawrence Snyder

Thinking About a Personal Database

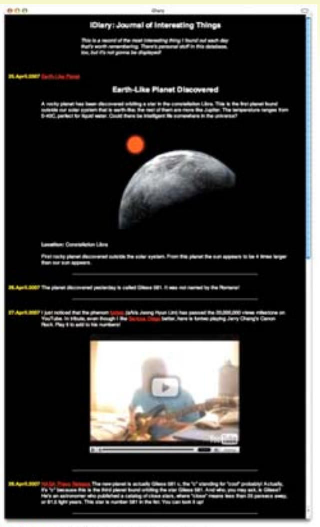
- Regular Versus Irregular Data
  - The iDiary will be an irregular data collection
    - Record things we find interesting in our daily lives
      - Text, photos, URLs, animations, poems, videos, etc.
  - Use XML to specify metadata
    - The database will be an XML tree
    - Use the Identity, Affinity, and Collection rules
  - Organize the database by date
    - The iDiary added to each day

Thinking About a Personal Database (cont'd)

- Physical Versus Logical
  - The XML tree will be our physical database
  - The logical database is our view of the iDiary
  - Use XSL to pick out data we want to display
    - XSL description converts the data to HTML
    - XSL description act like a query with a relational database

Thinking About a Personal Database (cont'd)

- The iDiary
  - Build the iDiary database and its stylesheet display together and incrementally
  - Step-by-step approach
    - Easier to debug
    - Mirrors how databases are enhanced over time



iDiary: Journal of Interesting Things

[illegible]

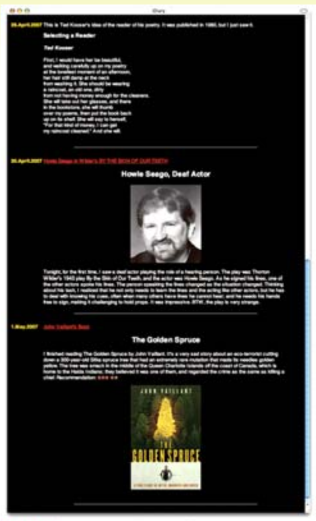
Earth-Like Planet Discovered

[illegible]

Location: Constellation Libra

[illegible]

Figure 17.1 Part 1: An example of the planned iDiary.



[illegible]

Selecting a Reader

Ted Kooser

[illegible]

Howie Seago, Deaf Actor

[illegible]

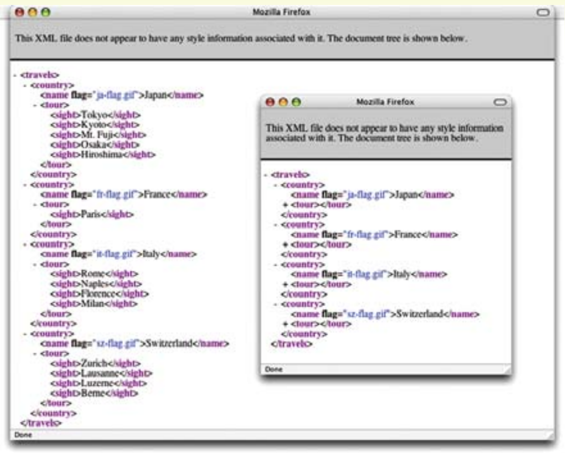
The Golden Spruce

[illegible]

Figure 17.1 Part 2: An example of the planned iDiary.

A Preliminary Exercise

- Travels Database
  - The XML Definition
    - Entries in the database will be a list of countries
    - Each will have a name and a tour that contains a list of sights, along with that country's flag
  - XML database file named `Travels.xml`
- Direct Check of XML
  - Can have a browser display our XML tree



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
- <travels>
  - <country>
      <name flag="ja-flag.gif">Japan</name>
    - <tour>
        <sight>Tokyo</sight>
        <sight>Kyoto</sight>
        <sight>Mt. Fuji</sight>
        <sight>Osaka</sight>
        <sight>Hiroshima</sight>
      </tour>
    </country>
  - <country>
      <name flag="fr-flag.gif">France</name>
    - <tour>
        <sight>Paris</sight>
      </tour>
    </country>
  - <country>
      <name flag="it-flag.gif">Italy</name>
    - <tour>
        <sight>Rome</sight>
        <sight>Naples</sight>
        <sight>Florence</sight>
        <sight>Milan</sight>
      </tour>
    </country>
  - <country>
      <name flag="sz-flag.gif">Switzerland</name>
    - <tour>
        <sight>Zurich</sight>
        <sight>Lausanne</sight>
        <sight>Luzerne</sight>
        <sight>Berne</sight>
      </tour>
    </country>
  </travels>
```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
- <travels>
  - <country>
      <name flag="ja-flag.gif">Japan</name>
    + <tour></tour>
    </country>
  - <country>
      <name flag="fr-flag.gif">France</name>
    + <tour></tour>
    </country>
  - <country>
      <name flag="it-flag.gif">Italy</name>
    + <tour></tour>
    </country>
  - <country>
      <name flag="sz-flag.gif">Switzerland</name>
    + <tour></tour>
    </country>
  </travels>
```

Figure 17.2 The display of the `Travels.xml` file using the Firefox browser. The inset shows the result of clicking on the minus signs (–) in front of the <tour> tags.

Displaying the Travels with XSL

- Connecting XML with Style
  - Style information tells the browser how to display a markup language like XML
  - Style information comes from a companion file with the file extension `.xsl`
  - Put in the XML file a line which tells the browser where to find the style information

```
<?xml-stylesheet type="text/xsl" href="TravelSS.xsl"?>
```

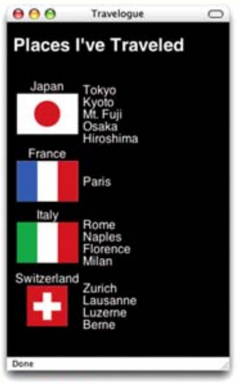


Figure 17.3 The display of the `Travels.xml` file using the `TravelsSS.xsl` style information.

Displaying the Travels with XSL (cont'd)

- The Idea of XSL
  - The `.xsl` file contains a series of rules (**templates**) on how to format (using HTML) the information enclosed in XML tags in the database

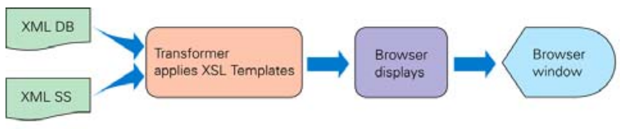
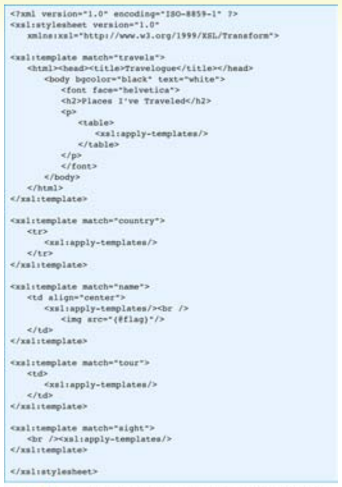


Figure 17.4 Schematic diagram showing how the XML database tree and the XSL style information are merged to produce HTML; the final HTML result is displayed by the browser.

Displaying the Travels with XSL (cont'd)

- XSL Templates
  - XSL is really just XML with one template for each XML tag with HTML for how to display the XML tag

```
<xsl:template match="XML tag name">
        …
</xsl:template>
```

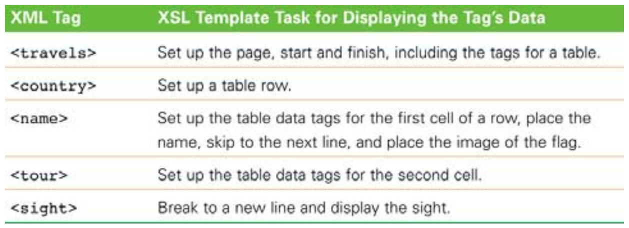


```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet version="1.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="travels">
   <html><head><title>Travelogue</title></head>
      <body bgcolor="black" text="white">
         <font face="helvetica">
         <h2>Places I've Traveled</h2>
         <p>
            <table>
               <xsl:apply-templates/>
            </table>
         </p>
         </font>
      </body>
   </html>
</xsl:template>

<xsl:template match="country">
   <tr>
      <xsl:apply-templates/>
   </tr>
</xsl:template>

<xsl:template match="name">
   <td align="center">
      <xsl:apply-templates/><br />
         <img src="{@flag}"/>
   </td>
</xsl:template>

<xsl:template match="tour">
   <td>
      <xsl:apply-templates/>
   </td>
</xsl:template>

<xsl:template match="sight">
   <br /><xsl:apply-templates/>
</xsl:template>

</xsl:stylesheet>
```

Figure 17.5 The contents of the TravelsSS.xsl file that produced Figure 17.3.

Displaying the Travels with XSL (cont'd)

- Creating the Travelogue Display
  - Each XML tag has a stylistic role to play in the overall creation of the Web page



XML Tag	XSL Template Task for Displaying the Tag's Data
`<travels>`	Set up the page, start and finish, including the tags for a table.
`<country>`	Set up a table row.
`<name>`	Set up the table data tags for the first cell of a row, place the name, skip to the next line, and place the image of the flag.
`<tour>`	Set up the table data tags for the second cell.
`<sight>`	Break to a new line and display the sight.

Displaying the Travels with XSL (cont'd)

- The Apply Operation

  `<xsl:apply-templates/>`
  - This tag means "now process whatever is inside this XML tag"

```
<xsl:template match="tour">
   <td>
      <xsl:apply-templates/>
   </td>
</xsl:template>
```

Displaying the Travels with XSL (cont'd)

- Tag Attributes
  - Use curly braces to place information in matched quotes
  - `@flag` refers to the value of the `flag` attribute of the `<name>` tag

```
<xsl:template match="name">
   <td align="center">
      <xsl:apply-templates/> <br />
         <img src="{@flag}"/>
   </td>
</xsl:template>
```

eg. `<img src="{@flag}"/>` becomes `<img src="fr-flag.gif"/>`

Displaying the Travels with XSL (cont'd)

- Summary of XSL
  - Browser opens the .xml file, finds a style specification, opens the .xsl file, and begins to process the XML tree
  - The process:
    - match a template
    - do what needs to be done before processing the enclosed information
    - process the enclosed information
    - do what needs to be done after processing the enclosed information
    - consider that tag processed

The iDiary Database

- Incremental approach
  1. Getting started
  2. Creating the first entry (April 26)
  3. Thinking about the nature of things
  4. Developing tags and templates
  5. Critiquing and evaluating the results

Getting Started

- Creating the XML Database (iDiary.xml)
  - Decide on root Collection tag (`<idiary>`) and Affinity tags to enclose daily info (`<entry>`)
- Creating the XSL Stylesheet (iDiarySS.sxl)
  - Contains the setup for the Web page (title, heading, italicized comment at the start of the page), table containing all the entries, table for each entry

Creating the First Entry (April 26)

- Date Tagging
  - Let date be atomic, pick a date format, and surround with `<date>` tags
- Revising an `<entry>`
  - `<mit>` tagging most interesting thing that day
  - Add XSL templates in `iDiary.xsl` for each XML tag
- Critiquing the Design
  - Vertically align the date
  - Modify the color and font

Thinking About the Nature of Things

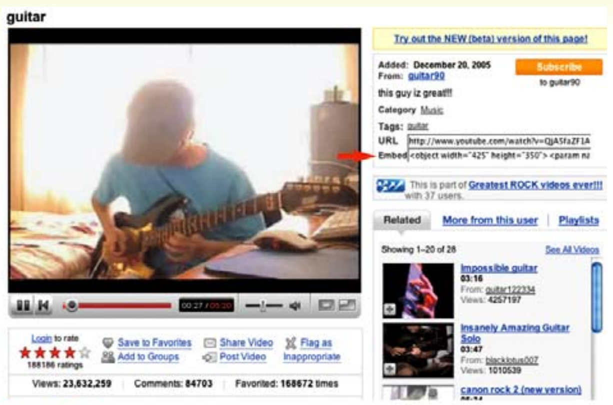
- Recognizing the Need for Specific Tags
  - Different kinds of data need different tags
- Choosing Specific Tags
  - Previously an Identification tag, `<mit>` tag becomes more of an Affinity tag
  - Still a sister to the `<date>` tag, still identifies the most interesting thing, style role continues

Developing Tags and Templates

- The Fact Tag
  - Enclosed by the `<mit>` tag
- The Title Tag
  - Announces the most interesting thing entry
- The Link Tag
  - Specifies a Web link `href="{@url}"`
- The Picture Tag
  - Stand-alone tag; no need for `<xsl:apply-templates/>`
  - All information expressed as tag attributes

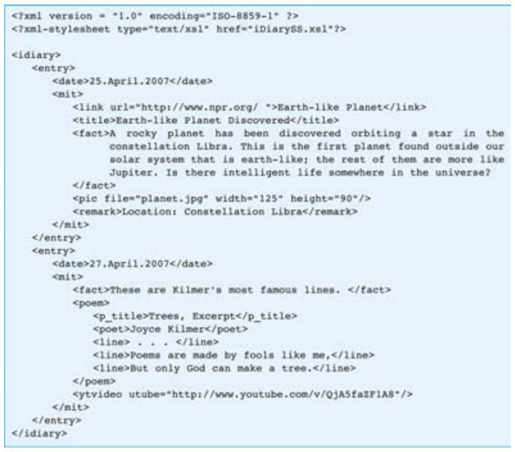
Developing Tags and Templates (cont'd)

- The Remark Tag
  - Captions and labels
- The Poetry Tags
  - Title, author, and lines of poetry – assign tags to each
- The Video Tag
  - Display a player as an embedded object (YouTube)
  - Stand-alone tag like the picture tag
- A Check of the Design



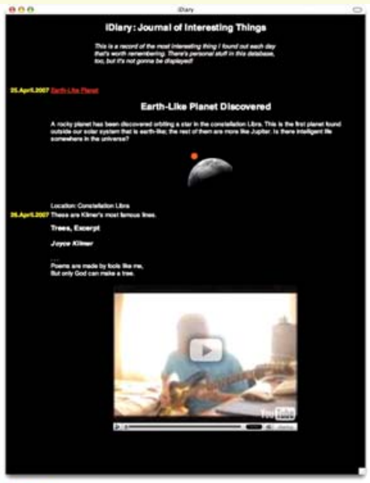
Added: December 20, 2005
From: guitar90
this guy iz great!!!
Category Music
Tags: guitar
URL http://www.youtube.com/watch?v=QjASfaZF1A
Embed <object width="425" height="350"> <param na
This is part of Greatest ROCK videos ever!!! with 37 users.
Showing 1-20 of 28
Impossible guitar 03:16 From: guitar122334 Views: 4257197
Insanely Amazing Guitar Solo 03:47 From: blacklotus007 Views: 1010539
canon rock 2 (new version)
188186 ratings
Views: 23,632,259 Comments: 84703 Favorited: 168672 times

Figure 17.6 YouTube display marking the embedding information.



```
<?xml version = "1.0" encoding="ISO-8859-1" ?>
<?xml-stylesheet type="text/xsl" href="iDiarySS.xsl"?>

<idiary>
   <entry>
      <date>25.April.2007</date>
      <mit>
         <link url="http://www.npr.org/ ">Earth-like Planet</link>
         <title>Earth-like Planet Discovered</title>
         <fact>A rocky planet has been discovered orbiting a star in the
               constellation Libra. This is the first planet found outside our
               solar system that is earth-like; the rest of them are more like
               Jupiter. Is there intelligent life somewhere in the universe?
         </fact>
         <pic file="planet.jpg" width="125" height="90"/>
         <remark>Location: Constellation Libra</remark>
      </mit>
   </entry>
   <entry>
      <date>27.April.2007</date>
      <mit>
         <fact>These are Kilmer's most famous lines. </fact>
         <poem>
            <p_title>Trees, Excerpt</p_title>
            <poet>Joyce Kilmer</poet>
            <line> . . . </line>
            <line>Poems are made by fools like me,</line>
            <line>But only God can make a tree.</line>
         </poem>
         <ytvideo utube="http://www.youtube.com/v/QjA5faZF1A8"/>
      </mit>
   </entry>
</idiary>
```

Figure 17.7 Sample database entries for checking the iDiary tags and templates.



iDiary

iDiary: Journal of Interesting Things

This is a record of the most interesting thing I found out each day that's worth remembering. There's personal stuff in this database, too, but it's not gonna be displayed!

25.April.2007 Earth-Like Planet

Earth-Like Planet Discovered

A rocky planet has been discovered orbiting a star in the constellation Libra. This is the first planet found outside our solar system that is earth-like; the rest of them are more like Jupiter. Is there intelligent life somewhere in the universe?

Location: Constellation Libra

26.April.2007 These are Kilmer's most famous lines.

Trees, Excerpt

Joyce Kilmer

. . .
Poems are made by fools like me,
But only God can make a tree.

Figure 17.8 Firefox display of the sample diary in Figure 17.7.

Critiquing and Evaluating the Results

- Form of Entries
  - Add breaks and horizontal line to separate entries
  - Compact entries by limiting the width of the table data
- Remarks On <remark>
  - New Label tag to bold the information it encloses

Using the iDiary Daily

- Archiving Photos
  - Putting the path to images in the XML file (versus the XSL file) allows us to make references to images stored in different places, including images stored elsewhere on the Internet

Using the iDiary Daily (cont'd)

- Hiding Information
  - Enclose personal information you do not want displayed in `<personal>` tags
  - Do not include `<xsl:apply-templates/>` tag in the XSL template for `<personal>`
  - Information inside the tags will be skipped
  - Note: Not enclosing information in a tag or tagging it but not providing a template for the tag will result in the content being displayed

Using the iDiary Daily (cont'd)

- Entering Data into the Database
  - Create a "template" for a new entry in the XML database file
  - Just copy/paste this "template" and edit it