Information Management Resource Kit

Module on Management of Electronic Documents

UNIT 5. DATABASE MANAGEMENT SYSTEMS

LESSON 2. USING A DATABASE FOR DOCUMENT RETRIEVAL

NOTE

Please note that this PDF version does not have the interactive features offered through the IMARK courseware such as exercises with feedback, pop-ups, animations etc.

We recommend that you take the lesson using the interactive courseware environment, and use the PDF version for printing the lesson and to use as a reference after you have completed the course.

Objectives

At the end of this lesson, you will be able to:

- understand the **requirements** for **information delivery**, and
- comprehend the **role of databases** in information delivery.

Introduction

How will the users access electronic documents?

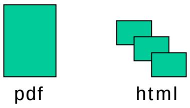
Staff in the Information Dissemination Division in the General Information and Public Affairs Department are considering the need for using a database to deliver their organization's information.

Focusing on the **delivery process**, they have to consider different aspects of their system.

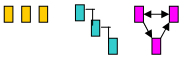
One important aspect, which is not directly related to databases, is that users should be allowed to access the documents **quickly and easily**.

Requirements for document delivery

We can break requirements for document delivery into four main areas:

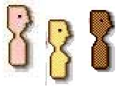


- requirement for **retrieval** of the document content;



- requirements for **browsing information**;

- **search requirements**; and



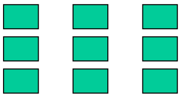
- **user related requirements**.

Requirements for document retrieval

Regarding **retrieval of document content**, users should be able to:

View information in the format it is supplied in

Plain text, HTML and XML formats, with open standard graphics, audio and video formats, are the best ways to deliver information so that everyone can view it.



Access information at the appropriate level of granularity

You need to deliver just the right amount of information that your user needs. For example, if some users are interested in only two or three steps of an entire procedure, each individual step should be made available as a self-contained unit of information.

Requirements for document retrieval

The main requirements for **browsing information** can be broken down into:

Navigating document collections

Browsing through sets of documents which are organized into **repositories or collections**.

Navigating by taxonomy

Browsing by **hierarchical classification** of documents. The simplest taxonomy is a fixed organization such as the folders you create on a file system for example in Microsoft Windows. More complicated (and useful) are **dynamic taxonomies** where you can overlay different hierarchical classifications on the same set of documents.

Navigation through hypertext

Hypertext provides a way to **link from an anchor point inside one document to another document** or target location inside the same document as the anchor or inside another document.

Requirements for document retrieval

Search requirements fall into the following main categories:

Full text search

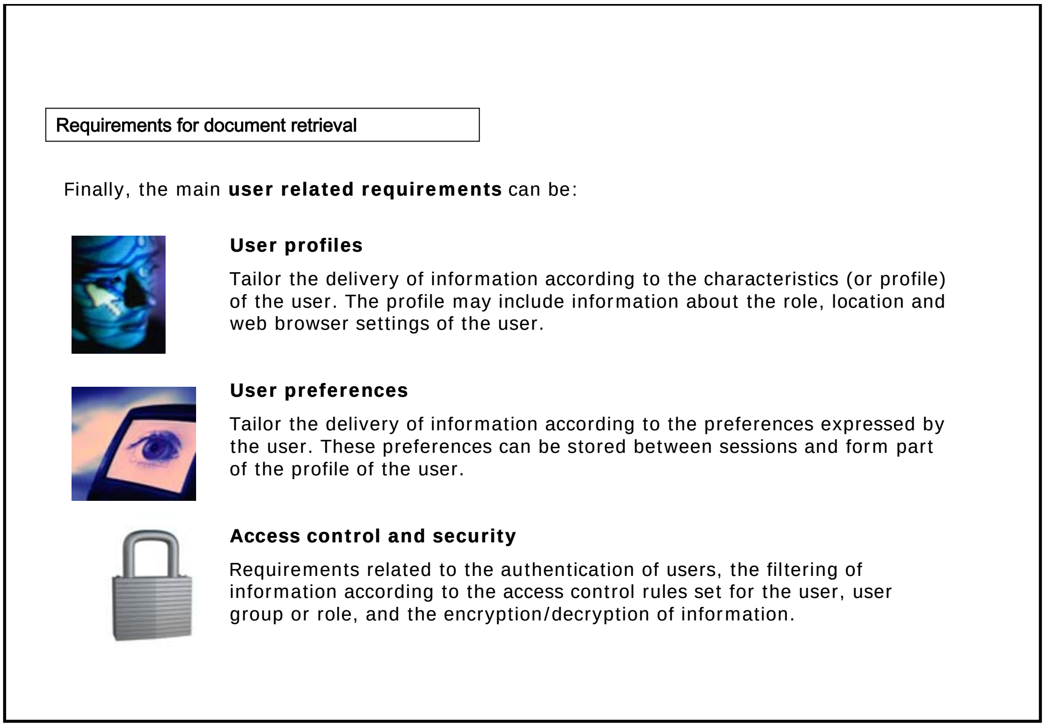
Searches on the text content of documents.

Metadata search

Searches using metadata items associated with document instances.

Structured text search

Searches which combine full text with the semantic constraints expressed in structured documents marked-up in XML (or to a limited extent, HTML).



Requirements for document retrieval

Finally, the main **user related requirements** can be:

User profiles

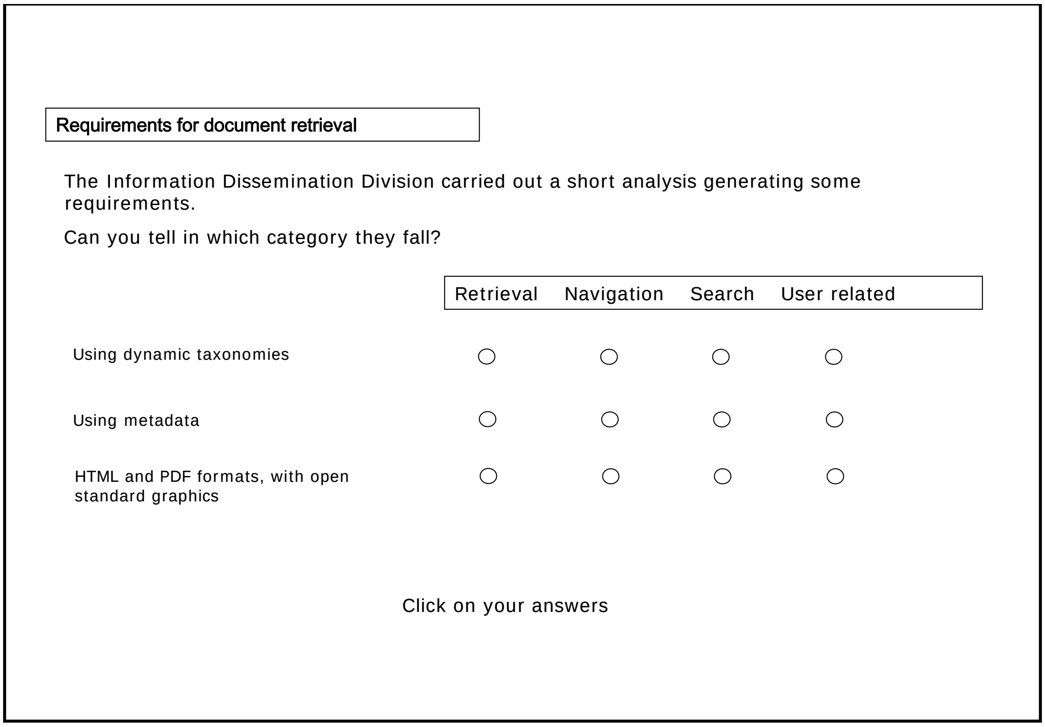
Tailor the delivery of information according to the characteristics (or profile) of the user. The profile may include information about the role, location and web browser settings of the user.

User preferences

Tailor the delivery of information according to the preferences expressed by the user. These preferences can be stored between sessions and form part of the profile of the user.

Access control and security

Requirements related to the authentication of users, the filtering of information according to the access control rules set for the user, user group or role, and the encryption/decryption of information.



Requirements for document retrieval

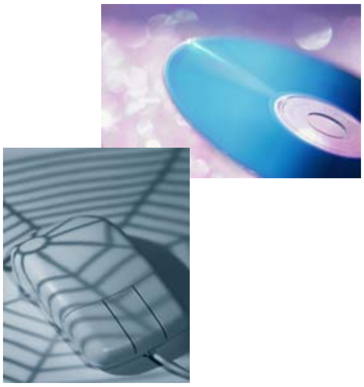
The Information Dissemination Division carried out a short analysis generating some requirements.

Can you tell in which category they fall?

	Retrieval	Navigation	Search	User related
Using dynamic taxonomies	○	○	○	○
Using metadata	○	○	○	○
HTML and PDF formats, with open standard graphics	○	○	○	○

Click on your answers

Using a database for delivery



Now, you can start to choose the delivery system, that is how information will be distributed.

Several specific options are available: users can access information from a **CD-Rom**, consult a **website**, use a **database**, a **portal**, etc...

Using a database for delivery

We should first consider the use of web technology as the main user agent for delivery, since…

When you choose the delivery system, remember the advantages of using web technology:

• most people **already have a web browser** on their desktop and so they don't need to install any special software to access your information;

• you **don't need to train users** to use a web browser interface – people already know the basic moves, and so the only training they are likely to need is in any special ways to navigate or search your information;

• you can make the **same information system work on a CD-Rom, the Internet or a local network**, which greatly reduces the amount of effort you need to put in to reach different groups of users.

Using a database for delivery

You should consider **delivering information on CD** when you know:

- your users have **no access to the Internet**,
- they have a **limited bandwidth connection** which might restrict the amount of information they can download, or
- they have **intermittent access** which might prevent them from seeing important information at the very moment when they most need it.

In this case, you have three choices in how to create the disk…

Using a database for delivery

Choices for CD creation

When CDs first appeared as a distribution media for electronic documents you really had only two choices in how to create the disk:

- Write a **collection of static documents** that could be browsed through the file system of the computer the disk was accessed on or through a web browser.
- Use a commercial product to compile an application that would run **as a database or indexed search engine directly from the CD** (which may or may not run an installer to install that application on the hard drive of the user's machine or network).

There is now a third way, in that many web applications can be bundled up (often using open source or other freely available software) so that **the entire application that would normally run on a server connected to the Internet, can be run from the CD**, including the web-server, application server and database. As an information provider this is quite a good option, because you don't need to create and maintain different versions of the information or application for the Web and CD.

Using a database for delivery

Retrieval	HTML and PDF formats, with open standard graphics
Navigation	Navigating by dynamic taxonomies
Search	metadata search
User related	NONE

Contents will be delivered through a website.

Considering the requirements in the table, which of the following opinions do you consider to be the most correct?

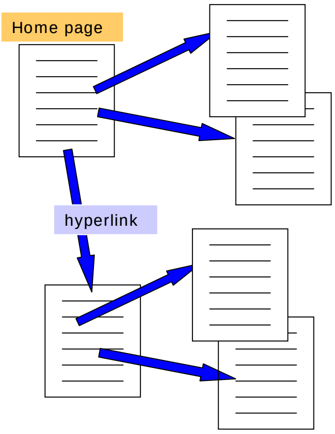
○ “As we chose to deliver contents though a website, I think we should use a database technology”

○ “We don’t need to track user access and we don’t have specific security problems: we don’t need a database at this stage”

○ “We have to allow metadata searches and navigation by taxonomies, and this is not only a collection of documents: we need a database”

Click on your answer

Static website



The simplest way to deliver information online (over the Internet or on a local network) is through a **static website**.

A static website is a simple collection of documents, connected by HTML hyperlinks which are accessed from a web-server by the user’s web browser.

You **don’t need a database** to run a static website, but as a result its **functionality will be limited** (though certainly sufficient to meet most simple information delivery requirements).

Now let’s look at other solutions responding to different requirements…

Full Text Search

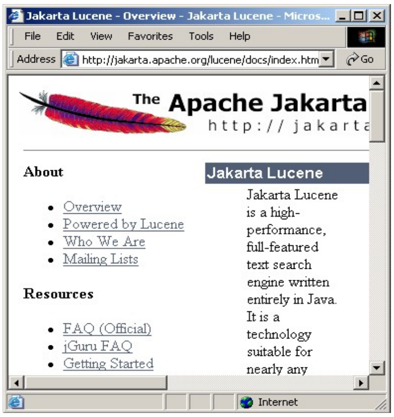
The user might want to search for all documents containing, for example, the word “agriculture”. If the system has to search “agriculture” within all the documents, this will take an unacceptable length of time for hundreds of documents!

A **full text search** is one where the user specifies search terms consisting of words or phrases and obtains documents which contain those words or phrases, subject to the constraints specified by the user.

When you have a requirement for full text searching, it is better to use a system which operates on a prepared **full text index.**

A **full text index** is a cross reference of words with the documents in which they occur**.** It is employed by the search engine to quickly identify documents containing the search terms.

Full Text Search


Jakarta Lucene - Overview - Jakarta Lucene - Micros...
File Edit View Favorites Tools Help
Address http://jakarta.apache.org/lucene/docs/index.htm
Go
The Apache Jakarta
http://jakarta
About
Overview
Powered by Lucene
Who We Are
Mailing Lists
Resources
FAQ (Official)
jGuru FAQ
Getting Started
Internet

Jakarta Lucene

Jakarta Lucene is a high-performance, full-featured text search engine written entirely in Java. It is a technology suitable for nearly any

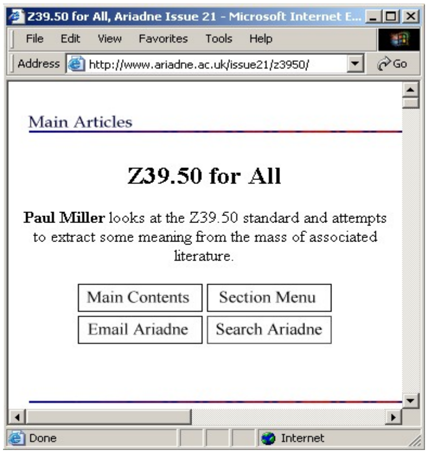
http://jakarta.apache.org/lucene

To allow full text searches you can use **indexing and search engines**, such as Verity, Inktomi and Jakarta Lucene, or **textual databases**, such as ISIS. Most **relational database** systems now incorporate full text indexing.

Features supported by full text index and search engines can include:

- Search with **wildcards** – common conventions are ‘?’ to represent any single character and ‘*’ to represent zero or more characters
- Boolean combinations **AND, OR and NOT** (e.g. Find ‘document’ AND ‘database’)
- Grouping of search terms in Boolean expressions using **brackets** (..)
- **Proximity searches** (e.g. Find ‘document’ within 5 words of ‘database’)

Full Text Search


Z39.50 for All, Ariadne Issue 21 - Microsoft Internet E...
File
Edit
View
Favorites
Tools
Help
Address
http://www.ariadne.ac.uk/issue21/z3950/
Go

Main Articles

Z39.50 for All

Paul Miller looks at the Z39.50 standard and attempts to extract some meaning from the mass of associated literature.

Some features of indexed text search engines are language-dependent, most notably:

Stop words. Common words such as ‘the’, ‘if’ or ‘it’ are excluded from the text index so that they don’t fill up the search results with lots of unwanted hits.

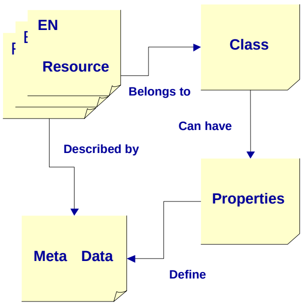
Linguistic stemming creates the text index on the stem (base linguistic form) of words rather than the actual words themselves. This means that a search for a word such as ‘goose’ will also return hits on its plural ‘geese’.

One standard that’s worth a look at is Z39.50. A good place to find an overview of what Z39.50 can do is at:

http://www.ariadne.ac.uk/issue21/z3950.

Metadata search

Databases can be used to store and index the **metadata** that is associated with electronic documents (resources).



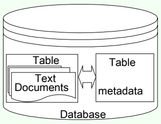
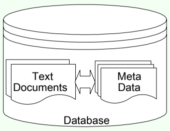
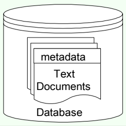
Resources are members of certain classes (e.g. 'technical documents' or 'documents about Agriculture').

Each class can have a number of properties, which define the **metadata slots** that can be filled for any particular resource instance.

So, for example, we know that an instance of a 'technical document' can have a title and subject.

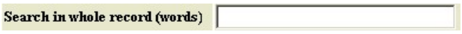
Metadata search

We can implement an **indexed metadata search** using database technology in several ways. Here you can look at three of these:

	**Metadata are in the tables** of a relational database and link to document text held either on the file system or in other tables.
	**Metadata are represented in a structured document** and connected to **the document with which it is associated**.
	If the documents in the database are all **structured XML documents** (or to a limited extent, structured HTML) then we can **embed the metadata in the documents themselves**.

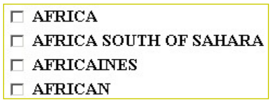
Metadata search

There are a number of different ways in which the metadata search can be implemented for a user:



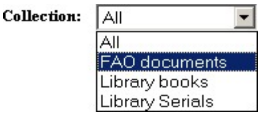
Search in whole record (words)

• **enter search terms as free text**, using terms supported by the query engine;



□ AFRICA
□ AFRICA SOUTH OF SAHARA
□ AFRICAINES
□ AFRICAN

• **select search terms** (available values from vocabularies or ontologies); or

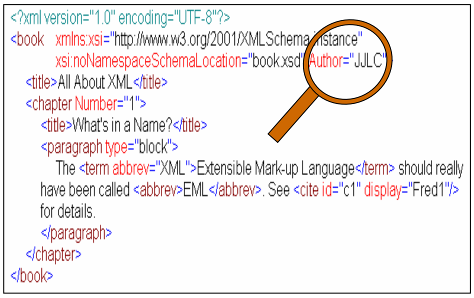


Collection: All

All
FAO documents
Library books
Library Serials

• **specify the class of documents** and then use the properties of that class to define a search form where they can fill out search terms for the allowable metadata slots.

Structured Text Search



```
<?xml version="1.0" encoding="UTF-8"?>
<book   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="book.xsd" Author="JJLC">
   <title>All About XML</title>
   <chapter Number="1">
      <title>What's in a Name?</title>
      <paragraph type="block">
         The <term abbrev="XML">Extensible Mark-up Language</term> should really
      have been called <abbrev>EML</abbrev>. See <cite id="c1" display="Fred1"/>
      for details.
      </paragraph>
   </chapter>
</book>
```

Structured text search allows us to combine some of the full text search features with search terms that incorporate information contained in the structured document mark-up.

To create an efficient search you first need to index the documents in the same way as full text and metadata searches.

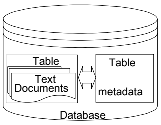
You can make a **structured text search** with HTML documents, provided they are marked up properly.

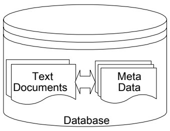
However, better structural search comes from **XML documents**, because in these the mark-up conveys some of the semantics of the document.

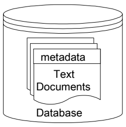
Some modern relational database products (such as Oracle 9.2) can index XML text held in database fields and allow database queries in SQL to be extended into the text and XML structures. There are also native XML database systems available (check out www.xmldb.org) which provide indexed search of XML documents. These native databases mostly use Xpath or the emerging XML Query language (both from the W3C at www.w3.org) to express XML queries.

Structured Text Search

In your opinion, in which of the following scenarios can structured text search be implemented?

○ 

○ 

○ 

Click on your answer

Information portals

In the past few years, organizations and enterprises are increasingly using Information portals.

An Entreprise Information Portal allows integration with applications and services available inside and outside the enterprise: the user can access all services required for his work from a single point, without using different passwords.

This kind of access is provided to all those involved in the enterprise activity, from employees to partners, suppliers and customers. EIP uses web technologies so that all available knowledge is accessible and updatable through a web browser.

Information portals

EIP definition

The following definition of a Enterprise Information Portal (EIP) comes from IBM: “Portals provide a **secure, single point of interaction** with diverse information, business processes and people, personalized to a user's needs and responsibilities”.

Although there is no 'official' specification of what an Enterprise Information Portal should do, it is commonly recognized that most portals have at least the following **five capabilities**:

- Single point of access to resources
- Personalized interaction with portal services
- Federated access to data repositories (information aggregated and categorised to provide a single view to the user)
- Collaboration technologies for group working
- Integration with applications and workflow systems.

Information portals

An information portal can provide **collaborative** tools between employees, partners and suppliers, i.e. workflow management and online community creation.

Most portal systems **provide indexed access to resources**.

Customization is a key element to have a “sticky” portal: the user should be able to choose the contents he/she wants to view in the enterprise portal window, according to his/her personal needs and preferences.

Although you could build your own portal using base technologies (web pages, database and programming tools of your choice), you may find a better return on your investment if you use a **product which has already implemented** the five features listed above.

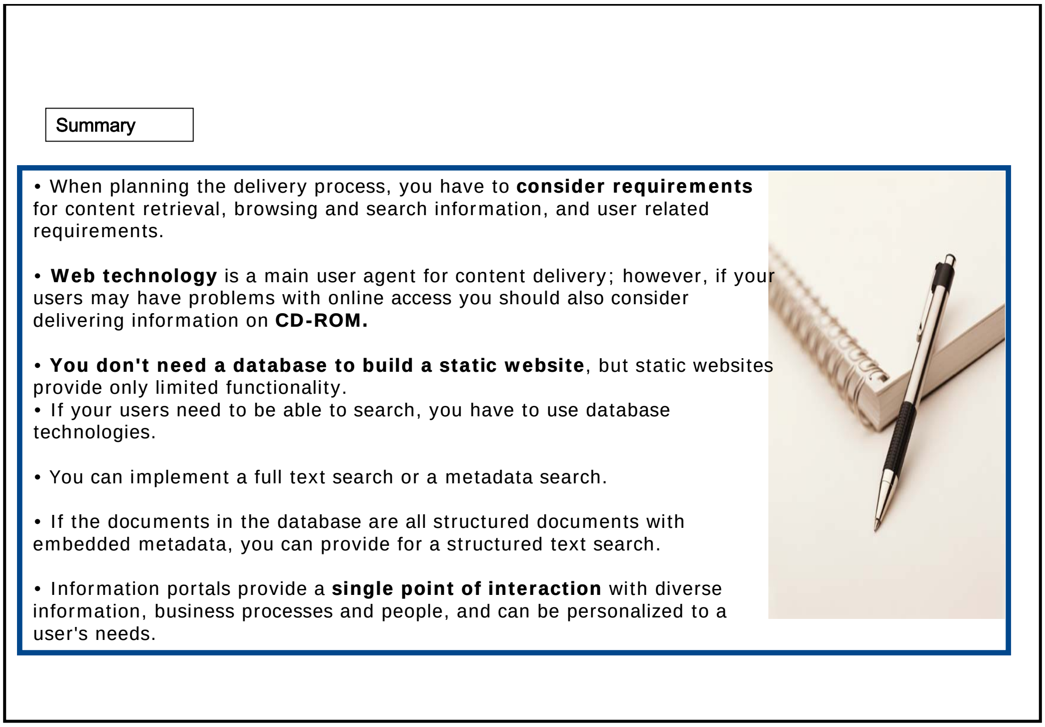
Portal products can be **very expensive**, especially from the major vendors such as IBM, BEA, Oracle and Sun (iPlanet). However, cheaper products are available from Microsoft (MS Sharepoint Server – available for Windows platforms only) or as open source (e.g. the JetSpeed portal from the Apache Project - www.apache.org).

Tools

From here you can download and print a guideline document to list the requirements for information delivery.

Click on the icon to open the document.

Guidelines for requirements analysis.



Summary

- When planning the delivery process, you have to **consider requirements** for content retrieval, browsing and search information, and user related requirements.

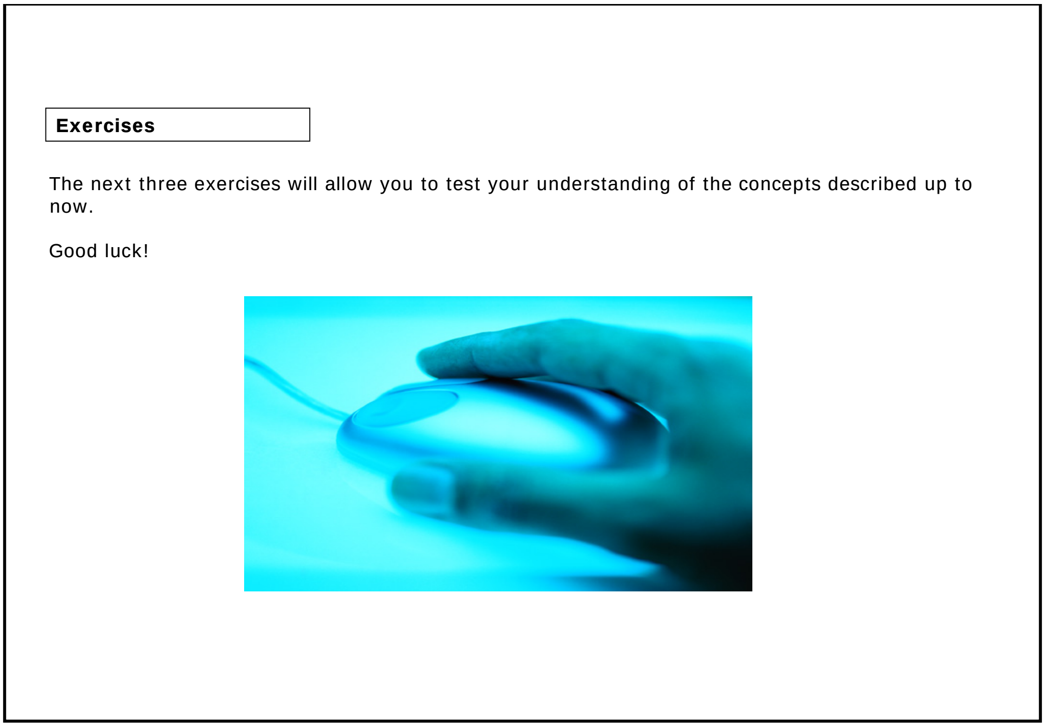
- **Web technology** is a main user agent for content delivery; however, if your users may have problems with online access you should also consider delivering information on **CD-ROM.**

- **You don't need a database to build a static website**, but static websites provide only limited functionality.
- If your users need to be able to search, you have to use database technologies.

- You can implement a full text search or a metadata search.

- If the documents in the database are all structured documents with embedded metadata, you can provide for a structured text search.

- Information portals provide a **single point of interaction** with diverse information, business processes and people, and can be personalized to a user's needs.



Exercises

The next three exercises will allow you to test your understanding of the concepts described up to now.

Good luck!

Exercise 1

You are planning the delivery of your information. After a first analysis of your needs, you draft your requirements.

According to this draft, can you fill your requirements in this table?

> Users will be able to browse XML documents by dynamic taxonomy. We will allow users to make structured text search to find the information (the unit of content provided will be the paragraph of a document).
>
> We must consider their browser setting: 70% use Internet Explorer, 30% use Netscape Navigator 4.

Requirements draft

RETRIEVAL		BROWSING	SEARCH	USER RELATED		
What will be the format of delivered information?	**What will be the level of granularity of delivered information?**	**How will users browse information?**	**How will users search information?**	**Are there user profiles requirements? If yes, what are they?**	**Are there user preferences requirements? If yes, what are they?**	**Are there access control and security requirements? If yes, what are they?**

Exercise 2

How does a full text search work?

○ The system searches for a term within the text of all the documents.

○ The system builds a full text index of the content, and uses it to search.

○ The system searches for a term in the information contained in the structured document mark-up.

Click on your answer

Exercise 3

What does it mean that an Enterprise Information Portal provides a single point of access to resources?

○ Several applications and workflow systems can be integrated in an single system.

○ System can be customized according to preferences of each user.

○ The user can access all services required without using several password.

Click on your answer

If you want to know more…

Leading commercial index/search engines include **Verity** (www.verity.com) and **Inktomi** (www.inktomi.com).

Lucene – a full text serach engine available as open source from the Apache Software Foundation (http://jakarta.apache.org/lucene)

Native XML database systems (check out www.xmldb.org).

Xpath and **XML Query Language** – languages for expressing structured searches in XML documents (both from the W3C at www.w3.org).

JetSpeed – an open source information portal from the Apache Software Foundation (www.apache.org).

The Ariadne magazine, reporting on information service developments and information networking issues worldwide (http://www.ariadne.ac.uk)