LEARNING FOR E-LEARNING

Carsten Lecon[1] and Marc Hermann[2]

[1]Department of Computer Science, Media Computer Science,
Aalen University, Germany
[2]Department of Computer Science, User Experience,
Aalen University, Germany

ABSTRACT

In response to the heterogeneity of previous knowledge of the students when beginning their studies, we present a solution, where undergraduate students as well as advanced students (hopefully) will benefit from 'AdLeR' (Additive Learning Resources): A tool for the rapid generation of small e-learning courses. The undergraduates can catch up lack of knowledge by our mini courses (self-regulated). The advanced students are involved in the development of our tool or in the creation process of learning material, which is suited for self-regulated learning. When implementing the tool, the students have to deal with various aspects of computer science domains for example, which consolidates their knowledge and their competences.

KEYWORDS

E-Learning, self-regulated learning, learning by teaching, XML, learning path, search functionality.

1. INTRODUCTION

In this paper, we address two challenges of higher education, especially for computer science. First: Due to the increasing heterogeneity of the students (for example in Germany, now also technicians are allowed to study), there exist different levels of knowledge at the beginning of their studies. By this, there exist different levels of knowledge at the beginning of their studies. With the Bologna-Reform put into practice, the teacher nowadays cannot respond flexible to the individual needs of the individuals due to the rigid curriculum. Although tutorials can help, these are not sufficient. An appropriate solution is to offer individual e-learning courses in the sense of microteaching. These are learning units, which cover specific learning matters. Learning paths offer a multimodal access to the learning material, which supports the self-regulated learning. Second: the students themselves (students in higher semesters) can compile just these learning objects (teaching units, exercises, etc.). In order to create 'mini courses', we use an easy-to-use tool: 'AdLeR' (*Additive Learning Resources)*. The development process of this tool covers many disciplines of the curriculum: Software Engineering/ Programming (the tool has to be implemented), human-computer-interaction (in order to allow a good-looking, motivating, as well as functional screen), formal languages (a subfield of the big topic 'theoretical computer science') for a flexible search functionality. All these subjects are handled in computer science lectures. By implementing our tool, these subjects can be deepened by the practical application, which will also lead to better and sustainable competence in the respective areas.

Up to now, most of the university teaching is organized like depicted in figure 1 (left side): the teachers present the learning matter in (presence) lectures, which take place in lecture halls, seminar rooms or labs. The students have access to the learning material by – for example – a learning management system (LMS). Because this system is available online, the students use these data outside of the campus of the university as well. Furthermore, especially in companies, virtual worlds (virtual reality, VR) are used for training and learning. Martín-Gutiérrez, Mora, Añorbe-Díaz and González-Marrero give an overview of such learning scenarios [11], Lueckemeyer presents a concrete VR learning setting for teaching programming languages [10].

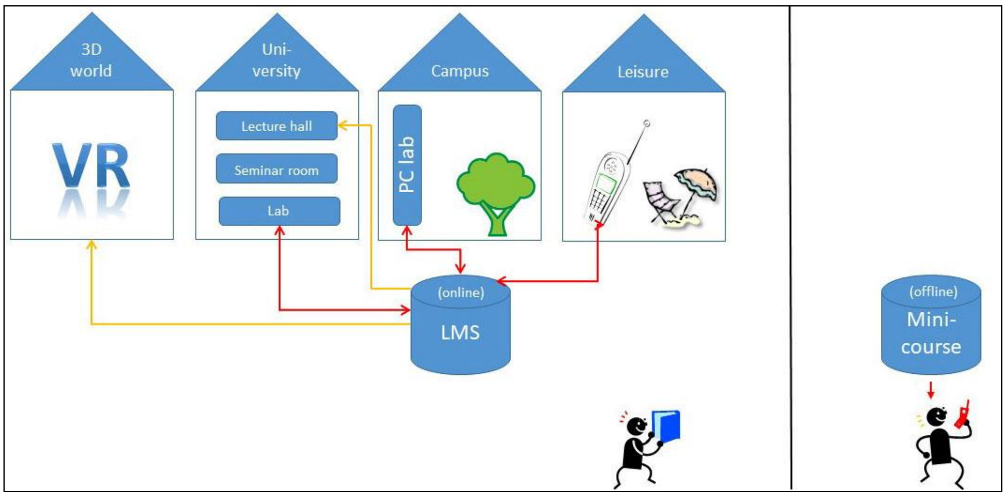


Figure 1: Learning places

Reading (real) books seems to be somewhat outdated since the availability of web sources, all needed information seems to be found in the internet (with the risk of incomprehension by 'fast food learning' – see section 2). Therefore, we propose (electronic) mini courses for the above mentioned self-regulated learning, which offer self-contained learning material. By this, learning outside the university infrastructure is possible. However, because in this way the safe environment of the university is left (see figure 1, right side), we have to consider some didactic issues in order to ensure a positive learning effect as much as possible.

The rest of the paper is organized as follows: First, we outline the didactic concepts, which are used in this work (section 2). Then we describe our tool 'AdLeR' (section 3).In this chapter, we also describe one important feature of this tool: learning paths (sometimes named 'learning trail', subsection 3.2) and aspects of human-computer-interaction (subsection 3.4). In our tool not only a classical full text search is available, but also a flexible search functionality, which will be described in chapter 4. The paper ends with as short summary and an outlook (section 5).

2. DIDACTIC CONCEPT AND RELATED WORK

It is a challenging approach to offer a kind of learning environment for students which are used to visit presence lectures at the university. This particularly makes sense, if the students get additional learning material just when there is a need: In this case the most intrinsic motivation of the students can be expected. Generally, in our approach several didactic concepts play a role and should be considered when developing learning material for the mini courses and when implementing the tool for generating the course.

In our paper, we consider the following didactic concepts: self-regulated learning, microteaching, blended learning, inverted/ flipped classroom, learning by teaching. In this project, one can often find these concepts in combination.

With the mini courses, we implement some important aspects of e-learning: learning at any place and at any location. This requires self-regulated learning of the students [13], although this term still is ambivalent ([5], [14]). In our context, we expect from the learners (students) to organize themselves and to be aware of their learning behaviour. This sometimes could be a serious problem for the students, which we try to address by the following steps (extract):

- The learning matter of the mini course fits the lacks (for example missing previous knowledge)
- The length of the courses is short
- The course offers possibilities of self-evaluation by quizzes
- A multimodal access to the learning units is possible, for example by learning paths (see subsection 3.3)

The above mentioned length of the course leads to the didactic concept *microteaching* [8], [2]. In [2] one can find an appropriate definition:

‘Microteaching is a teaching situation which is scaled down in terms of time and number of students. […] The lesson is scaled down to reduce some of the complexities of the teaching act, thus allowing the teacher to focus on selected aspects of teaching. […]’

This definition reflects our intention: students can learn the subjects in small portions and just when a specific knowledge is needed. This can be done at any place, for example using a mobile phone when commuting to the university. However, a challenge is to avoid ‘fast food learning’ (quickly consuming learning along the way); that means that some topics cannot be treated without previous knowledge. For example, it would be difficult to understand the JPEG compression method using the DCT (discrete cosines transformation) without a fundamental knowledge about trigonometry. The elements of a microteaching learning material can be: texts, short videos, animations, pictures as well as hyperlinks to opportune sites in the internet. In addition, the learners can review their acquired knowledge by performing quizzes. Therefore, every mini course should begin with the required previous knowledge (eventually measured by a short initial test).

Another more general concept is the *Blended Learning* (see for example [7]): The students visit lessons at the university, and also learn by electronic learning objects (self-regulated) – if necessary. In contrast to the usual variant of blended learning, here most of the study matter is presented in presence lessons whereas the electronic material is an (optional) enrichment.

The aim to support all students is also the idea of the didactic concept ‘inverted classroom’/ ‘flipped classroom’ [4]: Here, the learning is transferred completely to electronic material, whilst the presence phases are used for repeating and applying the learning matter. The motivation for this is similar to our observation, that ‘not all students learn in the same way at the same pace’ [6]. This concept is promising, but is rarely used nowadays, because a great effort is necessary, for example, the learning materials have to be prepared for self-study considering various kinds of learning behaviour. If the learners are not motivated, the worst case could be, that they participate in the presence phase without having learnt anything.

The didactic concepts described so far, refer to consuming (additional) learning material. From another point of view, the persons – in our case students – who create learning contents, apply the didactic concept ‘*learning by teaching*’ (see for example [3], [18]). This is like a tutorial, but electronically and without face-to-face communication. Therefore, this is a special challenge for the students, because the teaching content has to be prepared for self-regulated learning by themselves.

3. THE TOOL 'ADLER'

In this section, we present our tool for generating mini courses. First, we give a motivation and an overview of our tool (subsection 3.1). The structure and the content of such courses is specified by an XML specification (subsection 3.2). The significance and realization of learning paths will be described in subsection 3.3. Some aspects of the human-computer-interaction, which is relevant for the screen design and for the user interface of our tool, are addressed in subsection 3.4.

3.1. Introduction to 'AdLeR'

For the above mentioned reasons, we have decided to offer additional (or deepening) learning material. This should not be a complete e-learning course, but a summary of a specific learning matter. We name this sequence of HTML pages 'mini course'. Because this is an addition to the curriculum, we named our tool *Additional Learning Resources* – shortened '*AdLeR*' (the German word for *eagle*).

The aim of this tool is not necessarily the creation of new learning material. Instead, our tool is able to use existing assets in order to combine this media to a mini course. Therefore, the assets can be used in multiple ways.

For this, we have implemented a prototype for the generation of 'mini courses' in the sense of microteaching. The structure (chapter, subsection, hyperlinks, etc.) and the content (assets like text, image, video, quiz, etc.) is specified in an XML document (see subsection 3.2).
The general mechanism of the tool is depicted in figure 2.

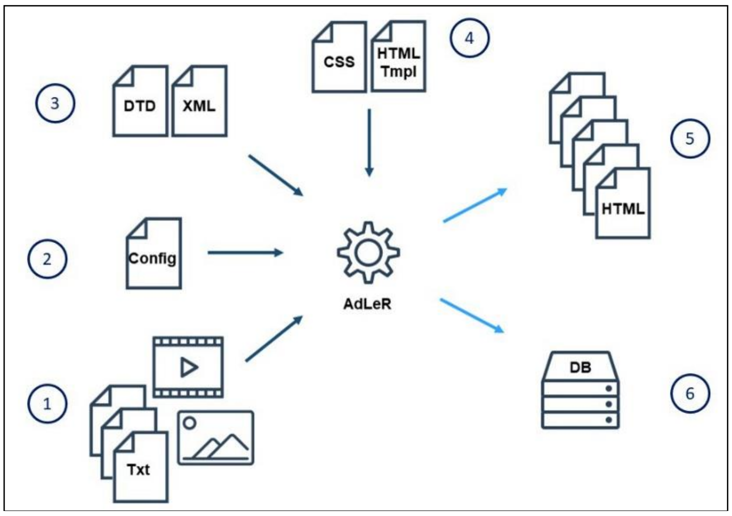


Figure 2: General mechanism of the tool 'AdLeR'

The programming language for this tool is Java. The graphical user interface for the end user is written in C++.

A generated HTML page (5) of the generated course consists of media assets (1) pages, which are specified in an XML document (3). For example, the syntax for a picture looks like this:

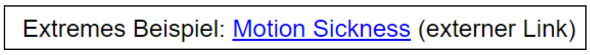
```
<content type="image" file="hmd-overwiew.jpg"
               height="300" width="400" alt="HMD Overview">
</content>
```

In this case, the picture file 'hmd-overwiew.jpg' will be included in the HTML page. If the size (height and width – because of space saving not denoted in the XML document in the next section) is given, the picture will be scaled accordingly; when clicking on the picture, it will be shown in original size – in another browser tab or window. The 'alt' attributed is used for the figure caption (and appears by 'mouse over').

In order to refer to external learning resources in the internet, hyperlinks are also possible. The syntax looks like this:

```
<content type="hyperlink" hrefDesc="Extremes Beispiel: " tab="5"
         href="https://www.youtube.com/..."
         text="Motion Sickness">
</content>
```

The attribute 'hrefDesc' is the caption of the hyperlink, the attribute 'text' describes the clickable text, when clicking on this text,the specified hyperlink (attribute href') will be opened.
In the generated HTML page, this content description looks like this:


Extremes Beispiel: Motion Sickness (externer Link)

'(externer Link)' symbolizes a hyperlink to a resource outside the actual course ('externer' means 'external'). This hyperlink will be opened in a new browser window. Besides, the attribute 'tab' signs an indenting (for formatting). Hyperlinks to other pages in the current course are also possible. In this case, the hyperlink would refer to a page id (see DTD in section 3.2). Links between different mini courses are not provided: that would presuppose that a course is online available – at a certain location.

Currently, the following media types are supported by the tool:

- text: text written directly in the XML file
- textfile: content of a text file
- image: picture file
- video: video file; with HTML5 a video player can easily be realized
- audio: audio file; with HTML5 an audio player can easily be realized
- hyperlink: link to an external resource in the internet (see above)
  (Internal links to pages inside the course also are possible by using the prefix '#' and the id of the referred page)
- A special content type is 'quiz': This is a hyperlink to webpage generated by an external tool (with hyperlinks to pages of the mini course, where the actual content is described, on the other hand). Quizzes are specified in the XML document by an own element 'quiz' (see section 3.2).

The external media (picture, video, audio) is copied into the target directory of the mini course.
The layout of every page should look similarly. Therefore, we use HTML template files (4 in figure 2; also see section 3.4), which can be adapted to special needs of the teachers by rewriting the HTML code of the standard template or by adjusting the CSS file.

Structure information and meta data are extracted for building a kind of database (6). In our approach, no traditional database management system (DBMS) is used, because the mini course should be used also online, and an installation of a DBMS on one's own device would be too wasteful. Instead, the data are stored as text files or serialized Java classes, respectively.

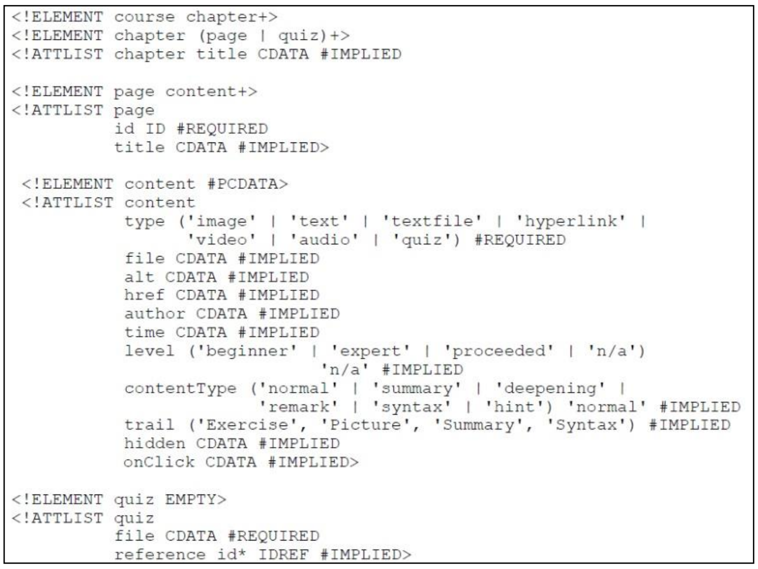
Several parameters of the tool can be adjusted by a configuration file (2 in figure 2; if such a file exists, these values are treated – otherwise default values). Beside the specification of filenames and directories (name of the template document, target directory of the generated files, location of the XML and the template file, etc.), for example – among others – these parameters can be set:

- IMPRESSUM: Imprint (there exits an appropriate placeholder in the template file)
- NUMBERING: Indication, if the chapter and sections should be numbered
- WRITE_AUTOR: Indication, if the authors name and the generation time should be included in the page
- TRAIL: Name of learning trails, which will be generated automatically based on the occurrence of this word in the full text (also see section 3.3)

3.2. XML: Structure and Meta Data

In principle, the structure of a mini course consists of chapters (and subsections if needed) which consist of pages. One single page is composed of assets. Furthermore, hyperlinks to other pages or to external learning sources in the WWW can be integrated in a page. The default navigation consists of 'page up'/ 'page down' or 'chapter up'/ 'chapter down', respectively. In addition, so called learning paths are possible (see subsection 3.3).

The structure, as well as meta data are specified in an XML document. A part of this specification is depicted as an XML-DTD (*XML Document Type Definition*) in figure 3.



```
<!ELEMENT course chapter+>
<!ELEMENT chapter (page | quiz)+>
<!ATTLIST chapter title CDATA #IMPLIED

<!ELEMENT page content+>
<!ATTLIST page
          id ID #REQUIRED
          title CDATA #IMPLIED>

 <!ELEMENT content #PCDATA>
 <!ATTLIST content
           type ('image' | 'text' | 'textfile' | 'hyperlink' |
                 'video' | 'audio' | 'quiz') #REQUIRED
           file CDATA #IMPLIED
           alt CDATA #IMPLIED
           href CDATA #IMPLIED
           author CDATA #IMPLIED
           time CDATA #IMPLIED
           level ('beginner' | 'expert' | 'proceeded' | 'n/a')
                               'n/a' #IMPLIED
           contentType ('normal' | 'summary' | 'deepening' |
                        'remark' | 'syntax' | 'hint') 'normal' #IMPLIED
           trail ('Exercise', 'Picture', 'Summary', 'Syntax') #IMPLIED
           hidden CDATA #IMPLIED
           onClick CDATA #IMPLIED>

<!ELEMENT quiz EMPTY>
<!ATTLIST quiz
          file CDATA #REQUIRED
          reference id* IDREF #IMPLIED>
```

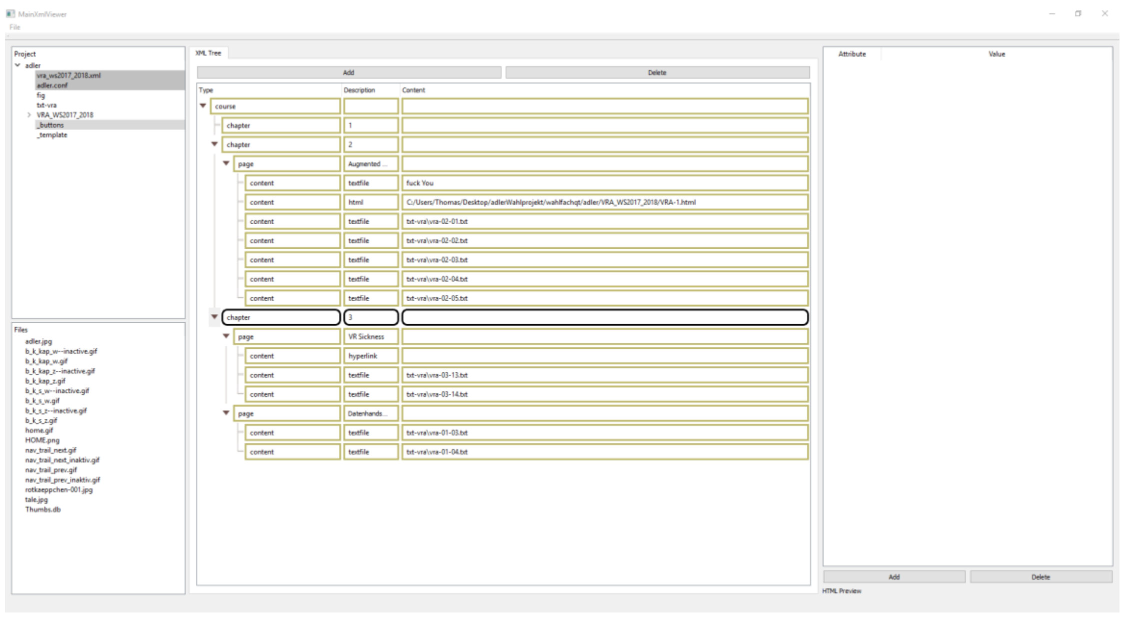
Figure 3: XML-DTD: Structure specification of a mini course (extract)

In this specification, a `course` consists of `chapters`; each chapter consists of subchapters or `page`s (which are transferred to HTML pages). Each page consists of assets (element `content`). For a better understanding, we explain the meaning of some attributes:

- *author*: This is an informative data. It will only appear on the generated HTML page, if the appropriate placeholder exists in the template (see section 3.4 User Interface).
- *level*: This optional attribute indicates the difficulty of the appropriate topic. It is not visible on the page, but can be used for generating individual learning trails (section 3.3) and for the search functionality (section 4).
- *trail*: In the DTD default learning trails are specified. Appropriate statements in the configuration document (see above) can overwrite these.
- *hidden*: This text is hidden for the user, but is accessible for the search functionality (section 4); furthermore, this text can be used in order to generate learning trails (see section 3.3).
- *onClick*: If this attribute exists, the text will appear – with a link to a first hidden text specified by the 'ref' attribute (not listed in the above DTD). In this way, for example, a kind of self-evaluation can be realized: the 'onClick' text shows a question, the answer is hidden and appears only when clicking on the question text.

Even attributes, which are not visible on the generated HTML page, are accessible for the search functionality (see section 4).

With regard to a pleasant use, when building a new course or editing an existing course,students have implemented a graphical editor (figure 4) for the easy-to-use composition of mini courses.



Project
adler
vra_ws2017_2018.xml
adler.conf
fig
txt-vra
VRA_WS2017_2018
_buttons
_template

Files
adler.jpg
b_k_kap_w--inactive.gif
b_k_kap_w.gif
b_k_kap_z--inactive.gif
b_k_kap_z.gif
b_k_s_w--inactive.gif
b_k_s_w.gif
b_k_s_z--inactive.gif
b_k_s_z.gif
home.gif
HOME.png
nav_trail_next.gif
nav_trail_next_inaktiv.gif
nav_trail_prev.gif
nav_trail_prev_inaktiv.gif
rotkaeppchen-001.jpg
tale.jpg
Thumbs.db

Type	Description	Content
course		
chapter	1	
chapter	2	
page	Augmented ...	
content	textfile	fuck You
content	html	C:/Users/Thomas/Desktop/adlerWahlprojekt/wahlfachqt/adler/VRA_WS2017_2018/VRA-1.html
content	textfile	txt-vra\vra-02-01.txt
content	textfile	txt-vra\vra-02-02.txt
content	textfile	txt-vra\vra-02-03.txt
content	textfile	txt-vra\vra-02-04.txt
content	textfile	txt-vra\vra-02-05.txt
chapter	3	
page	VR Sickness	
content	hyperlink	
content	textfile	txt-vra\vra-03-13.txt
content	textfile	txt-vra\vra-03-14.txt
page	Datenhands...	
content	textfile	txt-vra\vra-01-03.txt
content	textfile	txt-vra\vra-01-04.txt

Figure 4: User interface for *AdLeR* (prototype)

After clicking on a button, a mini course will be generated as a sequence of HTML documents – which can be used offline on any device (or online on a server) –at all times and locations.

3.3. Learning paths

The standard navigation path – page by page – serves as a 'guided tour' through the course. This gives the learner a sense of safety. The creator of this course orders and selects appropriate content and media. However, e-learning allows to address the individual needs of the learner. For example, this concerns the selection of media (learning by text, audio, pictures, videos, etc.) as well as the scope and the order of the learning material. With expanding the standard navigation by so-called learning paths, more flexibility is possible during the learning process.

As an example, we consider some use cases (following [16]):

- Modal structure: The students can learn best with pictures, a navigation path could lead through all pages, which contain at least one picture.
- Selective structure: A course about object-oriented programming consists of the introduction of object-oriented programming in general and the concrete realization with the programming language Java. If a learner is familiar with the concepts of object oriented programming, but wishes to learn how this is done with Java, he/ she can skip the path ‘concept’ and can only use a path ’syntax’.
- Repetitive structure: By a path ‘summary’ and/ or ‘exercises’ the learner can prepare oneself for an exam, whereas the navigation path leads through pages, which include summaries or exercises, respectively.
- Didactic structure: By this, it is possibility to react to the heterogeneity of the previous knowledge of the learners, for example offering paths on a beginner level, which includes more details than the standard learning path.

The specification of learning paths in our tool can be done via two ways:

- Automatically: Based on the meta data in the XML document (attributes ‘level’, ‘type’, ‘contentType’, etc.) specific navigation paths will be generated.
- Manually: The attribute ‘trail’ marks the belonging to a specific learning path.

The belonging to learning path(s) is recognizable on every page. This could be helpful for the orientation. On figure 5 an example of learning paths of a mini course about selected themes about augmented reality (AR) and virtual reality (VR) is depicted (meanwhile, the design of the page has been revised; see next section).

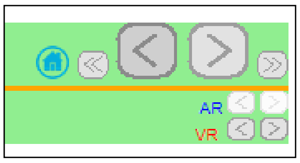


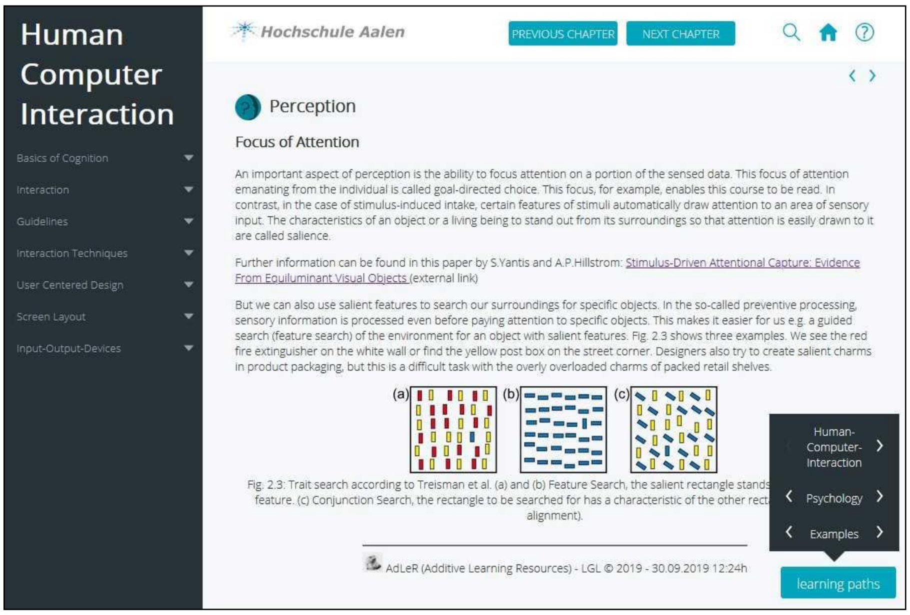
Figure 5: Learning Paths

One can see that this page deals with augmented reality (AR) as well as with virtual reality (VR). This page is the only one, which includes the topic AR (white – inactive – navigation buttons) whereas the topic VR will be treated on the previous and the next page.

3.4. User Interface

In modification to ‘you eat with your eyes’, one can say ‘you learn with your eyes’; that means, the screen design plays an important role at learning on the screen (monitor screen, mobile phone screen, etc.). Thus, we consider the appearance when displaying the learning objects – which also is a student´s project in one of our lessons.

As a student´s project, templates for HTML pages – using HTML5 and CSS3 –were designed and evaluated. The design of the template can easily be adapted by the external CSS3 file to special needs. The template contains placeholders for elements/ functionality, which are filled at the generation process: title, content blocks, navigation, footer text, etc. for example 'navP' for page-navigation, '$navT' for path-navigation ('T' stands for 'trail'), '$foot' for the logo, copyright, and date. Actually, this project is in progress and some drafts have already been evaluated. In the proposal at figure 6, one can see a typical distribution of the single elements: On the left, there is a table of content, at the top there is a chapter up/ chapter down navigation, at the subjacent level a page up/ page down navigation. When clicking on the button at the bottom right, the list of (clickable) available learning paths appears.



Perception

Focus of Attention

An important aspect of perception is the ability to focus attention on a portion of the sensed data. This focus of attention emanating from the individual is called goal-directed choice. This focus, for example, enables this course to be read. In contrast, in the case of stimulus-induced intake, certain features of stimuli automatically draw attention to an area of sensory input. The characteristics of an object or a living being to stand out from its surroundings so that attention is easily drawn to it are called salience.

Further information can be found in this paper by S.Yantis and A.P.Hillstrom: Stimulus-Driven Attentional Capture: Evidence From Equiluminant Visual Objects (external link)

But we can also use salient features to search our surroundings for specific objects. In the so-called preventive processing, sensory information is processed even before paying attention to specific objects. This makes it easier for us e.g. a guided search (feature search) of the environment for an object with salient features. Fig. 2.3 shows three examples. We see the red fire extinguisher on the white wall or find the yellow post box on the street corner. Designers also try to create salient charms in product packaging, but this is a difficult task with the overly overloaded charms of packed retail shelves.

Fig. 2.3: Trait search according to Treisman et al. (a) and (b) Feature Search, the salient rectangle stands feature. (c) Conjunction Search, the rectangle to be searched for has a characteristic of the other rect alignment).

AdLeR (Additive Learning Resources) - LGL © 2019 - 30.09.2019 12:24h

Figure 6: Layout for an HTML page of the mini course

In addition, designing the user interface for our application was done considering the results of the human-computer-interaction research (a student´s project as well).

4. Search Functionality

Even though the mini courses are mostly small, it could be useful to offer a search functionality. For example, to decide, if a mini course is meaningful for the current need, and to find easily the actual relevant topics which are part of the mini course. In our context, we have interesting possibilities: The underlying XML document offers rich structure and metadata information, which can be exploited for a powerful search functionality. In addition, because an XML structure bases on a (simplified) formal grammar, aspects of the theoretical computer science can be integrated in the data modeling (for the search function) and offers an alternative application of formal languages to the students –in contrast to the traditional dealing with this topic in lessons. Thus, we can present a flexible search functionality, which goes much further than a simple full text search.

It is possible to search for all attributes, which are specified in the XML document: Apart from *title*, *date*, *size*…, one can also look for prerequisites, etc.

In addition, structure information is available: the hierarchy of the XML elements and the linking between elements. The structure (hierarchically order, sequences) exists implicitly in the underlying document(XML document). There is a semi structured data view to the data, which has already been described in [1], adapted in [9].

This idea allows some structure-oriented queries, for example by using:

Structure-describing attributes: The (recursive) structure of the learning object can be described by object-valued attributes. For example, the titles of all objects of a hierarchy can be described as a character string like o6.title$_{path}$='Databases/Languages/SQL'. In this manner, also regular expressions can be used, for example in order to look for all subsections with the title 'Data Models' or 'SQL': */('Data Models'|'SQL')/*'.

Derived attributes: The theoretical concept of attributed context free grammars (for example used at compiler construction), can be adapted to hierarchically structured data: The deriving (inferring) of attributes and attribute values of the XML elements. This can be done alongside the hierarchy (parent/ children relationship) or alongside the links or paths.

As an example, consider a part of a lesson about extended reality (XR) (see figure 7), which is organized in a hierarchical structure (see figure 7, left side): At the top, the title of the chapter of the lesson is described (title='XR'). This chapter consists of three subsections, each with a subtitle and the estimated time to complete this topic (*dur*: duration). The duration can be inherited upwards, whereby the sum of all times is calculated, so the top element (chapter) offers the total estimated time to complete the topic 'XR'. On the other hand, the title of the top element can be inherited to the subjacent lessons downwards. In figure 7, the derived attributes are written in *italic* (based on [10]).

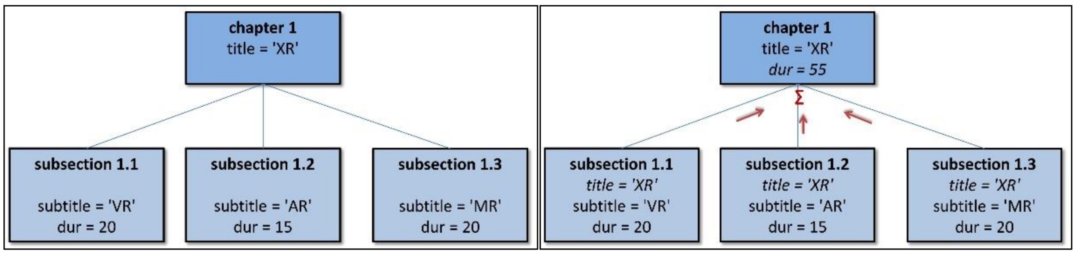


Figure 7: Inheriting of attributes and attribute values

This means, searching for objects with the title 'XR' results not only in the top element (chapter with the title 'XR'), but also in the dependent lesson objects 'VR', 'AR' and 'MR'.

In order to distinct origin and derived attributes, the derived values can be weighted when performing the query processing. This will be used to rank the search results.

In addition, pages, which are connected by learning paths, can be considered as a hierarchical structure. The linked pages will be decreasingly weighted.

Attributed context free grammar approach

In order to process such structure-oriented queries, we need an adequate internal representation of the data. The underlying data model can be described like an attributed context free grammar (foundations see [15], prior working concerning data modeling see [19]). Referring to the above example, we extend the specification as follows:

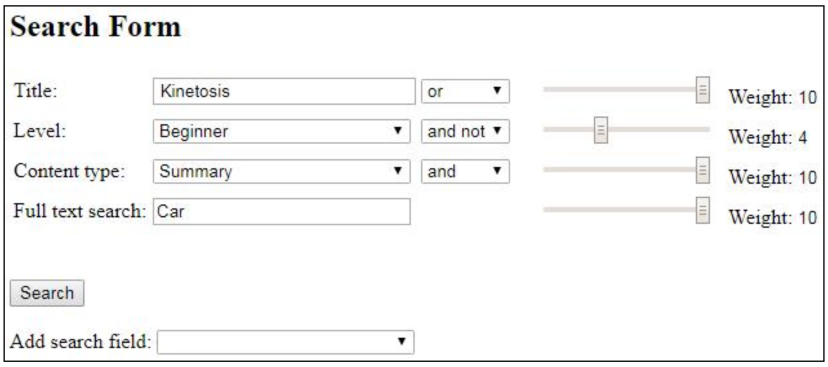
- A course consists optionally of slides (described by a heading) and of at least one chapter(described by a title).
- A chapter consists of at least one subsection (described by a title and a duration - dur)
- A subsection consists of inner subsections (optional) or at least one page.
- A page represents the generated (HTML) page and consists of at least one asset (text, picture, etc.)

As an addition to the actual course, slides are possible, for example, in order to present some topics for classroom teaching. An attributed context free grammar for this scenario can look like this (highly simplified; words with capital letters are non-terminal symbols, words with lowercase letters are terminal symbols, words with italic letters are attributes):

COURSE	→	SLIDE*CHAPTER*
		title := SLIDE.*heading*∪ (∪ CHAPTER.*title*)
SLIDE	→	asset
CHAPTER	→	SUBSECTION+
		dur := ∑(SUBSECTION.*dur*)
		title := ∪(SUBSECTION.*subtitle*)
SUBSECTION	→	SUBSECTION* |PAGE+
PAGE	→	CONTENT+
CONTENT	→	asset

Remark: The XML-DTD above (figure 4, as well as the following example in figure 8) does not include subsection elements: it uses a simplified specification, which in many cases is sufficient. For more complex learning units, the XML-DTD can easily be customized; the generator tool ‘AdLeR’ is prepared to deal with subsections.

This data model allows expressive possibilities of the search functionality. The query processing itself is invisible (deriving/ inheriting and calculation of attributes and attribute values), but some extended parameters of the search process can be controlled by user input directly using an appropriate search form like in figure 8.



Search Form

Title:	Kinetosis	or	Weight: 10
Level:	Beginner	and not	Weight: 4
Content type:	Summary	and	Weight: 10
Full text search:	Car		Weight: 10

Add search field:

Figure 8: Search form

The data are extracted from the XML document and are stored as a file, because the mini course is used for offline learning, so that an online database system is not appropriate.

5. CONCLUSION AND FURTHER WORK

5.1. Summary

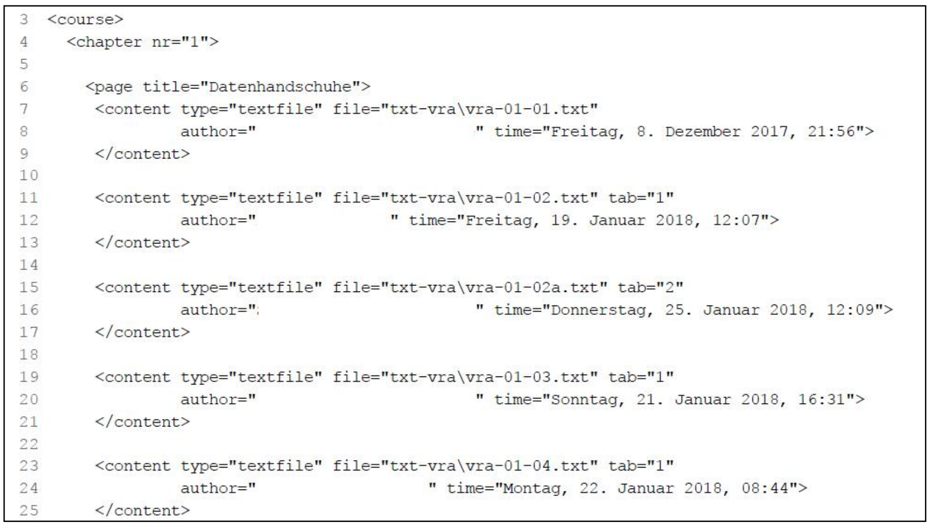
Considering the observed gaps in previous learning, we offer mini courses in the sense of microteaching, so that students can catch up on or repeat the required knowledge.

Undergraduate Students can benefit from advanced students, who might explain the learning matter in another way than the teacher by using mini courses for self-regulated learning. Advanced students themselves benefit from a better understanding of the learning matter by putting the theory of the visited lessons into practice (*learning by teaching*).

Concretely, students are involved in the development of our tool in the following part disciplines of computer science:

- Software engineering/ programming: The students have to understand existing software, and have to specify and implement new features – including testing and debugging.
- Human-computer-interaction: The learned concepts of user experience have to be adapted to the screen design of the generated mini courses, as well as to the design of the user interface for the generation tool *AdLeR*.
- Theoretical computer science & database theory: The concept of formal languages (theoretical computer science) has to be adapted to the data modelling using context-free grammars inclusive ranking and query processing (database theory).

The mentioned benefit for undergraduate students (learning missed teaching content) can be adapted for advanced students as well: Every now and then interesting topics or deeper analysis cannot be handled in the lessons because of the tight schedule. Mini courses can fill this gap. For example, in our lecture ‘Virtual Reality and Animation’, the students picked out interesting topics. The results were combined to a mini course. The appropriate XML file is depicted in figure 9 (reduced and extracted; the names of the students are disguised).



```
<course>
  <chapter nr="1">

    <page title="Datenhandschuhe">
     <content type="textfile" file="txt-vra\vra-01-01.txt"
              author="                      " time="Freitag, 8. Dezember 2017, 21:56">
     </content>

     <content type="textfile" file="txt-vra\vra-01-02.txt" tab="1"
              author="             " time="Freitag, 19. Januar 2018, 12:07">
     </content>

     <content type="textfile" file="txt-vra\vra-01-02a.txt" tab="2"
              author=";                     " time="Donnerstag, 25. Januar 2018, 12:09">
     </content>

     <content type="textfile" file="txt-vra\vra-01-03.txt" tab="1"
              author="                      " time="Sonntag, 21. Januar 2018, 16:31">
     </content>

     <content type="textfile" file="txt-vra\vra-01-04.txt" tab="1"
              author="                 " time="Montag, 22. Januar 2018, 08:44">
     </content>
```

Figure 9: XML document for a mini course (extract)

5.2. Future Work

Our future steps are divided into two parts: the tool itself and using the tool.

First, we will evaluate and improve the user interface of the generation tool and we will improve the implementation of the search functionality and other features (graphical sitemap, etc.). In addition, an HTML preview is planned.

Because the particular learning objects (assets) are separated from the kind of presentation, a multimodal publication of the learning matter is possible: besides HTML web pages, the output could be a PDF file or slides for a presence presentation. XML tags can mark this. When considering a PDF output, the intermediate step LaTeX is also interesting: The XML structure is translated into LaTeX, which can easily be transformed to a PDF file; furthermore, the LaTeX document can be adapted to the special needs after compilation of the mini course.

Furthermore, we observe an increased use of mobile phones for (self-regulated) learning. Thus, we intend to use augmented reality learning apps like the use of so-called 'Learning Factories', in which the reality is augmented with additional information [12]. One-step further is the visualization of non-visible processes or abstract issues, which could improve the motivation and the learning success [20].

Secondly, we will motivate students to create new assets and reuse existing learning material in order to generate more mini courses. This brings the students to recapitulate their knowledge and force them to structure the learning subjects. The advanced students are nearer to undergraduate students, they know about their own problems and are able to present the learning matters in another way as the teachers. The tool AdLeR allows concentrating on the intended learning goal, the transfer to a course is done automatically. An intensive evaluation is planned in order to confirm or disprove the effects to the learning students – regarding the comprehension (found by tests) and the motivation (found by questionnaires).

ACKNOWLEDGEMENTS

The program ‘Hochschuldidaktische Projekte’ (Baden-Wuerttemberg, Germany, 2017) funded previous parts of this work. The authors would like to thank the students Paula Schwenger, Florian Russl, Ufuk Karaali, and Thomas Dennighoff for the implementation of parts of the tool ‘AdLeR’.

REFERENCES

[1] Abiteboul,Serge (1999) “On Views and XML”. Proc. of ACM Symposium on Principles of Database Systems, pp 1-9. 1999

[2] Allen,Dwight William & Cooper,James Michael & Poliakoff,Lorraine (1972) “Microteaching”, U.S. Department of Health, Education, and Welfare, Office of Education, National Center for Educational Communication. 1972

[3] Aslan,Safiye (2015) “Is Learning by Teaching Effective in Gaining 21st Century Skills? The Views of Pre-Service Science Teacher”, Educational Sciences: Theory and Practice, 2015

[4] Bergmann, Jonathan & Sam,Aaron (2012) „Flip your classroom – Reach every Student in Every Class Every Day”, International Society for Teaching in Education, 2012

[5] Boekarts,Monique (1999) “Self-regulated learning: where we are today”, International Journal of Educational Research, Volume 31, Issue 6, pages 445-457, 1999

[6] Campos-Sánchez, Antonio &del Carmen Sánchez-Quevedo,María &Crespo-Ferrer,Pascual Vicente & García-López,José Manuel & Alaminos,Miguel (2013) “Microteaching as a self-learning tool. Students' perceptions in the preparation and exposition of a microlesson in a tissue engineering course”, Journal of Technology and Science Education, Vol. 3, No. 2, 2013

[7] Garrision,D. Randy & Heather,Kanuka (2004) “Blended Learning: Uncovering its transformative potential in higher education”, The Internet and Higher Education. Volume 7, Issue 2, pp 95-105. 2004

[8] Hug, Theo (Ed) (2007) “Didactics of Microlearning”, Waxmann. 2007

[9] Lecon, Carsten & Seehusen,Silke (2002) “Combining Structure Search and Content Search for Online Courses”, 13th International Workshop on Database and Expert Systems Applications (DEXA), 2-6 September 2002, Aix-en-Provence (France)

[10] Lecon,Carsten &Hermann,Marc (2019) “Flexible Search Function for Online Courses in the Sense of Attribute Grammars”,15th International Conference on Information Technology & Computer Science. 20-23 May 2019, Athens (Greece)

[10] Lueckemeyer,Gero (2015) “Virtual blended learning enriched by gamification and social aspects in programming education”, 10th International Conference on Computer Science & Education (ICCSE), 22-24 July 2015, Cambridge (UK)

[11] Martín-Gutiérrez,Jorge & Mora,Carlos Efrén & Añorbe-Díaz,Beatriz & González-Marrero.Antonio (2017) “Virtual Technologies Trends in Education", EURASIA Journal of Mathematics Science and Technology Education, 2017 13(2):469-486

[12] Mueller-Frommeyer,Lena C. et. al. (2017) “Introducing competency models as a tool for holistic competency development in learning factories: Challenges, example and future application”, 7th Conference on Learning Factories (CLF 2017), 2017

[13] Pintrich,Paul R.& V. de Groot, Elisabeth (1990) “Motivational and self-regulated learning components of classroom academic performance”, Journal of Educational Psychology, 82(1), pp 33-40. 1990

[14] Pintrich,Paul R. (2000) “The Role of Goal Orientation in Self-Regulated Learning”, Handbook of Self-Regulation, chapter 14, Academic Press, pages 451-502, 2000

[15] Reghizi, Stefano Crespi & Breveglieri, Luca & Morzenti, Angelo (2019) „Formal Languages and Compilation“, 3rd edition. Text in Computer Sciences, Springer, 2019

[16] Seehusen, Silke & Lecon, Carsten & Kaben, Cay (2000)„Specification of learning trails in virtual courses”, Frontiers in Education Conference. FIE 2000, Vol. 2. 2000

[17] Stansburry,Meris (2010) “Teachers turn up learning upside down”, eSchool News. December 22nd 2010. https://www.eschoolnews.com/2010/12/22/teachers-turn-learning-upside-down/?all (read on 02/16/2020)

[18] Stollhans. Sascha (2016) “Learning by teaching: developing transferable skills”, in E. Corrandi & K. Borthwick & A. Gallagher-Brett (eds), Employability for languages: a handbook (pp 161-164), Dublin: Research-publishing.net. 2016

[19] Stuschka, Ulrike & Linnemann,Volker(1995) “Attributierte Grammatiken als Werkzeug zur Datenmodellierung“. Datenbanken in Büro, Technik und Wissenschaft (BTW), Dresden (Germany), 22-24 March 1995 (in German)

[20] Yuen,Steve Chi-Yin & Yaoyuneyong,Gallayanee & Johnson,Eric (2011) “Augmented Reality. An Overview and Five Directions for AR in Education”, Journal of Educational Technology Development and Exchange (JETDE), Vol. 4, Iss. 1, Article 11, 2011

AUTHORS

Prof. Dr. Carsten Lecon

Short Bio

- Study of computer science
  (Technical University Braunschweig, Germany)
- Software Quality Assurance
  (Siemens AG, Braunschweig)
- Database systems, Media archives
  (University Luebeck, Germany)
- Virtual University of Applied Sciences
  (FH Luebeck, Germany)
- Since 04/2004 Professor for media computer science
  (Aalen University for Applied Sciences, Germany)

Dr. Marc Hermann

Short Bio

- Study of Computer Science
  (Ulm University, Germany)
- Certificate of Higher Education Pedagogy
  (Baden-Württemberg Certificate)
- Software Developer
  (Inneo Solutions GmbH, Germany)
- Senior Developer
  (Veroo Consulting GmbH, Germany)
- Since 04/2009 Lecturer for several courses
  like C, C++ and Java Programming, Software Engineering, Human Computer Interaction
  (Aalen University for Applied Sciences, Germany)