

Document management

Patryk Czarnik

materials mainly by
Maciej Ogrodniczuk

XML and Applications 2015/2016
Lecture 13 – 06.06.2016

Why is document management important?

- Because the documents are important.
 - 90% of the information resources of the companies are stored in documents, not in databases. (*Deloitte and Touche*)
- Types of document management systems:
 - Web Content Management System,
 - Enterprise Content Management System (managing company business documents),
 - workflow system,
 - publication system,
 - corporate portal,
 - workgroup system,
 - electronic archive,
 - ...

Document management of yesterday (and today)

- „Traditional” methods of document management:
 - paper workflow (cabinets, binders, office assistants, messengers...),
 - e-mail,
 - floppy disks (ugh!), pen drives, network drives, ...
- Problems (revealing needs):
 - redundancy (the same information duplicated many times) vs. reuse,
 - outdated information,
 - problems with finding the right information,
 - problems with coordinating editorial teams,
 - difficult multimedia publication,
 - no personalization.

Even more problems with documents

- How to manage:
 - large documents?
 - complex documents?
 - valuable documents?
 - long-lasting documents?
 - frequently updated documents?

which are used in:

- geographically dispersed organisations?
 - large-scale organisations (with numerous employees)?
 - highly specialised organisations?
- The solution:
 - content/document management systems (CMS/DMS),
 - search systems (IA, Information Access),
 - knowledge management systems (KMS) (or their simpler equivalents).

Cheap and effective: versioning systems

- Well known to programmers
- Typical functions:
 - central storage,
 - local copies (synchronized with the repository),
 - locking documents for edition and releasing the lock afterwards,
 - document versioning,
 - possibility of simultaneous edition of documents by many people and merging the changes.
- The most popular:
 - CVS (Concurrent Versions System),
 - SVN (Subversion),
 - GIT.

Wiki-like solutions

- ≈ Web pages which can be edited “by anyone”.
 - should work directly in the browser, without any additional plugins,
 - simplified markup syntax can be used for editing.
- Some representatives: MediaWiki, MoinMoin, TiddlyWiki...

Architecture of a typical CMS

- the repository
 - centralised, neutral pool of resources,
- the application:
 - business logic,
 - workflow (process management),
 - search,
 - presentation/publication,
- user interface:
 - navigation,
 - editing system.

Repository functional requirements

- Repository – of documents:
 - possibility to store any document types,
 - versioning,
 - locking documents to edit:
 - pessimistic – conflicts are avoided at any cost, the document is locked immediately after it has been open to edit,
 - optimistic – conflicts are not frequent, so just the modification can be protected,
- – of metadata (information about the document – its authors, publication dates, version numbers...):
 - metadata usually stored outside documents – need of synchronization,
 - most likely: possibility of arbitrary metadata configuration (names, types, labels, display properties, ...)
 - sometimes: structured metadata (lists, hierarchies).

Workflow (or process)

- It's all about “the automation of business processes which involves passing documents, information or tasks between employees according to predefined management procedures”. *Workflow Management Coalition, www.wfmc.org*
- Two methods:
 - the process is being steered by people,
 - the process is triggering actions.
- Setting up the process involves at least definition of:
 - subsequent work phases of the document (workflow states),
 - allowed transitions between states,
 - roles of users authorized to perform actions on the document in a given state.

Two main approaches to document management

- **Content management:**
 - all resources are available for (authorized) users
 - the user can decide which resources he/she uses
 - typical methods of access: navigation, search
- **Process management:**
 - strictly defined roles and competences
 - the user is executing tasks assigned by the system
 - the system passes the document to subsequent users
 - typical method of access: a task list

Variants of CMS/DMS

depending on actual needs...

- Document repository
 - storage and access, often also: versioning and history tracking, access control, metadata, search,
- Office document management (in a company or public institution)
 - tracking status of documents, status changes have formal consequences, access privileges depends on the status
 - often digital documents represent their physical counterparts
- Electronic archive
 - safeness and durability of stored documents is crucial
 - no change allowed, eventually we'll get another version to store
 - sole electronic documents or digitalised forms of something physical
 - different data formats

Variants of CMS/DMS – ctnd.

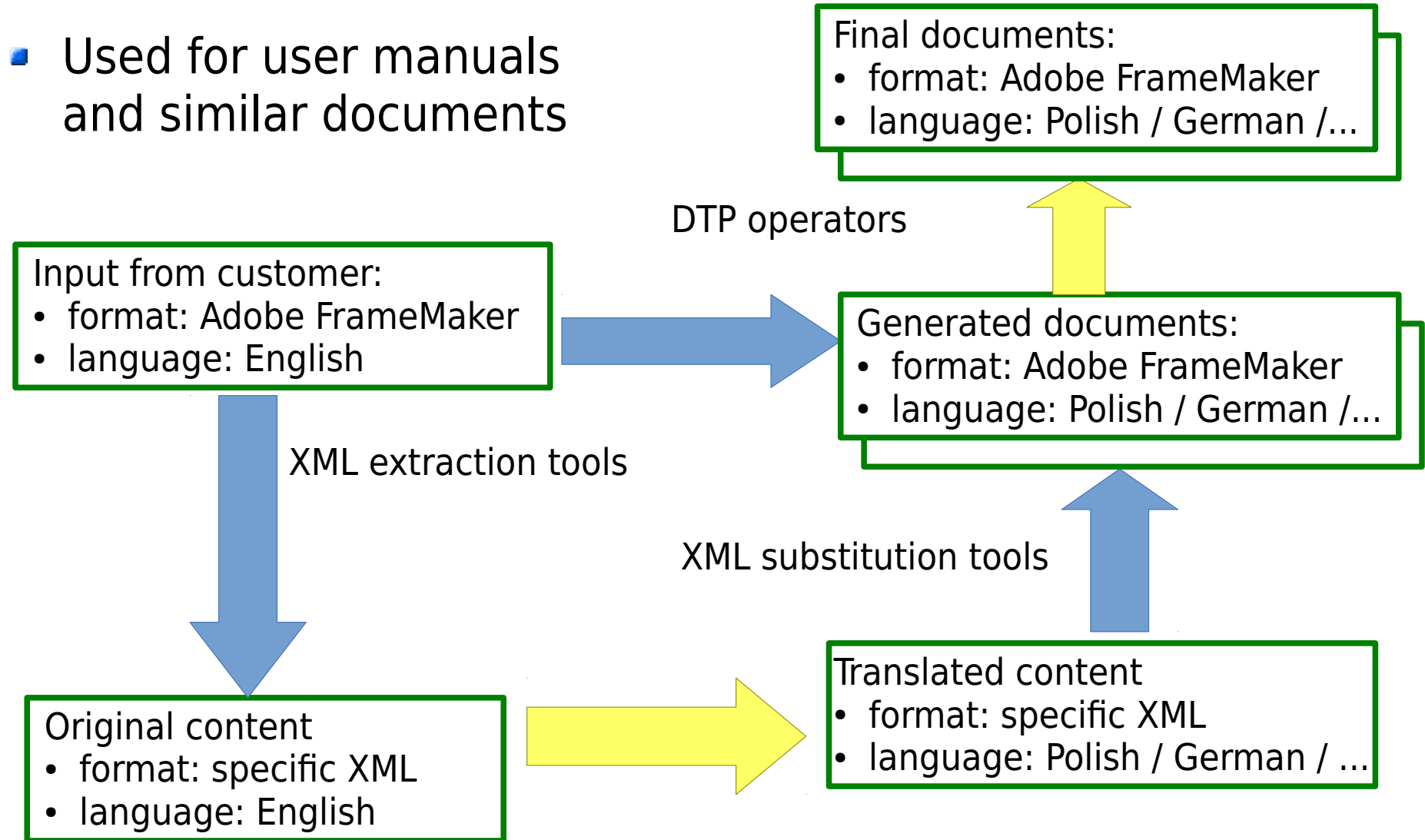
- Publishing system
 - the aim of processing a document is to publish it
 - more presentation/publication-related metadata
 - expected feature – publication tools
 - not so obvious in fact: different means of publication; sometimes we might not want to focus on a single publication, but rather to develop a universal content (knowledge) base
 - content may be shared among documents; rich relations between documents or content fragments
 - advanced content management issues: content variants, etc.
- Web content management
- Universal system
 - flexible, configurable in a high degree
 - more costly in deployment
- Specific system built on demand

Publication

- Should document management system be at the same time the publication system (should contain the publication module)?
 - + it is the publication what we do it for!
 - + storing useful information coming from the typesetting system (e.g. where page breaks appeared) outside CMS does not make sense!
 - – the repository is independent and publication should be maintained by a specialized engine,
 - – document management processes should not depend on the shape of any future publication.

Real use case - translation workflow

- Used for user manuals and similar documents



Translators translate single paragraphs, descriptions, etc. one by one

Office document management

- To facilitate receiving (registering), internal and external distribution of documents.
- Specific issues:
 - the process is subject to internal regulations - detailed
 - description of formal procedures,
 - classification of documents according to subject index,
- IT involved in either way:
 - traditional paper document circulation supported by a system
 - document metadata: documents identified with bar codes, RFID, ...
 - the system stores information on paper document storage (bookcase/shelf),
 - electronic document management:
 - documents are created in electronic form,
 - paper documents are scanned (sometimes even OCR-ed) and saved in the system.

Document archive

- Specific issues:
 - the process conforming to the detailed archiving guidelines,
 - documents added according to the received register,
 - classification of documents according to subject index,
 - archiving categories:
 - A – document with permanent value, to be preserved in the state archive,
 - Bn – document with temporary practical value, stored in the archive for n years (e.g. B50 – 50 years),
 - BEn – document is subject to expert evaluation after n years.
 - library of the archived resources,
 - controlled deletion of documents without any value (to the archive).

Use case: The Presidential Archive

- The system for managing archive resources from 1952 until present.
- Main archive contents:
 - 3 km of paper documents,
 - picture archive,
 - audio/video content (hundreds of hours of recordings).
- Solution:
 - customized system (basing on existing components),
 - dedicated GUI.

Links

- A general link is any type of relation between documents and their content (links = hyperlinks).
- Link types:
 - OO: between documents (treated as a whole),
 - CO: from content to the document (hyperlinks, subdocument inclusion),
 - CC: between content fragments (hyperlinks, version/variant management),
 - uni- or bidirectional,
 - with two or more ends (anchors),
 - described with metadata.
- Link storage options:
 - full link information in the document,
 - identifiers in the document, link information in the database,
 - full information in the database (with paths to document fragments).

Version management

- Purpose: possibility to return to some previous version of the document.
- Multilevel versioning:
 - revisions created automatically at document save:
 - every time,
 - at the release of the lock,
 - **releases** created on demand:
 - at any (crucial) moment of the document life,
 - at publication – to “freeze” all document components.
- Important: not just documents, but also:
 - metadata,
 - links,
 - ...

Variant management

- Variants are documents “differing slightly” and most likely semantically related.
- Two examples:
 - amendments of legal documents,
 - documentation of subsequent versions of some appliance.
- The main idea: avoiding redundancy of document parts common to all variants.

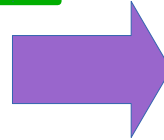
Content variants: an example

- Until 31 December 2010:

```
...  
<article nr="212">  
Book prices are not  
subject to VAT.</article>  
...
```

- After 1 January 2011:

```
<article nr="212">  
Book prices are taxed at  
a rate of 5% VAT.</article>
```



- In the document we insert:

```
<variant id="a3819"/>
```

- In a “versions database” we have:

```
<article nr="212" until="2010-12-31">  
Book prices are not  
subject to VAT.</article>  
<article nr="212" from="2011-01-01">  
Book prices are taxed  
at a rate of 5% VAT.</article>
```

Enterprise search

- Enterprise search is not about searching the Web (i.e. “Google syntax”) or database search, but about indexing, querying and presenting of company documents to authorized users.
- Motto: search engine is the most important component of the document management system.
- Differences, compared with Internet search:
 - limited scope (usually intranet, familiar access rights etc., ...)
 - possibility to build on company standards (e.g. set of common metadata),
 - document ranking not that important,
 - need for searching various sources, various formats,
 - better relevance of the result list,
 - no garbage!

Search process

- User's point of view:
 - target: „find me the best answer for a given question”,
 - enter the query,
 - wait for result list.
- System's point of view:
 - analyze the query,
 - get access to data,
 - analyze data,
 - create the result list, apply user permissions,
 - order results,
 - present the list to the user.

But what is really happening when we search?

- Naive idea:
 - browse documents one by one,
 - if a document contains the element user was searching for (metadata, word, phrase, pattern) – inform the user,
 - if results are to be sorted, collect them in some temporary data structure and display only the matching ones after checking all documents.
- Works for 100 documents. But what for 100,000?
 - Huge amounts of data cannot be searched effectively without indexing the content first.
 - Index is a data structure comparable to what we can find in (good) books, optimized for search and typically containing information:
 - on occurrence of a word in a document,
 - usually also about the exact place where it occurred.

Index properties

Important issues:

- the index must be up-to-date since it is a primary source of information returned to the user,
- frequency and method of synchronization of the index with indexed documents depends on application and technical constraints:
 - when incremental update is not feasible, the whole index must be rebuilt,
 - when systematic update is not possible, cyclical update must be performed,
- the process of indexing can generate additional data useful for result display (e.g. document summaries).

More search issues

- Tokenization
 - split text to words, but sometimes tokens should be shorter than words
 - dziesięciozłotowy
- Stopwords
 - typical approach: they are not important and may be not indexed
 - but sometimes... "this or that"
- Inflected words
 - The problem is not trivial, especially in Polish:
„Dudek, obciąć pensję” vs. „Real obetnie pensję Dudkowi”.
- Spelling hints

Sorting the results

- What does it mean that the documents “matches” the query?
- The place of the result on the list results from many parameters – sometimes not very obvious:
 - occurrence of a word from the query in a document (appeared in the lead = very important),
 - occurrence of a word in metadata, link texts (PageRank),
 - any advance on that?
 - ...
- Most popular model for representing documents and queries:
 - vector space made by all indexed words (each making a separate dimension).

Dialog-controlled search

- When some important search criteria (to be assigned to metadata/attributes in the model) are missing from the query and number of results is high, the system can automatically generate some additional questions to the user.
- The site of a used car dealer's:
 - car make - Audi, Fiat, ...
 - model - make-dependent: A4, A6, A8, TT, ...
 - production year, price, mileage, colour, ...
- A query: Audi for less than 10,000 EUR.
- System help: a form showing additional criteria basing on indexed documents:
 - which model? A4, A6, A8? (no TT at that price),
 - which year?
 - which mileage? less than 100K, 100K-200K, over 200K?

Search user interface

Tips and tricks from designer's notebook (M.O.):

- keep it simple, stupid:
 - no sophisticated help system can fight the intuition of user who would like “to start searching immediately”,
 - graphical design is worth investing in,
 - usability tests are more than necessary (for search form and result list).
- everyone should be happy:
 - most queries are no longer than 3 words and only 5% uses operators,
 - but: advanced users can need more, so “advanced search” is still needed.

More designer's notebook hints

- metadata:
 - use the most important ones (e.g. document type/format) even in the simplest search form,
 - don't overload the single result with metadata.
- result list:
 - add header information: terms which were searched for, number of results, spelling hints etc.,
 - limit the basic layout to most important metadata,
 - show document sizes (to alert users to large documents);
 - if multimedia content is found, add some visual player,
 - show the search context with query terms highlighted to make it clear why a certain result was retrieved,
 - use sorting, paging, grouping, filtering results.

More designer's notebook hints

- less means more:
 - present data from different perspectives right on the first page by categorizing them,
 - allow search in returned results,
 - use tabs and filters to group the results,
 - maintaining taxonomies is costly, but having them is
 - appreciated by users,
- be careful about bells and whistles:
 - a hyperbolic tree is good for a demo, but does it really help in daily work?
 - one new component showing unobvious relations between data (an interesting metadata filter, context links etc.) will always pay for itself,
 - forget the grumblers - user must learn to use the new
 - interface — even the best-tested and most intuitive one.

Search performance and scalability

Two basic methods of dealing with extensive usage of a computer system:

- parallelizing installations,
- modularizing the system (e.g. separating index update from result retrieval, splitting the index into parts).

Note:

- 100% availability of a system is not possible in practice and improving it 10 times (e.g. from 99% to 99,9%) generates 10 times higher costs,
- a ratio of costs of improving availability to losses related to keeping it untouched should be always considered.
- Another important rule: avoid reacting to unrealistic threats.

Search and security

Two methods of controlling search result display:

- indexing content with access permissions (early binding) which automatically excludes protected documents from the result list,
- verification of permissions at resource access (late binding):
 - showing all documents on the list, checking permissions at access,
 - removing restricted documents from the result list even before displaying it,
 - or both!