The LEBONED Metadata Architecture

Frank Oldenettel
Oldenburger Forschungs- und Entwicklungsinstitut für Informatik-Werkzeuge und -Systeme (OFFIS)
Escherweg 2
26121 Oldenburg, Germany
frank.oldenettel@offis.de

Michael Malachinski
Oldenburger Forschungs- und Entwicklungsinstitut für Informatik-Werkzeuge und -Systeme (OFFIS)
Escherweg 2
26121 Oldenburg, Germany
michael.malachinski@offis.de

ABSTRACT

This paper presents the project LEBONED that focuses on the integration of digital libraries and their contents into web-based learning environments. After discussing the several important aspects that have to be considered for this task, we describe in general how the architecture of a standard Learning Management System has to be modified to enable the integration of digital libraries. The *LEBONED Metadata Architecture* is an important part of this modification: it describes the handling of metadata and documents imported from digital libraries. Apart from a general description of this architecture, the contribution focuses on two selected aspects. On the one hand we present the metadata format lxSCORM, a downwards compatible extension of standard SCORM that has been developed to enable the appropriate description and use of imported documents. On the other hand, the fragmentation of imported documents of different types and file formats is a special problem that is not considered by standard Learning Management Systems so far. This paper presents our approach for the extraction of single components (like images) from monolithic documents that enables the appropriate reuse of existing learning materials.

Categories and Subject Descriptors

H.1.2 [**Information Systems**]: Miscellaneous; H.3.7 [**Information Storage and Retrieval**]: Digital Libraries—*Standards*; I.7.2 [**Document and Text Processing**]: Document Preparation—*Format and notation,Multi/mixed media,Standards*; K.3.m [**Computers and Education**]: Miscellaneous

General Terms

Design, Theory

Keywords

Learning Management Systems, LMS, Digital Libraries, SCORM

1. INTRODUCTION

With the permanently growing expansion of the internet, more and more people get access to world wide information resources and realise the internet as a medium which can be used for learning. Temporal and local independence of students during their learning process are considered as the most important benefits of web-based learning. Schools and universities in Europe and the USA are already creating and offering study courses via the internet for a few years [29, 31]. [10] predicts that until 2005 virtual universities will influence the area of e-learning significantly. This initiated the development of numerous web-based learning environments. In these web portals, students can work with electronic teaching materials, join online courses, pass tests, and communicate with other students or instructors. These services are provided by a so-called *Learning Management System (LMS)*. It consists of several components representing different services to be used within a learning environment, like presentation and administration of online courses or testing and assessment functionalities. In the following we consider only a few demands which shall be satisfied by an LMS. Detailed descriptions can be found in [8, 29].

[29] defines several requirements which have to be fulfilled by an LMS. One fundamental demand is the efficient management of teaching materials within a content repository. Contents may appear in several different document types (e. g. books, journals, hypermedia documents) and data formats (e. g. PostScript, PDF). Especially multimedia documents (e. g. audio, video) provide an added value for learning purposes. A flexible, fine-grained data model, called *document model*, describes elementary parts of document structures like chapters, sections, or images and facilitates the access to separate subparts, called *document components*. By utilising this structure reuse of document components can increase benefits for learners and instructors. Learners could take and embed document components in different contexts. Thereby they would be able to create own records of learning material tailored to their specific needs. Instructors would be able to create very specific teaching materials for online courses efficiently by reusing existing document components.

Another important requirement described by [29] is the integration of external knowledge management resources. It seems obvious that *digital libraries* are predestinated for this purpose because materials of many digital libraries are valuable for learning. A look at the very detailed overview of the state of the art of LMS given by [5] shows that unfortunately none of today's existing LMS is able to fulfil this demand.

We think that integrating digital libraries into an LMS will provide benefits for both LMS and digital libraries. On the one hand the availability of teaching materials provided by an LMS could be enlarged by reverting to existing materials. On the other, contents of digital libraries could be used in new contexts. To address this subject we initiated the project *LEBONED*.

1.1 The LEBONED Project

The project LEBONED (*Learning Environment Based on Non Educational Digital Libraries*) which is completely funded by the *DFG (Deutsche Forschungsgemeinschaft)*, the German Research

WWW2003, May 20–24, 2003, Budapest, Hungary.
ACM 1-58113-680-3/03/0005.xxx.

Council, is running since April 2002 and will go on until March 2004. Therefore our work is still in progress and evaluation of our results is not yet completed.

Our main goal is the development of a methodology to integrate digital libraries into LMS. Additionally, we are developing an infrastructure and some essential tools in order to support this task. Every important step will be described extensively in a process model. Because integration of digital libraries will influence several aspects of a conventional LMS in a significant manner, one of our most important sub goals is to develop an appropriate LMS architecture.

The search and retrieval mechanisms of conventional LMS are only suitable for their own content repository, but they are not designed to search in external information sources. To achieve this, the LMS has to be supplemented by these functionalities. We utilise the wrapper concept [19, 28] to obtain homogenous access to heterogenous digital libraries from an LMS.

Unfortunately, file formats and structures of the published documents have been chosen appropriate to publication aspects rather than learning aspects. As a result, most documents delivered from digital libraries are monolithic with poor physical structuring, i. e. they consist only of a single file (e. g. in PDF format) even if they feature a complex logical structure (e. g. chapters, sections, frames, figures, examples, formulas,...). Imagine, for example, the common case of a digital book published as a single PDF file. The book consists of several chapters spread over several pages. It may also contain some images or even multimedia supplements like audio or video clips. In a common LMS, the entire PDF file may be used as one *learning object* (document or document component associated with metadata). But what if only a certain chapter of the book is of interest? Or even an image within that chapter? And what if these components should be reused in another context, e. g. by combining them with components from another book? In order to be appropriate for learning tasks, such monolithic documents have to be fragmented into several smaller components. What should be taken for granted by users is, in fact, a hard technical problem that is ignored by most common LMS. Therefore, one goal of the LEBONED project is to develop concepts and software solutions that enable the identification and physical extraction of document components and structures within monolithic documents. This enables the appropriate use and reuse of document components. In this context, legal issues (e. g. copyright) are also important but beyond the scope of this project.

If documents are found in the connected digital libraries, they have to be accessible from within the learning environment. In order to create learning objects from documents or document components, they have to be imported into the content repository of the LMS. But the content repository is not able to store and manage the learning objects without any further processing. There is a lack of adequate descriptions by domain specific metadata. Since most digital libraries are originally not designed to be part of an LMS, they are only able to provide bibliographic metadata rather than e-learning specific metadata. In this context more and different meta information is needed, like learning topics or the educational level that should be achieved. Metadata fulfilling such demands is described in standards like *Learning Object Metadata (LOM)*[1] [21], *Sharable Content Object Reference Model (SCORM)* [2] [8], *Aviation Industry Computer Based Training Committee (AICC)*[3] or *Instructional Management Systems Project (IMS)*[4]. In [5] these standards are mentioned as the most important ones for learning aspects.

Another reason for missing metadata is that the metadata delivered by digital libraries mostly is assigned to an entire document. Thus there is no specific metadata for learning objects that were created after fragmentation of a monolithic document. Therefore metadata suitable for learning objects has to be added afterwards.

Fragmentation of monolithic documents into learning objects as well as addition of non-existing metadata will be supported by corresponding tools.

The document model which builds the basis for metadata descriptions within the content repository has to be flexible enough to describe the numerous heterogenous document types and data formats which may be delivered by arbitrary digital libraries. Therefore it has to provide appropriate descriptors for various kinds of learning objects.

Handling of metadata and fragmentation of monolithic documents are the main aspects we concentrate on in this contribution. These aspects will be discussed in detail in the next two sections where we present an architecture that describes the processing of metadata and related documents. The fourth section shows the relation of our work to other activities in the areas of digital libraries, e-learning or metadata research and the last section closes this contribution with an outlook.

2. THE LEBONED META DATA ARCHITECTURE

The integration of digital libraries into LMS requires some modifications and extensions of existing standard LMS components (Fig. 1, patterned) as well as the addition of components (Fig. 1, white).

The connected digital libraries are external systems and thus not part of the LMS architecture. Since the wrappers are building specific interfaces for the connection of the digital libraries, one is needed for each library. The wrappers get search queries (initiated by the user) from the *search and retrieval component* and pass them to the digital libraries. If documents from the query results shall be used within the learning environment, the *import component* initiates the wrapper of the respective digital library to access these documents. At the same time, the *metadata import and conversion component* requests metadata so that documents together with corresponding meta information can be gathered into the *content repository* of the LMS. This procedure and the interaction of the components mentioned here is described in detail in the LEBONED Metadata Architecture.

2.1 Metadata Architecture

Our LEBONED Metadata Architecture represents just a specific part within a whole integrative LMS architecture and is indicated through the grey area in Figure 1. The *search and retrieval component* was held apart from this area because our Metadata Architecture starts at the point where concrete documents shall be imported into the content repository of the LMS. At this point the search procedure is already done and the desired documents were already chosen from the query results. Figure 2 shows an illustration of the architecture that will be described in the following.

The import procedure starts with the wrappers that get the desired documents as well as the corresponding metadata from the digital libraries. The documents are directly forwarded to the *document import component* of the LMS. Because the wrappers have to ensure that they provide imported metadata in a unitary form to the

[1] http://ltsc.ieee.org/wg12/

[2] http://www.adlnet.org/index.cfm

[3] http://www.aicc.org

[4] http://www.imsproject.org

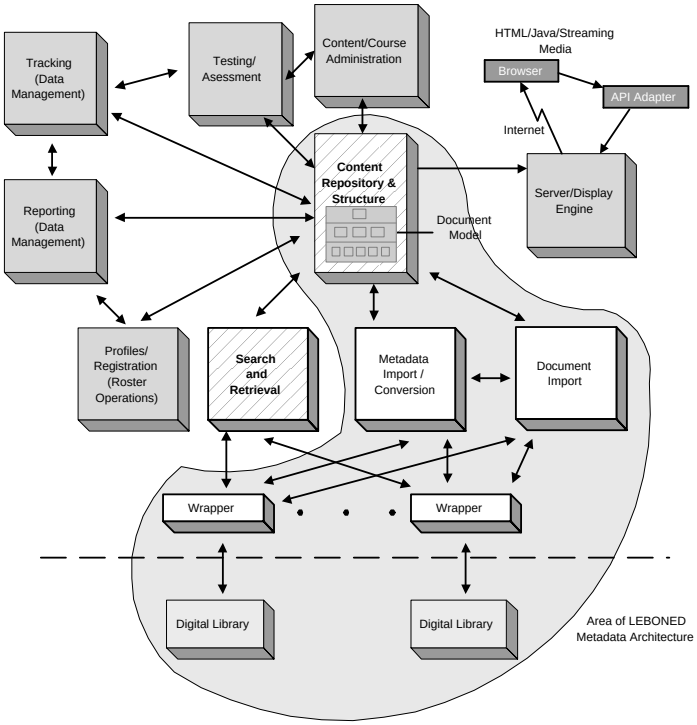

Figure 1: Abstract architecture of an LMS with additional integrative components

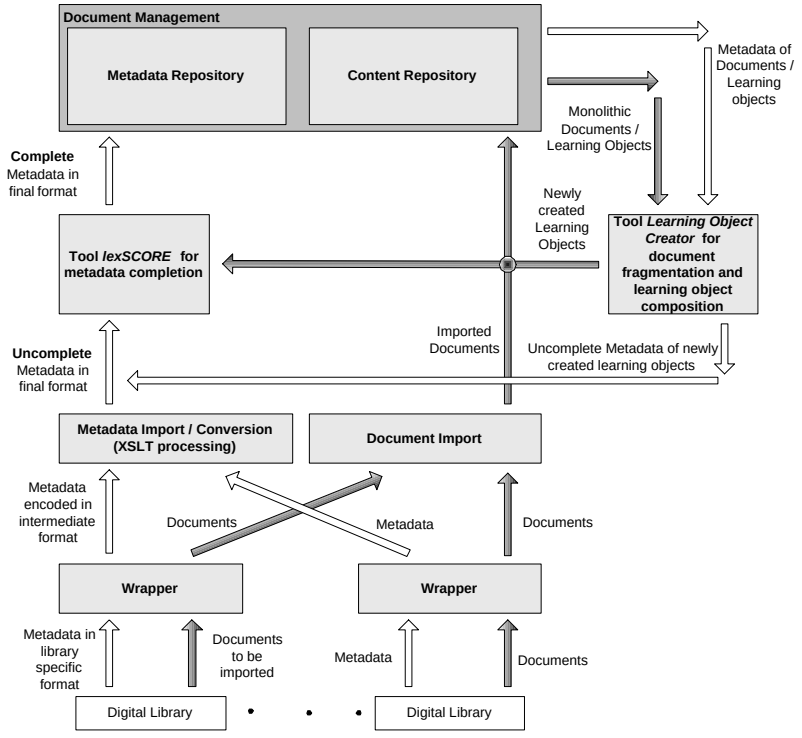

Figure 2: Abstract architecture for metadata and document import

LMS, a first conversion has to be performed. Therefore the wrappers in the LEBONED Metadata Architecture convert the metadata into a simple intermediate XML format (see Sect. 3.1) that supports generic bibliographic metadata but not e-learning aspects. So these wrappers could be kept light-weighted and could also be used to connect digital libraries to other applications than LMS.

The next level above the wrappers is built by the *metadata import and conversion component* and the *document import component*. These components load the metadata descriptions delivered by the wrappers and the corresponding documents respectively. The *metadata import and conversion component* performs another conversion on the metadata and translates it from the intermediate format into the final metadata format that is more suitable for learning needs and on which the metadata repository is based (see Sect. 3.3).

But the metadata presented in the final format must be considered as incomplete because, as already mentioned in Sect. 1.1, digital libraries are only able to deliver bibliographic information. E-learning specific information must be added in order to complete the metadata descriptions. The tool *lexSCORE* supports a human user to add missing metadata. This tool loads the incomplete metadata description as well as entire documents or even single learning objects. By presenting the content of the documents or learning objects and the corresponding metadata fields, it enables comfortable and efficient input of lacking information. Afterwards, *lexSCORE* exports complete metadata descriptions presented in the final metadata format and initiates the storage into the *metadata repository*.

The *content repository* stores the original version of each imported document as a basis for any further processing at arbitrary time. At this point the path taken by metadata and monolithic documents during the integration process from digital libraries to the document management is completed.

Figure 2 shows another path that represents a loop starting and ending at the *document management*. This is taken if smaller learning objects shall be created from monolithic documents or complex learning objects or if simple atomic learning objects shall be composed to more complex ones. For this task we develop the tool *LEBONED Learning Object Creator* that provides functionality to semi-automatically identify and extract components and document structures within monolithic documents. Proceeding from a computer-based analysis of the original document's internal structure, it enables the user to extract the desired pieces from a document and to use these as new learning objects. In fact, this process is executed on a copy of the document while the original remains unchanged in the repository for further use. In addition, the tool provides functionality to compile single learning objects to more complex ones. This enables the creation of new learning materials by arranging existing document components from different contexts. A detailed description of this tool and the underlying concepts is given in Sect. 4. A basic metadata description for any created learning object is generated automatically based on the input metadata of the processed learning objects. This basic description and the new learning objects are passed to *lexSCORE* for manual correction and addition before they can be stored in the *document management*. At this point, the second path of the LEBONED Metadata Architecture is completed.

The intermediate and the final metadata format as well as the identification of components within monolithic documents will be explained in more detail in two following sections.

3. THE LEBONED METADATA FORMATS

In Section 1.1 we already mentioned some standards like LOM, IMS, AICC or SCORM for the declaration of learning specific metadata. Of course, there are also several well known standards in the context of (digital) libraries, like *Dublin Core (DC)*[5], *Encoded Archival Description (EAD)*[6], *Machine Readable Cataloguing (MARC)*[7], *Open Archives Initiative (OAI)*[8] [3] or *Metadata Encoding and Transmission Standard (METS)*[9]. All of these standards were developed by reputable organisations with strong expertise like the *Library of Congress (LOC)*[10] or the *IEEE Learning Technology Standards Committee (LTSC)* [11]. Because of this numerous activities it would not be reasonable to develop yet another proprietary format. Therefore we define the intermediate and the final metadata format of the LEBONED Metadata Architecture on basis of some of these standards. Even if none of them would fulfil all of the requirements of an integrative LMS (see Sect. 3.2 and Sect. 3.3), we can take the one that fits best as a basis for further extensions.

The first format we want to present in the next subsection is the unitary metadata representation that is delivered by the wrappers to the *metadata and conversion component* (Sect. 2.1).

3.1 Intermediate format

The LEBONED intermediate metadata format is a XML representation conform to the *Metadata Encoding and Transmission Standard (METS)*. METS is an initiative of the *DLF (Digital Library Federation)*[12] and is promoted by the LOC. We decided to use METS rather than Dublin Core or the OAI protocol because it is a well structured and very simple format. Since it defines no own metadata descriptors, it is very flexible to use with other standards like Dublin Core, MARC or EAD [11]. This means metadata descriptors of these standards can be encoded directly within a METS document. Short and simple meta descriptions with Dublin Core descriptors are as well possible as large and complex encodings with MARC descriptors.

If we would take Dublin Core as intermediate format, it may be the case that because of the very small element set information would be lost. For instance, it is not possible to recognise the document type (e. g. book, journal, proceeding, article, image) uniquely within a Dublin Core encoded document. Indeed Dublin Core offers the element `TYPE` that can be used like `TYPE=book`, but recommended best practice is to select a value from a controlled vocabulary [16]. The vocabulary suggested by the Dublin Core Metadata Initiative is not very appropriate because the only term that can be taken for document types containing text is `TEXT` [18]. It is not possible to distinguish between several kinds of text documents. Other vocabularies are not defined within the standard.

If a digital library delivers information which cannot be encoded by Dublin Core adequately, we do not want to loose it since it could be helpful for further usage. For this reason a more complex description is necessary, and so we decided to use EAD descriptors within METS because EAD is even more flexible and detailed than MARC but it is much easier to encode in XML [2, 6]. According to the above example, encoding a document type like an image would look like this:

```
<genreform source="gmgpc">
  Photographs
</genreform>
```

[5] http://www.dublincore.org/
[6] http://www.loc.gov/ead/
[7] http://www.loc.gov/marc/
[8] http://www.openarchives.org/
[9] http://www.loc.gov/mets/
[10] http://www.loc.gov/
[11] http://ltsc.ieee.org/
[12] http://www.diglib.org/

The XML attribute `source` specifies a vocabulary or thesaurus from which the entry of the `genreform` element has to be taken. In this example, the value "gmgpc" indicates a thesaurus for graphical material.

The attribute definition of the `genreform` element contains many further values that indicate very detailed concrete vocabularies or thesauri. This enables very precise document type descriptions for different genres. Of course, it is also possible to use different vocabularies with Dublin Core, but in contrast to EAD they are not listed within the standard specification. So the vocabulary is a free choice and it may lead to wrong interpretations if a vocabulary is not known by another application. The document type example is only one among others. A comparison of the element sets of Dublin Core and EAD makes it easy to understand that the complexity of EAD prevents loss of meta information rather than Dublin Core.

A loss of information could also occur if we would use the OAI protocol instead of METS. This protocol uses even unqualified Dublin Core for meta descriptions, i. e. qualifiers which can be used for refinements of element descriptions are omitted [20]. Indeed it is possible to declare more complex metadata descriptions within an OAI record. But this is only optional and the Dublin Core description is always required. This means if more complex descriptions shall be encoded with the OAI protocol, the wrapper has to perform two conversions: one from the specific format of the digital library to Dublin Core and one to the other optional format, what means an increased effort for the wrapper. With METS this is not necessary and so we think that this is the most suitable format for our needs.

A METS encoding consists of the five major sections *Descriptive Metadata*, *Administrative Metadata*, *File Groups*, *Structural Map*, and *Behaviour*. A detailed description of these sections and the corresponding XML elements can be found in [11]. Here we only want to show how the above EAD element could be embedded in a METS description.

```
<dmdSec ID="dmd002">
  <mdWrap MIMETYPE="text/xml"
          MDTYPE="EAD"
          LABEL="EAD Metadata">
    <...>
      <genreform source="gmgpc">
      Photographs
      </genreform>
    </...>
  </mdWrap>
</dmdSec>
```

This is only a very small fragment of the intermediate metadata description generated at the wrapper level of the LEBONED Metadata Architecture. After this processing, the metadata is forwarded to the metadata import and conversion component that produces the final metadata format on which the document architecture of the LMS is based. This final format also relies on existing standards. The next section will discuss which standard is the most appropriate by considering a few requirements that have to be fulfilled.

3.2 Requirements to metadata standards for an integrative LMS

[14] describes several characteristics that are considered to be typical for general metadata. One is, for example, the classification of metadata elements into several categories like *administrative*, *descriptive*, *preservation*, *technical*, and *use*. This is very important for the creation of a detailed and comprehensive metadata description and can be considered as a basic requirement.

Since a metadata standard is the specification for a certain metadata design, it should show these characteristics, too. This applies to all of the mentioned standards. The names of their categories may differ but the meanings are quite similar.

When integrating digital libraries into LMS, on the one hand bibliographic aspects have to be considered. This requirement is fulfilled by the mentioned bibliographic standards in different ways depending on the intended field of application. A minimalist amount of bibliographic metadata elements is given by Dublin Core (see [17]) while MARC ([13]) or EAD ([2]) provide very extensive descriptions. Because of the complexity of the latter ones, they fulfil bibliographic requirements more than sufficient.

On the other hand, for learning aspects more meta information is needed. The existence of metadata elements which describe e. g. the *difficulty level* of a document or the estimated *learning time* are very important. Such elements (and many more) are, of course, part of the e-learning metadata standards LOM, IMS, ARIADNE and SCORM. These standards also support minimalist bibliographic aspects. The bibliographic standards do not support any learning aspects. Therefore these standards can be precluded to be taken into account as a basis for our LEBONED metadata format. From the remaining e-learning standards we decided to take SCORM for our further development, because all these standards are very similar and SCORM was built upon the results IEEE LTCS (LOM), IMS, AICC and ARIADNE and others. Therefore we think that SCORM is the most well-engineered one of the standards. Furthermore, SCORM is not only a simple metadata standard. The so-called *SCORM Run-Time Environment* providing a framework for the implementation of learning resources [9] is also part of the specification and so SCORM builds an ideal basis for our further work. This comprises among other tasks an extension of the metadata specification (*lxSCORM*) because, as shown in the next section, the SCORM metadata element set is not sufficient for several aspects of our approach.

3.3 lxSCORM (leboned extended SCORM)

Authoring of teaching materials is generally done from scratch. In this case, learning objects are created and combined with each other in order to define the physical and logical structures of these teaching materials. For the corresponding metadata descriptions SCORM builds a very comprehensive standard which covers all relevant aspects.

But the creation of learning objects from monolithic documents delivered by digital libraries is different from the normal way of authoring because, as mentioned in Sect. 1.1, these documents are originally not designed to be used within LMS. A standard for the corresponding metadata description has to fulfil some strong requirements that are not fulfilled by SCORM. So we developed *lxSCORM (leboned extended SCORM)* that contains some extensions which are indispensable in order to satisfy these demands. All extensions are declared as additional descriptor elements. None of the existing elements was modified in order to keep lxSCORM downwards compatible to the original standard. Therefore other SCORM conform applications are able to handle lxSCORM descriptions by simply ignoring the additional elements. If we had modified existing descriptors, there would be the danger of misinterpretation. In the following we give some examples of requirements not fulfilled by SCORM and the corresponding lxSCORM declaration.

As mentioned in Sect. 3.1, different digital libraries can provide numerous of different typical document types. For learners it

is very important to distinguish e. g. between books, articles, proceedings, journals, newspapers, audio sequences, or video films in order to be able to choose the appropriate one for a specific learning task. The SCORM element concerning this is presented in Figure 3:

Nr.	Name	Explanation
5.2	Learning Resource Type	Specific kind of Resource, ... Vocabulary Exercise, Simulation, Questionaire, Diagram, Figure, Graph, Index, Slide, Table, Narrative Text, Exam, Experiment, Problem Statement, Self Assesment

Figure 3: SCORM specification of element 5.2. Learning Resource Type

With the terms of the vocabulary it is not possible to describe the document types mentioned above. Therefore it is absolutely necessary to extend this value space in an appropriate way. lxSCORM defines a subelement *5.2.2 Extended Learning Resource Type* with a vocabulary as shown in Figure 4.

Nr.	Name	Explanation
5.2.1	Extended Learning Resource Type	Specifies Learning Resource Type, if vocabulary of element 5.2 is not sufficient. Vocabulary extension: Book, Chapter, Section, Journal, Newspaper, Article, Manual, Manuscript, Lecture, Proceeding, Thesis, Report, CBT, Audio, Video, Electronic Resource, ...

Figure 4: lxScorm specification of element 5.2.2 Extended Learning Resource Type

Another aspect concerns document components created by fragmentation of monolithic documents (see also Sect. 4). It may be that a monolithic document cannot or shall not be fragmented physically (separated into several files) according to its complete logical structure (chapter, section, image etc.). If a file representing a document component is logically structured, it should be possible to describe the logical structure with metadata by defining several learning objects. This is possible with SCORM by the following elements (Fig. 5) in a restricted way.

Nr.	Name	Explanation
4.3	Location	A string that is used to access the resource (URL, URI) [...] This is were the learning resource described by this metadata instance is physically located.
4.3.1	Type	This item specifies the type of the string that may be used to identify the location of a learning resource as used in the location item. [...] At this time there are two restricted strings identified to be used to describe the type: TEXT, URI

Figure 5: SCORM specification of elements 4.3 and 4.3.1

These elements can be used to relate the current learning object to a resource (file). It is also possible to define several different SCORM learning objects referencing the same resource. But how can different learning objects be assigned to different parts of the resources internal logical structure? For example, when several scenes of one video sequence shall be described by several learning objects, it must be possible to declare at which point a scene starts and ends. One way to achieve this with lxSCORM would be to declare a specific URI syntax where the begin and end of the logical subpart is encoded. This would be done within the specification of element *4.3 Location*. But because we decided not to modify any existing elements, we added the two subelements *4.3.2 Begin* and *4.3.3 End* as shown in Figure 6.

Nr.	Name	Explanation
4.3	Location	[...]
4.3.1	Type	[...]
4.3.2	Begin	Indicates the beginning of the part of the physical resource to which this metadata instance is related. Value type depends on item 4.1. Format. [...]
4.3.3	End	Indicates the end of the part of the physical resource to which this metadata instance is related. Value type same as 4.3.2 [...]

Figure 6: lxSCORM specification of element 4.3.x with additional items

The possible entries for these two new elements depend on the document type which is encoded in element *5.2 Learning Resource Type*. If the document type is an audio or video sequence, *4.3.2 Begin* and *4.3.3 End* describe the point of time where the part described by the current learning object starts and where it ends. For other document types the begin and the end of these elements contain unique marks that indicate the corresponding point within the resource. How such a mark looks like depends on the data format (MIME type) of the document described in element *4.1 Format* and may be a tag combination, line number, character number or binary data in case of a binary data format. A user does not have to care how to define a mark. This will be done by *lexSCORE*.

As mentioned above, the aspects presented here are only a few examples. In the following we give some further examples of lxSCORM elements.

- *1.10 lxType:* Indicates that this learning object is (part of) the result of fragmentation of a original monolithic document.
- *4.3.4 Origin:* Indicates from which digital library this learning objects was received.
- *4.3.4.1 Query:* The query (specific to the digital library named in 4.3.4) that was used to get the document from which this learning object was created.
- *4.8 Quality Level:* Abstract description (low, medium, high) of the technical quality of the resource.
- *4.8.1 Type:* Keyword (resolution, sample rate, frame rate) to describe the quality of the media type described by element 4.1.
- *4.8.1.1 Value:* Concrete value depending on the embedding element.
- *7.3 Original document:* Reference to the original (monolithic) document from which this learning object was created. This element is to be used alternatively to element 7.1.

These extensions of SCORM are very important in order to benefit from digital library integration into LMS. Since lxSCORM is downwards compatible to SCORM all of the new elements are defined as *optional* within the corresponding metadata application profiles (see [7]).

4. IDENTIFICATION OF LEARNING OBJECTS WITHIN MONOLITHIC DOCUMENTS

As mentioned in Section 2.1, we create a graphical software tool named *LEBONED Learning Object Creator* that deals with the extraction of smaller components from monolithic documents and the composition of existing learning objects to new learning materials. As important preparatory work, appropriate concepts and technical solutions for computer-based analysis of a document's internal structure had to be developed. These concepts include the identification of components and structures and their physical extraction into a new file of a common file format. The developed concepts and solutions are subject of this section.

Generally, different document types (e. g. print/web publishing documents, video clips, etc.) and file formats of the input document have to be supported, nevertheless trying to provide as general and format-independent concepts and solutions as possible. To make a choice of the document types and file formats to be supported within LEBONED, we analysed several digital libraries (e. g. Digital Library of the ACM, Digital Library of the IEEE, NZDL, California Digital Library) in order to choose the most important types and formats according to frequency of occurence and future tendencies. The final selection includes print and web publishing formats (HTML, PDF, PostScript and RTF), audio sequences (Wave, MP3) and video sequences (MPEG, QuickTime, AVI Video).

In the first step, the selected document types and formats have been analysed in detail with regard to the identification and physical extraction of separate components and structures. The results of this analysis are described in Section 4.1. The concepts and solutions that we developed based on this knowedge are represented in Section 4.2.

4.1 Characteristics of document types and file formats

This section is structured according to the different document types that are analysed. However, as audio and video are both so-called *time-based media* with similar characteristics, they are combined together in one subsection. Each subsection gives a brief overview about the component types that may be contained and that appear appropriate for reuse as separate learning objects.

4.1.1 Print and web publishing formats: HTML, PDF, PostScript and RTF

The file formats belonging to this group are especially used for publishing documents that consist mainly of text (e. g. digital books). Nevertheless, they may also contain components of other media types. In general, this can be images, audio or video sequences, references to other documents or even files of any type (through special mechanisms e. g. embedded files). Furthermore, a document may be structured into separate pages or into several chapters and sections.

It depends strongly on the respective file format not only which kinds of components and structures are generally supported, but also if these may be identified and extracted at all (due to technical feasibility). A complete list of the extractable components and structures for the file formats HTML, PDF, PostScript and RTF is shown in Figure 7.

	HTML	PDF	PostScript	RTF
Images	×	×	×	×
Audio	×	×		
Video	×	×		
References	×	×		
Any type	×	×		
Page structure		×	×	
Chapter structure	×			×

Figure 7: Extractable components and structures in print/web publishing formats

4.1.2 Time-based media: Wave, MP3, MPEG, QuickTime and AVI

For the purpose of learning, it is advantageous to use temporal subsequences of a audio or video clip as single learning objects (e. g. a certain part of a lecture). These subsequences can be specified by start and end time within the entire sequence. However, to enable flexible and easy reuse of a subsequence, it is more appropriate to extract and store it into a separate file. This technical problem is addressed within our project.

Because separating a audio clip into subsequences is reasonably done with regard to the meaning of the content, it is not possible to do this task automatically or even semi-automatically at adequate expense. However, the user may be supported by appropriate visualisation of the audio clip's waveform within the final software tool.

Unlike audio, video clips consist of a sequence of single camera shots and therefore contain a syntactical structure. This syntactical structure can be identified via computer-based analysis at a satisfying level of correctness. For this purpose, conventional shot boundary detection algorithms (e. g. [22]) can detect the single camera shots within a video clip and thereby automatically prestructure the sequence. Based on units of single shots, the user can easily specify and extract the favored subsequences with regard to the meaning of the content.

4.2 Concepts for identification and extraction of components

In this section, the basic decisions and concepts for the task of identification and physical extraction of components and document structures are described in brief. Like the last section, it is structured according to document types.

4.2.1 Concept for print and web publishing formats

One of the most basic decisions concerning the extraction of components from print/web publishing documents is about the input data of the analysis. An approved approach is known as *Document Image Analysis* and performs analysis based on a bitmap image of the document (e. g. identification and extraction of an image from a screenshot of a PDF document). On the one hand this approach is nearly independent of the document's file format, but on the other hand, it may lead to a loss of quality and is not suitable to extract components like audio or video clips at all. Therefore we decided to perform identification and extraction of components based on the document's source code. Generally, this contains either the component's data itself or a reference to this data (e. g. a link). Though this approach is highly format-dependent, it enables extraction of arbitrary components without any loss of quality.

Based on this decision, we developed the object-oriented architecture presented in Figure 8 for the identification and extraction of

components and document structures for print and web publishing file formats in the programming language JAVA. This extendable architecture consists on the one hand of format-independent classes and interfaces that do not only implement common properties and functionality, but also declare common interfaces for uniform and format-independent access. On the other hand, the implemented architecture has been adapted to the selected file formats using derived classes with format-specific implementations. Further adaption to other file formats is possible.

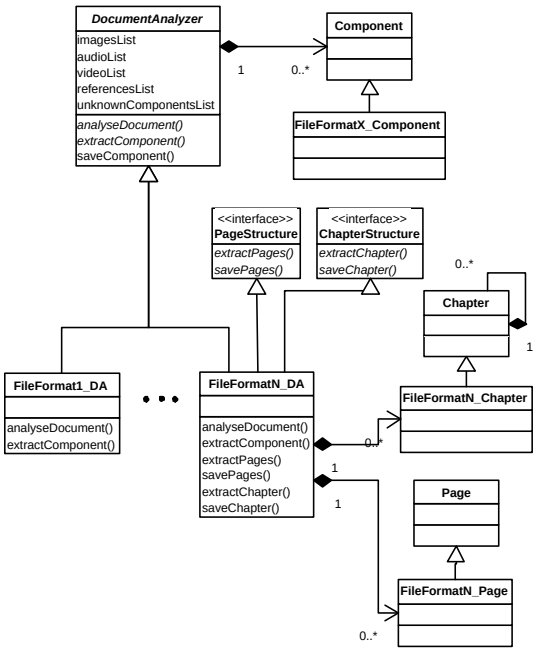

Figure 8: Architecture for publishing file formats

Central component of the architecture is the class DOCUMENTANALYZER that is overriden for each concrete file format. It manages a component list for each component type (e. g. images, videos, ...) and provides default implementations for saving components. Identification and extraction of components and structures is done in the derived classes. Components are modelled by the class COMPONENT (or derived classes) that contains enough metadata to identify and handle a component. Optionally, the classes derived from DOCUMENTANALYZER can implement interfaces that represent document structures and declare appropriate methods to access and extract these. This way, documents consisting of several pages (Interface PAGESTRUCTURE) and documents structured into chapters and sections (CHAPTERSTRUCTURE) can be handled. Single pages resp. chapters are modelled by PAGE- resp. CHAPTER-objects (or derived classes).

For the purpose of identification and extraction of components and structures, we use source code parsers (either third-party or proprietary) as a basic instrument. In detail, we use the HTML-parser provided with the JAVA class library (JAVAX.SWING.TEXT.-HTML.HTMLEDITORKIT.PARSER) to analyse HTML documents, a proprietary implementation for parsing RTF files and the package Etymon Pj[13] to analyse PDF documents. The PostScript file format on the one hand turned out to be very hard to handle and on the other hand can easily be converted to PDF, so we decided to implicitly convert it to PDF. While the use of parsers makes identification of components quite an easy task, the real data extraction itself and storage in common file formats often require special post-processing routines like decompression and interpretation of raw data.

Figure 9 shows the components and page structure of a sample PDF file that has been analysed with our system.

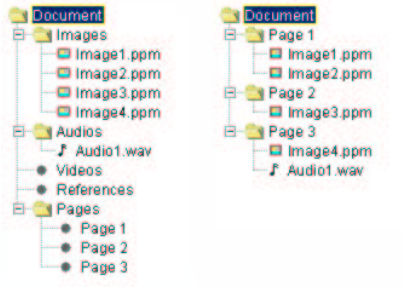

Figure 9: Example of an analysed PDF document. L. component view, r. structure view

4.2.2 Concept for time-based media

The requirement to extract and store subsequences of audio and video clips into a single new file has been identified as a basic prerequisite. Our solution to this problem is based on the Java Media Framework[14] (JMF) offered by Sun Microsystems. The JMF allows working with time-based media on a high level of abstraction and therefore mostly format-independent. The component we developed offers an easy-to-use interface to copy subsequences (defined by start and end time) of audio or video clips into new files.

For a video sequence, detection of its syntactical structure on the level of single shots is done using a shot boundary detection algorithm. Basically, these algorithms extract certain features for each frame of the video and compare the feature differences between consecutive frames to a threshold. Exceeding the threshold is interpreted as a shot boundary. Our implementation is a slight modification of the shot boundary detection algorithm developed within the project AVAnTA at the TZI Bremen [23].

The detected syntactical structure of a video clip is presented to the user in a graphical user interface that also provides functionality to correct shots due to faulty recognition. Furthermore, the user can specify and extract subsequences on the level of single shots according to his special needs. Figure 10 shows the prototypical GUI that presents the syntactical structure of a clip to the user and offers the described functionality.

Figure 10: GUI presenting the syntactical structure of a video

[13] http://www.etymon.com/pj

[14] http://java.sun.com/products/java-media/jmf/

5. RELATED WORK

One of the major activities to be named first is the *ARIADNE* project[15] which is funded by the European Union Commission. This project also deals with interconnected knowledge pools. But the integration of existing digital libraries is considered not as detailed as our project does, therefore our work will be an adequate addition to this.

Furthermore, several existing LMS have to be mentioned. [5] gives a comprehensive overview of the state of the art. In this context special attention has to be paid to the LMS *Blackboard* [34]. It provides a component based architecture called Building Blocks. Since published APIs are available, this software suite supplies good prerequisites for our work.

Because major activities of our work are dealing with metadata, works about the different standards (Sect. 1.1) have to be considered. [12] for instance describes the adaptive hypermedia system *Multibook* used to teach multimedia technology. This application uses metadata to create course sequences semi-automatically. What especially has to be mentioned in the context of document models is the work of the UC Berkeley that has developed the *Multivalent Document Model (MVD)*. The aim of this document model is to support annotations and cooperative work on documents [33]. The goal of the *CUBER* project is to develop a system that supports learners in searching higher education materials from European universities. In this project, metadata also plays an important role and the work also led to modifications of the original LOM schema [27]. Another work described in [30] considers the reusability and adaptability aspects of interactive multimedia content in web-based learning systems. Here dynamic metadata has to be managed which also made extensions of the LOM schema necessary.

In the area of digital library research we are touching the aspect of distributed digital libraries. The *New Zealand Digital Library (NZDL)*[16] project of the university of Waikato, New Zealand, has developed the software suite *Greenstone*. It enables users to create own distributed digital library collections [4]. Another digital library distributed over all campuses of the University of California is the *California Digital Library (CDL)*[17]. One goal of this project is the seamless integration of distributed resources and providing access to them [24]. Very closely related to the CDL is the *Stanford Digital Library Technologies Project*[18] which created the *InfoBus* technology. InfoBus is a CORBA based infrastructure that enables the integration of distributed heterogeneous digital library collections [26]. Because of the high flexibility of the InfoBus technology, it is taken into special consideration for our work. This applies for the *DAFFODIL* project, too. DAFFODIL is an initiative of the university of Dortmund, Germany. The main goal is to develop an agent-based infrastructure for federated digital libraries [15].

A project dealing with infrastructures and tools for the reuse of learning objects is *Teachware on Demand*[19]. Goal of this project is the automatic composition of user-specific learning materials from existing fragments (like text, images or videos). Unlike our project, it assumes that these fragments are created from scratch for this purpose and does not support the fragmentation of existing monolithic documents.

Much work has been done in the field of document analysis. In contrast to our approach, most of these solutions are based on the analysis of document images. An overview of such algorithms for layout detection is given in [25], a work dealing with the recognition of logical structures in document images is [32]. An approach for document analysis based on page description languages is AIDAS that deals with the identification of the logical structure of PDF documents [1].

[15]http://www.ariadne-eu.org/main.html
[16]http://www.nzdl.org
[17]http://www.cdlib.org/
[18]http://www-diglib.stanford.edu
[19]http://www.teachware-on-demand.de

6. CONCLUSIONS AND FURTHER WORK

In this contribution we presented some results of the project LEBONED. The main goal of this project is to find a general solution for the integration of digital libraries into LMS. We briefly described how an original LMS architecture has to be modified to enable the integration task. The developed LEBONED Metadata Architecture is an important part of such a modification. It provides a solution for the handling of metadata and documents during the document import and further processing. We presented in more detail our solutions for the aspects of metadata formats and identification of document components (learning objects). The described metadata formats supplement the LEBONED Metadata Architecture appropriately. Especially the development of lxSCORM is an important step because it takes the special requirements into consideration that occur when working with documents imported from digital libraries. In the same context, the identification and extraction of components and structures within monolithic documents is a basic prerequisite for the appropriate use and reuse of learning materials. Our concepts provide an extendable solution for this important problem.

Currently, we are implementing the lxSCORM format as XML schema specification. After that we will implement XML templates, which will be the basis for the processing of the *metadata import and conversion component* and the metadata tool. For future activities we intend to provide the complete lxSCORM specification for further consideration to organisations like ADL or ARIADNE.

The described concepts and solutions to identify and physically extract components from monolithic documents are almost completely implemented. These software components are most important preparatory work for the development of the *Learning Object Creator* that will be the next step. This tool will not only support the extraction of components and document structures for several document types and file formats in a convenient manner, but also contain authoring functionality for the creation of new learning materials by composition of existing learning objects.

Furthermore, we will work on other aspects of the metadata architecture that were not considered in detail here. This will be for instance the mentioned tools and the wrappers. The tools have to be implemented and for the wrappers we are currently developing a framework which will reduce the effort of implementation to a minimum. Since the LEBONED Metadata Architecture is only one part of an entire integrative LMS architecture, we will work on the concrete implementation in the near future.

At last we will note down all our experience of the LEBONED project into a process model which will describe the whole integration process. This shall give advise for future intentions of digital library integration into LMS.

7. REFERENCES

[1] A. Anjewierden. Aidas: Incremental logical structure discovery in pdf documents. In *6th International Conference on Document Analysis and Recognition (ICDAR)*, pages 374–378, Seattle, September 2001.

[2] S. o. A. Archivists. Application guidelines for version 1.0 - encoded archival description (ead) - document type

definition (dtd), version 1.0 - technical document no. 3. http://lcweb.loc.gov/ead/ag/aghome.html, Society of American Archivists, 1999.

[3] W. Y. Arms. *Digital Libraries*. MIT Press, 2000.

[4] D. Bainbridge, D. McKay, and I. H. Witten. Greenstone digital library - developer's guide. Manual, University of Waikato, New Zealand, Okt. 2001 2001.

[5] P. Baumgartner, H. Häferle, and K. Maier-Häferle. *E-Learning Praxishandbuch - Auswahl von Lernplattformen*. StudienVerlag, 2002.

[6] N. Development and L. MARC Standards Office. Development of the encoded archival description document type definition. http://lcweb.loc.gov/ead/eadback.html, 1998.

[7] P. Dodds. Sharable content object reference model (scorm) - version 1.2 - the scorm content aggregation model. Specification, Advanced Distributed Learning (ADL), 1st Oct. 2001 2001.

[8] P. Dodds. Sharable content object reference model (scorm) - version 1.2 - the scorm overview. Specification, Advanced Distributed Learning (ADL), 1st Oct. 2001 2001.

[9] P. Dodds. Sharable content object reference model (scorm) - version 1.2 - the scorm run-time environment. Specification, Advanced Distributed Learning (ADL), 1st Oct. 2001 2001.

[10] J. Encarnaçao, W. Leidhold, and A. Reuter. Szenario: Die universität im jahre 2005. In B. Stiftung and H. N. Stiftung, editors, *Studium online: Hochschulentwicklung durch neue Medien*. 1999.

[11] D. L. Federation. Mets: An overview and tutorial. http://www.loc.gov/standards/mets/metsoverview.html, Digital Library Federation, 2002.

[12] S. Fischer. Course and exercise sequencing using metadata in adaptive hypermedia learning systems. *Journal of Educational Resources in Computing (JERIC)*, 1(1es):5, 2001.

[13] B. Furrie. *Understanding Marc Bibliographic: Machine-Readable Cataloging*. Library of Congress, 6th edition (january 2001) edition, 2001.

[14] T. Gill, A. Gilliland-Swetland, and M. Baca. *Introduction to Metadata - Pathways to Digital Information*. Getty Information Inst, 1998.

[15] N. Gövert, N. Fuhr, and C.-P. Klas. Daffodil: Distributed agents for user-friendly access of digital libraries. In *European Conference on Digital Libraries (ECDL)*, pages 352–355, 2000.

[16] D. Hillmann. Using dublin core. http://dublincore.org/documents/usageguide/, Dublin Core Metadata Initiative, 2001.

[17] D. C. M. Initiative. Dublin core metadata element set, version 1.1: Reference description. http://dublincore.org/documents/1999/07/02/dces/, Dublin Core Metadata Initiative, 1999.

[18] D. C. M. Initiative. Dcmi type vocabulary. http://www.dublincore.org/documents/2000/07/11/dcmi-type-vocabulary/, Dublin Core Metadata Initiative, 2000.

[19] C. A. Knoblock, K. Lerman, S. Minton, and I. Muslea. A Machine Learning Approach to Accurately and Reliably Extracting Data from the Web. *IEEE Data Engineering Bulletin*, 23(4):33–41, 2000.

[20] C. Lagoze and H. V. d. Sompel. The open archives initiative. In *International Conference on Digital Libraries Proceedings of the first ACM/IEEE-CS joint conference on Digital libraries*, pages 54 – 62, Roanoke, Virginia, United States, 2001. ACM Press.

[21] I. Learning Technology Standardization Committee. Draft standard for learning object metadata. Draft Standard IEEE P1484.12/D6.1, Institute of Electrical and Electronics Engineers, Inc., 18.April 2001.

[22] R. Lienhart. Comparison of automatic shot boundary detection algorithms. In M. M. Yeung, B.-L. Yeo, and C. A. Bouman, editors, *Storage and Retrieval for Image and Video Databases VII*, volume 3656 of *Proc. SPIE*, pages 290–301. 1998.

[23] A. Miene, A. Dammeyer, T. Hermes, and O. Herzog. Advanced and adapted shot boundary detection. In D. W. Fellner, N. Fuhr, and I. Witten, editors, *Proc. of ECDL WS Generalized Documents*, pages 39–43, 2001.

[24] J. Ober. The california digital library. *D-Lib Magazine*, 5(3), 1999.

[25] O. Okun, D. Doermann, and M. Pietikinen. Page segmentation and zone classification: a brief analysis of algorithms. In *Information Science Innovations (ISI'2001)*, Proc. of the International Workshop on Document Image Analysis and Understanding, pages 98–104, American University in Dubai, March 2001. UAE.

[26] A. Paepke, M. Baldonado, C.-C. K. Chang, S. Cousins, and H. Garcia-Molina. Building the infobus: A review of technical choices in the stanford digital library project. Working paper, Stanford University, 18. Jan. 2000.

[27] P. Pöyry, K. Pelto-Aho, and J. Puustjärvi. The role of metadata in the cuber system. In *Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology*, ACM International Conference Proceeding Series, pages 172–178, Port Elizabeth, South Africa, 2002. South African Institute for Computer Scientists and Information Technologists.

[28] S. Pulkowski. Intelligent wrapping of information sources getting ready for the electronis market. In *Proceedings of the 10th VALA Conference on Technologies for the Hybrid Library, Melbourne, Australia*, pages 113–124, 2000.

[29] M. J. Rosenberg. *e-Learning - Strategies for Delivering Knowledge in the Digital Age*. McGraw-Hill, 2001.

[30] A. E. Saddik, S. Fischer, and R. Steinmetz. Reusability and adaptability of interactive resources in web-based educational systems. *Journal of Educational Resources in Computing (JERIC)*, 1(1es):4, 2001.

[31] R. Schulmeister. *Virtuelle Universität, Virtuelles Lernen*. Oldenbourg Verlag, 2001.

[32] K. Summers. *Automatic Discovery of Logical Document Structure*. PhD thesis, Cornell Computer Science Department, 1998. Technical Report TR98-1698.

[33] R. Wilensky and T. A. Phelps. Multivalent documents: A new model for digital documents. Tech. Report CSD-98-999, University of California, Berkeley, 1998.

[34] D. Yaskin and S. Gilfus. Blackboard 5 - introducing the blackboard 5 learning system. White paper, Blackboard, 2001.