

# Web Services

## Unit -5

### Linking in xml

#### Two marks:

##### **1. Links in xml:**

\* Link is used to refer to a relationship between two pieces of information, to a connection between two physical resources (web pages) and used to user interface objects, such as a hot spot on a Web page, which let you jump to another page.

\* It expressing relationships between pieces of information in a content based tagging language, expressing relationships between resources in a synthesis-based tagging language, and creating hotspots and other user interface objects in a presentation-based language.

##### **2. Generated links:**

\* The generated links is easy to query a database and create web pages based on rows in table with links to pages based on rows in a related table.

\* All that is required is a file naming convention based on data in the tables.

\* Any unique field or combination of fields will do as a source of HTML filenames.

### **3. Internal links:**

\* These have two approaches to create a broken internal link:

1. The first is to generate the pages of our site from a central xml data store and content management system in a batch process. The batch process determines both the names of the files that it creates and the names of the URIs it insets into links as it.

2. The second is to deliver each page of our site by generating it on the fly from our XML data store or content management system as the user requests it. Every internal link on a page generated is effect, link to the process that generates our pages, and it contains query information that the application that generates pages what data to assemble to create the next page.

### **4. External links:**

\* With external sites, is to ensure that the URIs of external sources are never recorded directly in content tagged XML data.

\* To mark up your content so that names are located in their logical name space and add links to the external sites at synthesis time, or you mark up a relationship to your import table and resolve that into a relationship to a site of synthesis time.

### **5. Common names in xml link:**

\* The common names are the name with the object normally call it. If the information object corresponds closely with a real object, its common name will usually be related to the common name of the object.

\* A common name is seldom a stable identifier. It cannot always be guaranteed to be unique within a name space.

## **6. Abstract names in xml link:**

\* Abstract names are generally created for the purpose of being guaranteed unique, unambiguous, and stable.

\* For instance, government assign all citizens a number identifies the uniquely despite any number of people with the same common name and any number of name changes.

## **7. Formal names in xml link:**

\* Formal names are usually stable and unique.

\* The set of abstract names often overlaps the set of common names. Formal names require a formal model and some authority changed with the maintenance of the model and collection of formal names associated with it.

## **8. Local names in xml link:**

\* Local names are essentially common name in a different environment.

\* Information objects and the objects describe frequently have different names in different locations and among people of different professions.

## **9. Nicknames in xml link:**

\* Nick names are subject to all the variability of common names but, because they tend to be short, are often subject to greater ambiguity.

\* This includes short forms of common names and other nicknames that bear no relationship to the common name.

## **10. Ad Hoc names in xml link:**

\* Ad Hoc names are a normal part of communication, to vary name references to suit the particular context, to bring out a particular emphasis or simply to provide a pleasant variation of language.

## **11. Xpointer in xml link:**

\* The Xpointer are intended to be used in conjunction with URIs, Where a URI points to a whole resource, an Xpointer points to a location or a fragment within a resource.

\* Xpointer is not about the behavior of any browser or other application, it is about pointing at fragments of XML files, and that's all.

## **12. Xml linking methods:**

\* The linking model defined here provides ways to create links that go beyond each of these specific characteristics, thus providing features previously available in dedicated hypermedia systems.

\* The XLink working draft embodies a view of a substantially different kind of hypertext application than the phenomenally successful web browser.

\* Xlink is not a language, we design our own information structures and markup language to specifically support the kinds to relationships our content requires for internal development efforts.

## XML style

### **1. Stages of publishing process:**

\* The business of making information accessible to an audience has four stages:

1. Content
2. Synthesis
3. Formatting
4. Rendering

### **2. Content stage:**

\* In a system based, content is a long way from publication.

\* The good data model expressed in a database and xml-based markup languages, a well ordered collection of content can be assembled into a book.

\* It can also be made into an effective CD-ROM, or a web site.

### **3. Synthesis stage:**

\* Synthesis is a laborious manual process and there was a strong incentive to pick one synthesis of material and stick to it.

\* If we have sufficient structure in our content, we can synthesize with design script and a processing application.

\* This opens the possibility of synthesis on demand for many types of information, enabling us to synthesize material to meet the needs of each individual.

#### **4. Presentation stage:**

\* All of the information would have been recorded in a design document that would have been passed off to typesetters and the printing plant.

\* Formatting languages that can describe the formatting of a print page or a web page.

\* We can use a processing application to take data with synthesis markup and a presentation formatting script and apply formatting automatically.

#### **5. Rendering stage:**

\* The web browser reads pages expressed in its formatting language and communicates with the operating system to turn individual pixels on and off, thus rendering the page on our screens.

\* To rendering the HTML formatting language consistently into the very different rendering languages of the different platforms on which they operate.

#### **6. Structure the data in content stage:**

\* If we structure the data at content level, have to do several information before it can be rendered and read.

\* If you store the data in HTML won't have any transformations at all, because browsers will take care of rendering the data.

## **7. Pros and cons of server-side processes:**

- 1.** Doing the processing on the browser allows you to distribute the processing.
- 2.** Doing them on the browser involves publishing your content or synthesis data on the web.
- 3.** More tools and languages are available on the server side.
- 4.** More information is available on the server side.
- 5.** You can exercise more control over your own information on the paper side.
- 6.** Server-side processes can address a wider variety of clients.
- 7.** Server-side processes can address other media.