

Bijlage 24

Summary of indexing-related activities at EuroT_EX '98

Roger Kehr
Computer Science Department
Darmstadt University of Technology
kehr@iti.informatik.tu-darmstadt.de

editor's note

This is an adaptation of the ascii file posted to the xindy newsgroup by Roger in the week following EuroT_EX.

As some of you may have noticed, I gave a talk together with Joachim about xindy at the EuroT_EX'98 conference in St. Malo, March, 30th.

On March, 31st we had a BOF session about xindy where we discussed several issues how xindy can be further improved and extended.

The most important results from this session and some other issues are as follows:

- Improvement of L^AT_EX support in the future. This mostly concerns a L^AT_EX style file that writes the index entries into the .aux files. Additionally, we think about L^AT_EX macros that write information about the document itself (such as encodings and language) into the .aux file as well. This enables one to add index style definitions into the L^AT_EX source, instead of a completely separate index style file, which seems to be one of the greatest hurdles for the acceptance of xindy.
- Support for input filters. Many have asked for such a feature. This allows to tag index entries from the raw index with an additional attribute, indicating the index class the index entry belongs to. xindy will be extended to accept only those index entries that belong to the selected classes.

My current suggestion is to add a new option :CLASS <string> to the INDEXENTRY command to indicate the class of an index entry. Additionally, we need another index style command such as (INPUT-FILTER (<list-of-class-names>)) that defines the set of class names to accept. This specification should be made available to the command line options of xindy as well.

- Hans Hagen asked for a way to pass data attachments through the indexing process into the tagged index. This would enable xindy to process glossaries (with the glossary text as a data attachment) and sort the entries with xindy.

Several small problems have to be solved to implement such a scheme. My suggestion is to add data in the form of a dictionary of keyword/value pairs in the raw index interface. The question is whether data should be attached to location references or index entries or both. Any ideas are welcome.

This dictionary asks how the mark-up of this data in the backend should be implemented. I have some ideas how to realize this, but this will need some time.

- Add new hooks in the sorting process to allow sorting rules that can be applied only once at the beginning of the rewriting procedure.
- Further simplification of the installation procedure.
- Re-structuring of the documentation.
- many other things to do . . .

Hope, this gives you an impression what is planned for the future.