Dynamic Teaching Materials for ESSLLI

R. Bernardi[1] **I. Dahn**[2] **G. Mishne**[3] **M. Moortgat**[4] **M. de Rijke**[3] **H. Uszkoreit**[5]

[1] Free Unversity Bolzano Bozen
[2] University Koblenz-Landau
[3] Informatics Institute, University of Amsterdam
[4] Utrecht University
[5] Universität des Saarlandes

Abstract

In the context of the European Network of Excellence in Computational Logic (CoLogNet, `http://www.cologne t.org/`), the European Association for Logic, Language and Computation (FoLLI, `http://www.folli.org`) has started a project on E-Learning in Computational Logic and the development of Dynamic Teaching Materials for its annual European Summer Schools (ESSLLIs). The project has a double aim: (i) to enhance the (re)usability of existing ESSLLI teaching materials by creating a richly structured repository; and (ii) to develop dynamic teaching materials for the upcoming ESSLLIs, integrating textual presentation, exercises, and computational tools (theorem provers, parsers) into a user-centered "living book". This paper presents the background of the project, gives some brief information about ESSLLI and describes the two subtasks in which the project is divided.

1. Background

While the term E-learning has only recently entered our vocabulary, it has already picked up many readings: such as on-line repository of teaching materials, learning by means of electronic tools, virtual courses or long distance learning. However, as we will explain below, we believe that in all its meanings, it has acquired great importance in the educational endeavor.

For a start, the increase of undergraduate students and lecturers' mobility within Europe due to the ERASMUS exchange programmes, and recently to the development of European Masters and Double Degree Progammes will lead to the establishment of standards for educational programmes which can be mutually recognized between several universities. It should also lead to well documented descriptions of the state of the art in educational methods in a broad variety of scientific disciplines so as to keep lecturers updated on the last scientific and didactic developments in their field. Thus, sharing teaching materials on line is an important stepping stone towards the establishment of such standards.

In addition, electronic tools have already proved to be an effective teaching support. First of all — as shown by Barwise and Etchemendy (1996); Cox et al. (1995) for teaching logic — when students can see what they are reasoning about, and when proofs are presented as graphs, they achieve far greater understanding of the subject than otherwise. Moreover, students have different learning rhythms. Using a computational assistant enables them to follow their own pace, checking their mistakes and working out further solutions as they need. When mistakes are pointed out by a machine, instead of a teacher, students are more motivated to understand the problems (Hoover and Rudnicki, 1996). For these reasons, we believe that electronic tools are an important support in teaching. This holds for Language Resources in general, and more particular for Computational Linguistics Tools (such as Parsers, Corpora and Ontologies).

Furthermore, e-tools are becoming part of the basic tool kits of researchers in applied fields, like Question Answering, Natural Language Interfaces to Databases, Semantic Web, etc. The modularity of the systems and the complexity of the tasks addressed require different areas of expertise which may be hard to be gathered within a single research group. Having ESSLLI's learning resources on-line will help speed up the research in these fields and enhance collaborations which go beyond traditional faculty borders and physical distance.

2. ESSLLIs

The European Summer School, organized by FoLLI, is the key European educational event for interdisciplinary exchanges in the fields of Logic, Language and Information for students, researchers and industrials. It has been organized yearly since 1989, contributing in this way to strengthen the community, facilitate sharing of common interests, form young researchers and provide a framework for the contact between the different fields. It offers around 40 courses per year at different levels (foundational, introductory and advanced) besides workshops, where outstanding results are presented.

The final output of the project will be a rich repository with integrated teaching materials provided by leading researchers in the area, and equipped with navigational tools and content search facilities which will be an endurable learning environment, easily accessible for both students and teachers. Below, we briefly describe the two subtasks of the project.

3. The ESSLLI Archive

The first subtask of the project is to create an infrastructure for on-line exploitation of the vast collection of existing ESSLLI teaching materials.

This project is supported by CoLogNet, Network of Excellence in Computational Logic, Contract No. IST-2001-33123. Maarten de Rijke was supported by the Netherlands Organization for Scientific Research (NWO) under project numbers 365-20-005, 220-80-001, 612.069.006, 612.000.106, 612.000.207, and 612.066.302.

Requirements

One of the first issues to be addressed in disclosing the ESSLLI teaching material is dealing with conversion into a standardized, easily accessible format. The ESSLLI CD archive (ESSLLI CD Archive, 2004) is a mirror of the CDs that accompanied the summer schools of the years 1997–2003. This archive includes various information about the courses given in the summer school of that year; typically, this data includes textual meta-data such as author name, description, prerequisites etc, and some teaching aids which are usually slides, handouts or articles in PDF, Postscript, Powerpoint, or DVI format. In some cases, the courses also contain pointers to additional external resources for the subject.

On the CDs, the course content is not provided in a standardized way (i.e., it differs from year to year in design and content); it is not searchable, and is only browsable to a limited extent

Within the task framework, web access should be granted to the ESSLLI archive, with the following features:

- **Search.** The data should be fully searchable. The search facilities must include full-text search, as well as searching by year, author, title and subject. Combinations of the search are also permitted, e.g., searching for a string of text in an article whose author is specified. The search mechanism will also include standard features of similar search facilities with which the target users are accustomed, such as searching for a quoted phrase, boolean searches, highlighting query terms in the results and displaying relevant snippets of the results.
- **Browse.** The data should be hierarchically browsable in several hierarchy trees, according to categories such as: YEAR, SECTION, and LEVEL.
- **Added Value.** Additional data which is currently not in the ESSLLI archive and may assist the user should be provided; this data includes cross references between related information, links to relevant external resources, and so forth.
- **Look and Feel.** The data should be presented in a clear, intuitive, and uniform way, making use of current web technologies. The user will also be able to access the original, non-modified ESSLLI CDs.

Implementation

The conversion of the current archive includes a number of stages. First, the "raw" data (the meta-information about the courses and the teaching material itself) is converted to a standard format; we chose to represent the data as XML documents, the de-facto standard for information representation. Every course and document from the ESSLLI CD archive is converted to an XML document using both publicly available tools for extraction of data from various formats, and specific tools developed for mining the archive for data.

Next, the data is enriched with external information, such as pointers to home pages of course authors, cross references between the courses, and so on. Once it is enriched, the uniform data is indexed and stored for fast retrieval; this allows for various queries such as *display all introductory courses from year 2000* or *search for "X" in the contents of all documents from courses in 2003.* For this stage, we use open source tools which are part of the Apache (Apache Software Foundation, 2004).

Finally, the repository is integrated into a Web Server which enables dynamic content, i.e. creation of web pages "on the fly" from XML data. For this, we use the Cocoon servlet technology, also part of Apache and based on the Java programming language. This enables maximum flexibility and security in the generation of web-accessible pages from the XML repository.

Current Status and Future Plans

As a pilot, we have started the ESSLLI archive creation for ESSLLI 2003. A large part of the teaching materials from this particular year has been converted into the standard XML format, indexed and stored; a Web Server has been set up for the repository, and currently allows browsing and searching it as defined in the requirements section. A methodology has been defined for extending the archive with additional years, so no changes to the server-side technology are required (only generation of additional content in XML, conforming to some rules). The pilot is accessible online (ESSLLI Web Site, 2004).

Our plans include overcoming technical difficulties which prevent some of the data from being indexed, as well as expanding and enriching the access methods to the information, and, of course, indexing the rest of the ESSLLI years.

4. Dynamic teaching materials

A specific advantage of events like the ESSLLIs is that they go beyond a single course and cover a topic by a series of courses that are closely related. As a consequence there arises a natural need to reuse parts of one such course for the preparation of or within another course. One of the objectives of the second subtask is to support this reuse by supporting the automated creation of specific content collections on the fly. Another advantage of ESSLLI is that the experts are available to support the students in getting hands on experience with actual research tools. This is interleaved with the teaching objectives in the second subtask by making these tools accessible from within dynamically generated documents.

To give the reader a feeling for what we are aiming for, we first sketch an existing setup, which already provides some basic levels of interactivity. Then we discuss how we can use Slicing Books Technology to leverage dynamic teaching materials into a user-centered 'living book'.

4.1. Hypertext functionality of the hyperref package

Sebastian Rahtz' hyperref package (available from (Comprehensive TEX Archive Network, 2004) and see (Goossens and Rahtz, 1999, 35–66)) is standardly used for turning the inherent document structure and cross-referencing information of LATEX documents into active hyperlinks. In addition to providing a basic navigational structure, the package has extended levels of functionality, al-

```
\hyperbaseurl{...} % URL for the CGI-executables

\newcommand{\hyperfrag}{...} % grammar fragment URL

\newcommand{\parsescript}[3]{\href{netgrail?% call the parser engine
  url=\hyperfrag&% load fragment from URL
  struct=yes&% structural rule output
  sem=no&% meaning assembly shown
  lexsem=yes&% substitute lexical semantics, 'no' for proof terms
  unary=inactive&% ignore semantics for Diamond/Box operations
  mode=nd&% e.g. natural deduction format
  goal=#1&% goal formula
  test=#2}% your test phrase for the script call ('+'separator)
  {#3}}% your test phrase for typesetting
```

Figure 1: Calling a cgi-script with the hyperref package

lowing the user to specify hypertext links to external documents and URLs, including linking through the Common Gateway Interface (CGI).

In the Computational Linguistics programme at Utrecht University, this extended functionality is used in dynamic teaching materials familiarizing students with a number of computational grammar formalisms. The client-server interaction takes the following form (see (Moortgat et al., 2002) for a full description):

- Students present a linguistic analysis in the form of a grammar fragment. Formalisms currently supported are Stabler-style Minimalist Grammars, and type-logical grammars. A fragment, in these frameworks, consists of a set of lexical type declarations, structural options, and a sample of test phrases.
- Fragments are submitted to the server, where they are turned into dynamic PDF documents. The test sample is hyperlinked to CGI scripts interfacing with general parsers/theorem provers for the formalisms under consideration (The GRAIL type-logical theorem prover (Moot, 1996), CKY deductive parser for Minimalist Grammars (Stabler, 2001)).
- The student (or teacher) can produce derivations 'on-demand' for the test sample in a number of available formats. The server returns these derivations as PDF documents, which can then further be integrated, commented, etc.

The reader is invited to try out the setup at the portal site `http://grail.let.uu.nl`, or to inspect the two possible permutations for the sentence 'Naoko ate Hiromi's sushi' below (from a syntax take-home test). Clicking the sentences fetches their derivation from the server, and displays them in the format specified by the user — Natural Deduction style, in this case. Figure (1) gives the essential code.

1. Naoko ga Hiromi no osusi o taberu.
2. Hiromi no osusi o Naoko ga taberu.

Using these tools with a PDF-enabled web browser, one obtains a seamless client-server interaction. Still, the described hyperref-based architecture has certain limitations: automatic navigational features are restricted to the LaTeX-internal crossreferences, additional linking has to be provided by hand; similarly, the integration of server output with the documents from which the parser engines are called, requires manual post-editing. In the next section, we show how these limitations can be overcome with the aid of SIT (Slicing Information Technology, 2004).

4.2. Living book

The core idea of the SIT approach is to semi-automatically break up LaTeX source code into semantic units, thus providing mark-up that goes beyond the logical document structure. The semantic text-units become 'recombinant' components of a dynamically unfolding document, customized to fit individual readers' needs. Depending on his/her background knowledge and preferences, the user can integrate textual presentation of the material thaught with his/her own crossreferences, explanations, exercises and solutions, possibly obtained with the use of integrated external parsers, theorem provers and the like.

To implement the SIT approach in the preparation of ESSLLI course materials, we distinguish the following steps.

- The authors of selected courses prepare their teaching materials according to ESSLLI style guidelines, and submit them to the SIT Splitter for initial slicing. This phase provides mark-up at the level of general logical-mathematical knowledge.The SIT Splitter decomposes these documents into re-usable learning objects at an agreed granularity. For a learning object to be reusable it is not necessary that it makes sense on its own. Rather it is essential that it makes sense in a context which can be precisely described so that it can be automatically reconstructed whenever this specific learning object is to be reused. Experience shows that in mathematics related documents like those to be handled for this project an average granularity of 5 slices per page is necessary to achieve maximum possibilities of reuse. Note that providing a uniform LaTeXstyle for ESSLLI authors is an important help for a reliable automated slicing.
- The sliced manuscripts are returned to the authors,

who deepen the re-engineering transformation on the basis of their domain-specific knowledge. This phase concentrates on further assigning key phrases, defining extra semantic relations between document slices, and adding components relying on server interaction, such as described above.

- The results of the document re-engineering process are made available as content packages according to the open IMS Content Packaging Specification (see (IMS Global Learning Consortium, Inc)). This specification is supported by many e-learning platforms, opening up the possibility for later reusing specifically built ESSLLI documents in these environments. Added meta-information is encoded in XML in accordance with the open Trail-Solution Metadata Specification and Thesaurus Specification (see (TRIAL Solution, 2004)).

The tool for the dynamic generation of personalized documents, the SIT Reader, utilizes declarative descriptions of the intended structure of documents to be delivered for specific usage scenarios. Deep inside the tool there is an automated theorem prover, called sl-engine, which combines these descriptions with the learning object metadata and with information about the knowledge of the user in order to infer what should be proposed to the user for reading. Another application of the sl-engine is to provide internal inferences in order to obtain a more complete user model. We mention that sl-engine is in part based on methods which are taught to students at ESSLLI.

In order to use a sliced book, the student selects parts she is especially interested in and asks the server to complete the selection automatically for a specific purpose, for example by adding necessary prerequisites or exercises or material from related courses but omitting material that has been inferred to be known.

In a second attempt interactivity is added to the sliced book, turning it into what we call the Living Book. The approach taken here uses interactive pdf documents with embedded JavaScript which make up a connection from the dynamically generated teaching material to some tools running on possibly remote servers.

As an application a student may enter some formula into an input field in the pdf document and will on request receive a newly generated pdf document where the server has added some evaluation of this formula. Another application is the random generation of exercises which take the knowledge of the student into account.

Authors use a simple generic interface in order to bind interactive systems to their teaching materials. Input forms are described in the LaTeX source documents by using the possibilities of the aforementioned hyperref package. At the places where the reply from the server should go into the document, the author inserts a LaTeX command `\tsdynamic{<script>}{<template>}`. `<script>` denotes the name of a program (which can be written in any programming language) that is called by the SIT Reader with parameters describing the user input and how to access the SIT Reader user model. In addition `<script>` may also store data in a protected area of the SIT Reader server in order to correlate different requests. `<script>` is supposed to generate a fragment of LaTeX source code which replaces the `\tsdynamic` command. The `<template>` parameter can provide prepared LaTeX source code which is then filled up by the `<script>`. Finally the SIT Reader generates the pdf document for the learner, using the content generated by `<script>`.

A variant of this approach is to launch an interactive system in a separate window from within the personalized pdf teaching material and only integrate the final result of the students work into a new version of the material, but not following each interaction.

The reader is invited to investigate these possibilities at (Furbach, 2004). We mention that this installation allows as additional features editing of personal annotations (which can be typed or hand written with a tablet pc) and the generation of different views for print, PC screen, Palm Pilot or Pocket PCs. These are basically applications of the same technology for adding interactive systems which has been described above.

The current status of this component of the project is that for the upcoming ESSLLI (ESSLLI 2004. 16th European Summer School in Logic, Language and Information), a coherent set of courses in the logic and language field have been selected to serve as a pilot for the application of the Living Book approach. The pilot is aimed at providing a set of procedures and tools that can then be used for the preparation of future course materials, and that can be made available as author instructions (ESSLLI Repository, 2004).

5. Conclusion and Outlook

The implementation of the outlined project results in a flexible learning environment with a functionality clearly extending beyond the CoLogNet project period in which some of the described methods and tools have been applied to teaching material in Computational Logics. Together with related initiatives (such as Milca (MiLCA, Medienintensive Lehrmodule in der Computerlinguistik-Ausbildung) and LoLaLi (LoLaLi. Logic and Language Links, 2004), it can provide the starting point for the set-up of an encompassing web-based resource center of life-long education in the various disciplines represented at the transdisciplinary ESSLLI summer schools such as Computational Linguistics, Formal Semantics, Computational Logics and Artificial Intelligence.

The segmentation of teaching materials into 'recombinant' units is a novel contribution to the transformation of todays educational instruments into an e-learning setup. In principle the method can be applied to written teaching material such as text books and course scripts in any academic field. However, the method favors clearly structured texts with recognizable units and sub-units such as statements, arguments, conjectures, theorems, proofs, examples, derivations. It may not have been accidental that the origins and first testing grounds for the methodology have been in Computational Logics.

However, the method has far-reaching consequences. For centuries, printed books have constituted the preferred representation for the storage and transfer of human knowledge. Today distributed knowledge sources, hypertexts and

powerful systems for computer aided instruction demand more flexible representations of the same knowledge utilizing a richer explicit structure. While this change is taking place, more books are written. The transformation of existing and new textual knowledge into a representation supporting novel forms of learning and knowledge management is an important challenge for all scientific disciplines.

The linking of conceptual units with external resources such as other information sources, computational systems and visualization tools also has applications that extend far beyond the utilization sketched in this paper. One of these applications is of special interest to the field of language recources and technology evaluation. A revolutionary development for the study of language and for the evaluation of language technology has been the annotation of data with linguistically motivated interpretations. It started with simple part-of-speech annotation, progressed with treebanks and has recently led to some semantic interpretation of data such as in prop-banks and frame-banks. On the other hand, scientific publications link back to the data that were exploited to obtain and verify the linguistic and technological results. The gradual emergence of a solid empirical methodology linking data and their scientific interpretation is changing both theoretical and computational linguistics.

A next logical step in the preparation of teaching material in the ESSLLI disciplines will be the exploitation of the hyper-referencing mechanisms for the linking of statements on language, methods and processing tools with data that exemplify the observations, illustrate the effects of the methods and evaluate the processing components. In this way remote language resources can be integrated into the teaching material. This linking will be especially relevant for courses on empirical methods. But it can also be utilized for courses on formal linguistic analysis or computational methods for the processing of human language.

We expect that the developed methods will be successfully tested in future ESSLLI summer schools. In this way a larger number of students and teachers will contribute to their improvement and hopefully also become interested in their application within other contexts. We hope for a dissemination into regular academic programs, e-learning initiatives, and the European Master Programmes.

6. References

Apache Software Foundation, 2004. `http://www.apache.org/`.

Barwise, J. and J. Etchemendy, 1996. Visual information and valid reasoning. In G. Allwein and J. Barwise (eds.), *Logical Reasoning with Diagrams*. Oxford University Press.

Comprehensive TEX Archive Network, 2004. `http://www.ctan.org/`.

Cox, R., K. Stenning, and J. Oberlander, 1995. Contrasting the cognitive effects of graphical and sentential logic teaching: reasoning, representation and individual differences. *Language and Cognitive Processes*, 10(3/4):333–354.

ESSLLI 2004. 16th European Summer School in Logic, Language and Information, 2004. `http://esslli2004.loria.fr/`.

ESSLLI CD Archive, 2004. `http://www.folli.org/CD/`.

ESSLLI Repository, 2004. `http://www.folli.org/ESSLLI/`.

ESSLLI Web Site, 2004. `http://www.esslli.org/`.

Furbach, U., 2004. Logic for computer scientists. Electronic course. `http://slice.uni-koblenz.de/~sitlog/`.

Goossens, M. and S. Rahtz, 1999. *The LaTeX Web Companion. Integrating TEX, HTML, and XML*. Reading, MA: Addison Wesley.

Hoover, H. and P. Rudnicki, 1996. Teaching freshman logic with mizar-mse. In *Workshop on Teaching Logic and Reasoning in an Illogical World*.

IMS Global Learning Consortium, Inc, 2004. `http://www.imsproject.org`.

LoLaLi. Logic and Language Links, 2004. `http://lolali.net/`.

MiLCA, Medienintensive Lehrmodule in der Computerlinguistik-Ausbildung, 2004. `http://milca.sfs.uni-tuebingen.de/`.

Moortgat, M., R. Moot, and R. T. Oehrle, 2002. Teaching tools for logic-based grammar development. *Eutupon*, 9:1–14. Available at `http://obelix.ee.duth.gr/eft/`.

Moot, R., 1996. *Proof Nets and Labeling for Categorial Grammar Logics*. Master's thesis, Utrecht University. Code available at `http://www.labri.fr/Perso/~moot/grail.html`.

Slicing Information Technology, 2004. `http://www.slicing-infotech.de`.

Stabler, E., 2001. Minimalist grammars and recognition. In C. Rohrer, A. Rossdeutscher, and H. Kamp (eds.), *Linguistic form and its computation*. Stanford: CSLI Publications. Online access at `http://131.211.190.177/mgcky/`.

TRIAL Solution, 2004. `http://www.trial-solution.de`.