WEB Based Lecture Notes - The LENO Approach

Kurt Nørmark
Department of Computer Science
Aalborg University
Denmark
normark@cs.auc.dk

ABSTRACT

A system for production of WEB based teaching material is described. The system, called LENO, takes its starting point in the production of slides, but in its current form it covers a broader need. We argue that a dedicated lecture notes system for the WEB is preferable to a more general presentation tools, such as Powerpoint. LENO is based on single sourcing ideas in order to avoid duplication of source materials. LENO supports a trail facility for selection of a subset of slides from a number of lectures, an automatically played sound track related to the slides, and a mechanism for inclusion of selected parts of external text files, such as source programs.

Keywords

Lecture notes, WEB, HTML, LENO

1. INTRODUCTION AND MOTIVATION

The World Wide Web has pervaded the area of computer science education. Today, it would be an almost impossible thought to give a university course in the CS area without arranging a WEB page with all the practical informations pertaining to the course. Besides such *course home pages* a variety of other WEB-based materials play a significant role in almost any course and project in the CS curriculum.

In this paper we will discuss WEB based lecture notes and teaching materials, beyond plain course home pages. We will in particular describe a concrete system—LENO—for the production of such materials, which has been in use at Aalborg University the last four years.

We will use the term *lecture notes* for a collection of teaching material which is centered around the substance presented during a lecture. Slides and overheads are popular possibilities, but other forms may also be used. In the one extreme, the lecture notes may be limited to a reproduction of some slides. Typically, the lecture notes contain additional material organized in relation to the slides, for instance exercises, quizzes, FAQ's, and other kinds of annotations. In the other extreme the lecture notes may be developed to subsume material in amounts comparable to traditional text books, or beyond.

Lecture notes have existed for decades in paper form. Today, it is attractive to publish such material in electronic form, in part to accommodate the needs for frequent changes of the material, and in part to make it available to students—both on campus and off campus—in a cheap and easy fashion. In this context it should be noticed that paper editions of lecture note material is still useful and popular, at least as a supplementary means to the electronic edition.

There exists very popular software to produce slide material, not least Powerpoint from Microsoft. Such tools can produce WEB material too, but the concepts behind the tools were invented before the age of the WWW, and therefore the WWW integration and the degree of utilization of the WWW potential is not perfect.

In our work on LENO and LENO-based teaching materials we have strived for production of lecture note materials using the native form of the WEB, namely pure and standard HTML. In that way it is possible to achieve an ideal, non-complicated integration with existing WEB material, such as the before mentioned 'course home pages' and the wealth of WEB resources on servers around the world. In particular, it is possible to link the lecture material with existing resources in a straightforward manner. In addition, the material can be utilized right away in every browser on every platform, without initial preparations, loading of plugin, long waiting times, complicated registration procedures, and similar frustrations.

The WEB-based approach shall be seen in contrast to approaches based on proprietary formats of commercial vendors, such as the various Powerpoint players, PDF, and others, which all give a more closed and monolithic impression of the material, at least seen in relation to 'the rest of the WEB'. Admittedly, some of the commercial vendors provides glamorous facilities which give the audience a professional impression, and which somehow set the trend in the area. In addition, these tools offer user friendly authoring facilities - not least for the casual user. We have found, however, that many of the visual effects, including color art, decorations, and many animations, do not necessarily make

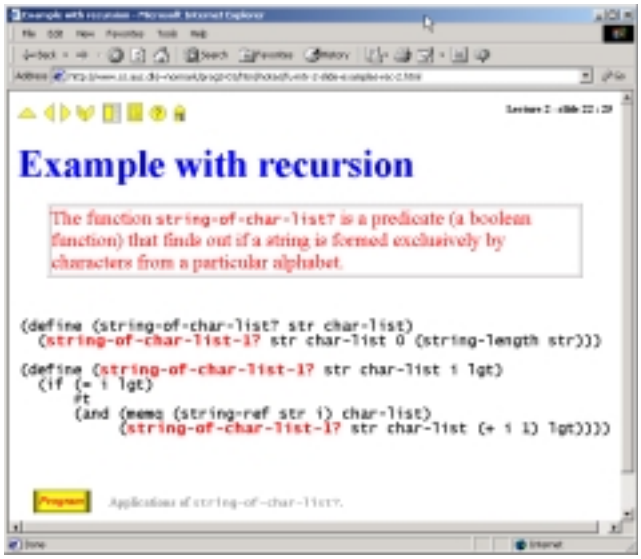

Figure 1: *An example of a note page shown in slide view.*

a difference in CS university courses. Moreover, the authors of more substantial materials are engaged in complex endevours (such as inclusion of many pieces of external materials and production of several editions of the material) which are not well-supported by the 'easy to use' visual interfaces of Powerpoint and similar systems.

In the following section we will discuss the features of LENO with emphasis of the pros and cons for both the students and the author of the lecture notes.

2. LENO AND LENO-BASED MATERIALS

As discussed above, the lecture notes made by LENO is structured with outset in the slides, which are presented during the lectures. The slides produced by LENO represent a particular high-level view of the material, rendered in such a way that it can be shown in an auditorium (large font and relatively few words) using a computer running a WEB browser, connected to a projector. Figure 1 shows an example of a LENO slide.

The central concept of LENO is the `note-page` clause, which contains a number of subclauses such as `title`, `item`, `image`, `source-program`, and others. The `note-page` and its subclauses contain the information of the slides as well as a number textual annotations, which are utilized in the other views supported by LENO. Figure 2 gives an overview of the current means of expressions of LENO (the subclauses of a `note-page` clause).

Besides the slide view, LENO also supports the annotated slide view and the aggregated lecture note view. In the *annotated slide view* the slide contents is shown side-by-side with annotations to the slide contents. The *aggregated lecture note view* presents all slide information and all annotations of a lecture on a single aggregated page. Figure 3 illustrates the annotated slide view, and part of an aggregated lecture view is shown in figure 4.

LENO operates by processing a single description of a lecture, using a special form of programmatic markup (to be discussed in section 2.4). As such, we use a single sourcing approach [3]. Each of the different views supported by LENO stems from common information in the LENO source description.

At a more basic level, LENO can be understood as a synthesizer which aggregates a large number of source pieces (text, graphics, icons, and sound clips) to an even larger number of HTML pages with references to the necessary media components. As one of the main tasks, LENO facilitates a systematic linking of these constituents to a coherent WEB. Any attempt to handle this linking effort in a manual way would fail, due to the large number of pieces involved.

2.1 Inclusion of source programs

In many CS contexts, it is useful and typical to include pieces of programs in the lecture note material. In order to avoid duplication of this kind of source materials, LENO is able to include external source programs in the resulting teaching material. The inclusion steps involve:

1. Addressing the program source file.
2. Selecting an appropriate part of the source file for inclusion in the lecture note material.
3. Superimposing colors and fonts to emphasize particular aspects of the program.

title	The title of the note page
section-title	A specialize title which marks a new section
point	A major idea expressed in brief form
items	Itemized text with subitems
opposing	A list of opposing items
text	A textual contribution
note-text	Textual contribution to the annotations only
slide-text	Textual contribution to the slide view only
image	A piece of graphics
image-series	A simple animation in an image series
slide-image	An image contribution to the slide view only
concept-list	A list of concept definitions
source-program	An source program included from a file
elucidator-program-fragment	A source-program prepared by an elucidator
example	An example
syntax	A syntactical form
tabular	A tabular exposition
cross-references	References to other materials
exercise	An exercise with formulation and solution
index-words	Contribution to the word index
applet-program	A Java applet
synopsis	A synopsis clause for summary purposes
quotation	A quote
quiz	A contribution to a multiple choice quiz
show-and-speak	A specification of sound clips on a note page
lecturer-photos	A specification of photos on a note page
lecturer-photos-and-logo	A variant of lecturer-photo with a logo
elucidate	A link to an elucidated program
comment	An internal comment
slide-space	Specification of extra slide whitespace

Figure 2: *The possible subclauses of a* `note-page` *in LENO.*

Figure 3: *An example of the annotated slide view of a note page. The left part of the window is identical to slide view shown in figure 1. The right part contains the annotations.*

The following clause show an example of a `source-program` clause for addressing, selecting, and decorating a source program:[1]

```
(source-program
 'src "sources/prog.java"
 'from-mark "class Account{"
 'to-mark "// end Account"
 (color-decorations
  (color-decoration 'from-mark "Account"
    'to-mark "" 'color "red" 'face "bold")
  (color-decoration 'from-mark "balance"
    'to-mark ";" 'color "blue" 'face "bold")
 )
```

[1]The source-program clause is shown in the surface syntax we call *XML-in-LAML*, which is an XML notation at the syntactical level, using Lisp S-expressions at the lexical level.

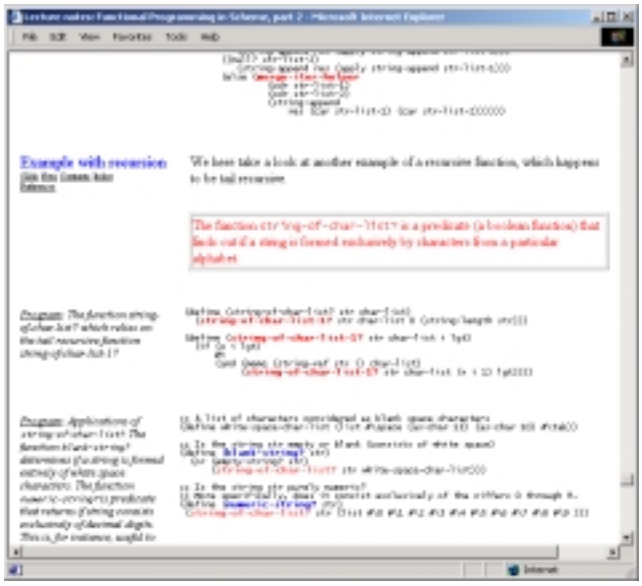

Figure 4: *An example of the aggregated lecture view, corresponding to figure 1 and 3. Both source programs are unfolded, and the annotations are shown as margin comments.*

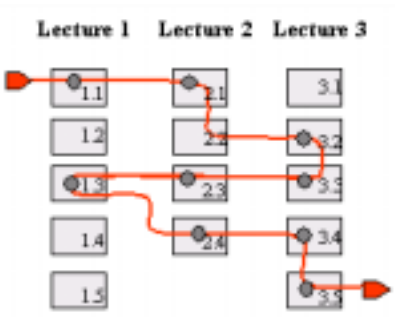

Figure 5: *A trail of note pages which includes material from page 1 of lecture 1, page 1 and 2 of lecture 2, page 2 and 3 of lecture 3 etc.*

```
  (main-text "A simple Bank account class")
  (annotation "Notice the instance variable.")
  'slide-mode "inline"
  'book-mode "inline"
 )
```

As an alternative to the LENO approach, it would be necessary for the author to copy part of the source program to the lecture note description. This protects and shields the lecture note material from accidental changes in the source program. On the other hand, it is likely that we always want to take advantage of the latest version of the source program in the lecture note material (to be sure that the lecture note version of the program is free of errors). Therefore we have come to the conclusion that automatic inclusion of external source programs upon each processing of a LENO lecture is the best of the two alternatives.

2.2 Lecture note versioning

A given part of a teaching material is often used in slightly different contexts. As another consequence of the single sourcing approach used in LENO, we strive to avoid duplication of source material to accommodate such variations of the notes.

In some situations only a selected part of the material is necessary. As a typical example at Aalborg University, open university seminars cover selected aspects of a course, which are covered in a number of regular lectures. In order to avoid a proliferation of different versions of the teaching material—reflecting different blends and selections—LENO supports the definition of *trails*. A trail specifies an ordered selection of note pages from the lectures in a batch of notes, cf. figure 5. The trail defines a new lecture, orthogonal to the basic structuring of the material. A trail page is implemented in term of a frameset, which addresses

1. one of the original lecture pages
2. a tiny, trail-specific navigation banner

Figure 6 shows an example of a trail page made by LENO. The vertical strip to the left shows the trail navigation frame.

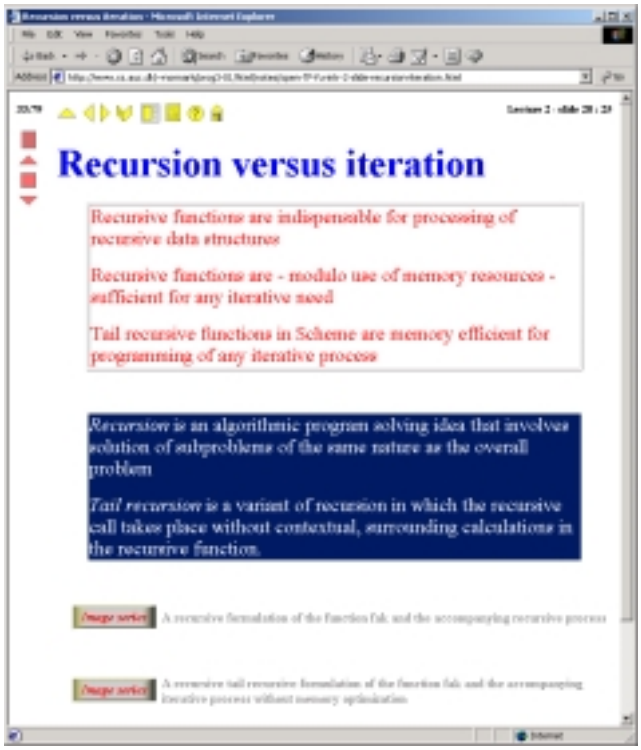

Figure 6: *A LENO trail page with a trail specific navigation bar in the left hand side of the window.*

As a somewhat unexpected advantage of embedding slide note pages in trail framesets, it is always easy to 'come back on track' after having followed a number of links. This is a convenient alternative to going back a number of pages, using the 'Back' button of the browser. In situations where many 'tangential side leaps' can be foreseen, LENO can support this by generation of a pseudo trail around a conventional lecture.

Another kind of versioning reflects different organizations of the lecture notes, for instance an organization on a given WWW server, an organization on a CD, and an organization on the teachers computer. Such organizations are typically distinguished by different inter-linkings of major parts of the material. As examples, links in one versions go to a WEB resource via absolute links, and in the CD edition these links address local CD material by means of relative links. The programmatic approach to authoring used in LENO supports *conditional linking* in a straightforward way, see section 2.4. In addition, LENO provides hooks such that the rendering of cross references can be associated with tiny icons that give hints to the reader about the targets of the links ('on CD', 'Remote', 'Local server', etc).

2.3 Sound tracks

Beside the textual annotations of the note page clauses, LENO also supports audio annotations, in terms of pre-recorded sound fragments, typically used for the lecturer's discussion of the material. When started, LENO will play the sound track and automatically advance through the slides until the end of the presentation. Still-images of the lecturer can be added to selected note pages in order to visualize who speaks. In the simple case, there is a one-to-one mapping between note pages and sound clips. However, LENO also supports sound clips related to programs, exercises, and image series which calls for multiple encounters of a single note page. In this way, the lecturer can discuss a note page, dive into a number of details, and come back to the note page, for instance to summarize the most important points.

When located on a WEB server, the sound can be addressed on a streaming server, using Real Audio (TM) technology. On low bandwidth connections, this provides for almost momentarily start of the sound clips—independent of their sizes (which is often more than one megabyte). On disk or CD editions of the material the sound may be taken from simple sound files on the disk/CD.

We compare the sound track facility of LENO with streaming video of real lectures, as given and recorded in an auditorium. The CS department at Aalborg University is involved in an experiment where a number of courses are recorded on a regular basis [1]. Using the video medium, the picture, the voice, and the presentations of the slides and the blackboard can be transmitted in the video signal. It is, however, difficult to capture slide and blackboard details in sufficiently good quality. To alleviate this problem, markers in the video signal may control an accompanying slide presentation in a separate browser frame. The value of transmitting the picture of the lecturer can also be questioned, but we find the voice of the lecturer to be absolutely central.

Using LENO, a well-prepared sound track can be organized relative to the note pages (see [9] for an example). As a consequence, it is easy to index the lecturer's spoken contributions, via the already existing overview page of a lecture, as generated by LENO. As another advantage, the total length of a well-prepared sound track tends to be shorter than an regular lecture, because a number of less important aspects can be left out. On the down side, a studio recorded lecture is less vivid than a recording of a real lecture 'in front of a live audience'. At the Computer Science Department of Aalborg University we are still exploring pros and cons of the two different approaches.

2.4 Programmatic authoring

The source description of a LENO lecture is a program written in the programming language Scheme [2], using the LAML libraries [5].

As part of the work on LENO, we have developed the discipline of *programmatic authoring* of WEB materials [10]. The key idea of programmatic authoring is to mirror the markup language (HTML) in a programming language. Using the Scheme mirror, the expressiveness of HTML is made available through functions in Scheme. On this ground, it is possible to use the principles of abstraction to form domain specific language for a variety of WEB purposes. LENO is one such language. In addition, it is possible to automate a number of routine tasks directly in the LENO document, which otherwise would have required manual work to carry out. As an example, we can check the target of a number of cross references, and we can check that file names do not get longer than the maximum allowed on a CD file system.

```
(define (jdk-doc-link final-path)
 (cond ((eq? java-doc-where 'cs-unix)
          (course-relative (string-append "vejledninger/" "jdk-docs/" final-path)))
       ((eq? java-doc-where 'home-pc)
          (string-append "e:/jdk1.2.2/docs/" final-path))
       ((eq? java-doc-where 'lab-top)
          (string-append "c:/jdk1.2.2/docs/" final-path))
       ((eq? java-doc-where 'oop-cdrom)
          (string-append "../java-stuff/jdk1.2/docs/" final-path))
       (else (laml-error "java-doc-link: Unknown location."))))
```

Figure 7: *A function that returns a link, which depends on the value of the constant* `java-doc-where`.

We have used the power of programmatic WEB authoring to support *conditional linking*. As mentioned in section 2.2, different versions of a teaching material is often characterized by different linking of the material. Systematic redefinition of links is easy to support if the links are generated by a function. An example of such a function is shown in figure 7. The function is taken from the implementation of the lecture notes of object-oriented programming [9].

In general, it is valuable to avoid multiple occurrences of long URLs with identical prefix parts in a document. It is better to define a function which holds the common constituents of the URLs. In that way, it is much easier to maintain the document in case the source material, or material on which it depends, is reorganized.

2.5 Integration with other tools

LENO is able to integrate a teaching material with other WEB-based services. At the time of this writing, the following connections between LENO and other WEB-based LAML tools exist:

- Use of a synchronous exercise manager in order to facilitate immediate help to students in an exercise situation [7].
- Use of an asynchronous distance education tool - IDA-FUS - to handle the dialogue between the students and the teacher with respect to exercises (including the ability to release a model solution in return to the student's solution).
- Definition of lecture specific quizzes, based on the LENO quiz subclause and a sever-side quiz checker.
- Use of an annotation facility which allows the students to comment on individual pages, exercises, and solutions in the material.
- Inclusion of source program fragments from the Java Elucidator [11], which short circuits step 2 and 3 of the inclusion steps for source programs in LENO (see section 2.1 above).

The integration of LENO with other tools adds considerably to the 'power' of the teaching material, not least due to the integration with the interactive services which allows various kind of dialogues between the readers and the authors. As the downside of the integration, it contributes with a considerable complexity on both the software side and the management of the material.

3. STATUS AND CONCLUSION

We have described LENO, which is a system for production of WEB-based teaching materials. The author has used LENO consistently during several years for production of teaching materials [8, 9] and accompanying slides to talks, such as the presentation [6] accompanying [7]. In this paper we have described the most interesting features of the well-established LENO system. LENO is, however, a system in active development, meaning that a number of new properties are currently being introduced. These new features are:

- **XML support.** A new surface version of the LENO markup language using the *XML-in-LAML syntax* has been made recently. The `source-program` example in section 2.1 illustrates this syntax. With this syntax, XML can be used to produce simple materials (via transformation of XML documents to the XML-in-LAML format). More complex materials will still have to be produced on LAML ground, due to the programmatic power of Scheme available at this level.
- **Style sheet support.** Recently, the slide view of LENO has been enriched by use of CSS. This allows the author a much better control of the appearance of the material. Furthermore, the presentation style can be changed at a late point in time, simply by addressing another stylesheet.
- **Support of themes.** The original LENO system has been centered around annotated slides. A *LENO theme* represents a more overall perspective on a teaching material. A LENO theme makes use of separate textual contributions, but it can also address individual clauses of note pages—as the 'bread and butter' of the theme. In that way, the author can use LENO to produce more coherent teaching materials, which make good use of existing lecture notes.

In conclusion, we find it important and interesting to produce complex WEB based teaching materials on the ground of standard HTML and CSS. The sound track feature is the only deviation from this course, because it needs the RealPlayer (TM) plug in. The LENO system allows the creation of such materials by means of a programmatic authoring approach.

LENO is available as free software, and as part of the LAML package, from the LAML home page [5].

4. REFERENCES

[1] Digital video service. http://video.auc.dk. The Faculty of Engineering and Science at Aalborg University.

[2] Richard Kelsey, William Clinger, and Jonathan Rees (editors). Revised^5 report on the algorithmic language Scheme. *Higher-Order and Symbolic Computation*, 11(1):7–105, 1998.

[3] Pamela Kostur. Information modeling for single sourcing. In *18th Annual Conference on Computer Documentation - IPCC, SIGDOC 2000*, pages 333–342. ACM and IEEE, 2000.

[4] Kurt Nørmark. The Elucidative Programming Home Page. `http://www.cs.auc.dk/~normark/-elucidative-programming/`, 1999.

[5] Kurt Nørmark. The LAML home page. `http://www.cs.auc.dk/~normark/laml/`, 1999.

[6] Kurt Nørmark. A slide presentation of [7]. Available at `http://www.cs.auc.dk/~normark/scheme/-slides/iticse/html/iticse.html`, July 2000.

[7] Kurt Nørmark. A suite of WWW-based tools for advanced course management. In *Proceedings of the 5ht annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education*, pages 65–68. ACM Press, July 2000. Also available from `http://www.cs.auc.dk/~normark/laml/`.

[8] Kurt Nørmark. Functional programming in Scheme—a WWW-oriented approach. WEB material available at `http://www.cs.auc.dk/~normark/prog3-01/html/-notes/`, 2001.

[9] Kurt Nørmark. Lecture notes in object-oriented programming (in danish). WEB material available at `http://www.cs.auc.dk/~normark/prog1-01/html/-noter/index.html`, 2001.

[10] Kurt Nørmark. Programmatic WWW authoring using Scheme and LAML. Submitted for publication, November 2001. Also available via [5].

[11] Kurt Nørmark and Thomas Vestdam. Elucidative programming in computer science education. Submitted for publication, November 2001. Also available via [4].