

The MATE Workbench

Laila Dybkjær and Niels Ole Bernsen

Natural Interactive Systems Laboratory, University of Southern Denmark
Science Park 10, 5230 Odense M, Denmark
laila@nis.sdu.dk, nob@nis.sdu.dk

Thematic area: language corpora and related tools.

Keywords: spoken dialogue corpora, multilevel annotation, information extraction, workbench.

1. Abstract of technical content to be demonstrated

The aim of the European Telematics project MATE (Multilevel Annotation Tools Engineering) has been to facilitate the use and reuse of spoken language resources by addressing theoretical issues and implementing practical solutions. MATE was launched in 1998 in response to the increasing need - not least in the area of spoken language dialogue systems research and development - for tools and standardisation efforts in support of efficient markup of spoken dialogue corpora at multiple levels.

To provide a solid basis for the coding standard to be proposed by MATE, more than sixty existing coding schemes belonging to five different coding levels (i.e. prosody, (morpho-) syntax, co-reference, dialogue acts, and communication problems) and their cross-level interaction were reviewed (Klein et al. 1998). On this background, the MATE markup framework (Dybkjær et al. 1998) was proposed as a standard which can facilitate uniform description of schemes across levels. For each level, one or more of the reviewed coding schemes were taken as the starting point for the definition, following the MATE markup framework, of one or more best practice coding schemes (Mengel et al. 2000).

The MATE software workbench (Isard et al. 1998, Isard et al. 2000) supports the MATE markup framework and incorporates the MATE best practice coding schemes. It is implemented in Java 1.2. It has been tested on Unix (Solaris) and Windows (NT and 98) but should run on any platform for which Java 1.2 is available. The workbench has a modular architecture which facilitates the addition of new modules and new tool functionality by its users. XML is used for internal file representation.

The MATE workbench has the following major components: an internal database which is an in-memory representation of a set of hyperlinked XML documents; a query language and processor which are used to select parts of this database for subsequent display or processing; a stylesheet language and processor which respectively define and implement a language for describing structural transformations on the database; a display processor which handles the display and editing actions; and a user interface which handles file manipulation and tool invocation.

Stylesheets are used for specifying the visual presentation of data to users. Stylesheets are written in the MATE Stylesheet Language (MSL). The emerging standard in this area is XSLT, but XSLT was not fully defined when the workbench was being designed, and lacked various necessary functionalities. It was therefore decided to implement MSL which uses the MATE query language but is otherwise similar to XSLT.

The MATE workbench offers support for annotation of spoken dialogue corpora and information extraction from annotated corpora as its two main functionalities. Both aspects will be demonstrated in detail during the presentation. The annotation support includes:

- Ready-to-use best practice coding schemes for the levels mentioned above. Stylesheets are included which allow the user to view and annotate files in an appropriate way using any of these best practice schemes. Example dialogues are also provided which illustrate how

a tagged dialogue and the accompanying tag set will be shown when a given best practice scheme is being applied.

- Easy addition of new coding schemes for existing or new levels. New coding schemes are entered by using the coding module editor. A coding module is the standard type of coding scheme description proposed by MATE. A coding module prescribes what constitutes a coding, including markup representation and relations to other codings.
- Editing of files. MATE operates with three basic kinds of files. XML files are coding files. MSL files are stylesheets used for visualisation of codings and for annotation support. Run files specify which stylesheet to apply to which XML file. Any XML, MSL or run file in a MATE project folder can be opened and edited.
- Creation of new project folders and addition or creation of new files, including header file documentation.
- An audio tool for listening to speech files and displaying the sound file as a waveform.
- Conversion to XML format of files in other formats. For the moment, conversion from XLabels and BAS Partitur to XML is enabled but new converters can easily be added, including converters which export from XML to other formats, such as HTML.

As regards information extraction, a query tool is available for selecting the document(s) to be searched and for specifying the information to look for. The information which can be extracted includes statistical information.

An online help facility is available and may be consulted at any time during use of the MATE workbench..

Acknowledgements

We gratefully acknowledge the support to the MATE project provided by the European Commission's Telematics/Language Engineering Programme. We would also like to thank all MATE partners. Without the very considerable joint efforts of the project consortium it would not have been possible to build the MATE Workbench.

References

MATE deliverables are available from the MATE web site at <http://mate.nis.sdu.dk>.

Dybkjær, L., Bernsen, N. O., Dybkjær, H., McKelvie, D. and Mengel, A.: The MATE Markup Framework. MATE Deliverable D1.2, 1998.

Isard, A., McKelvie, D., Cappelli, B., Dybkjær, L., Evert, S., Fitschen, A., Heid, U., Kipp, M., Klein, M., Mengel, A., Møller, M. B. and Reithinger, N.: Specification of Workbench Architecture. MATE Deliverable D3.1, 1998.

Isard, A., McKelvie, D., Mengel, A., Møller, M. B., Grosse, M. and Olsen, M. V.: Data Structures and APIs for the MATE Workbench. MATE Deliverable D3.2, 2000.

Klein, M., Bernsen, N. O., Davies, S., Dybkjær, L., Garrido, J., Kasch, H., Mengel, A., Pirrelli, V., Poesio, M., Quazza, S. and Soria, C.: Supported Coding Schemes. MATE Deliverable D1.1, 1998.

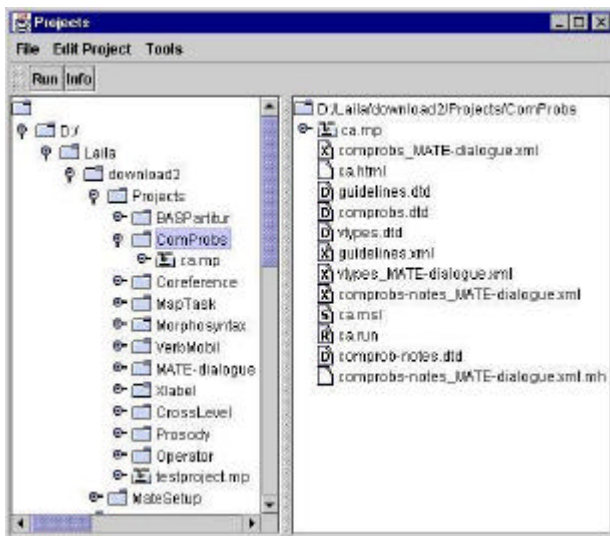
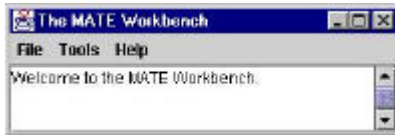
Mengel, A., Dybkjær, L., Garrido, J., Heid, U., Klein, M., Pirrelli, V., Poesio, M., Quazza, S., Schiffrin, A. and Soria, C.: MATE Dialogue Annotation Guidelines. MATE Deliverable D2.1, 2000.

2. Presentation script

The MATE workbench can be downloaded from <http://mate.nis.sdu.dk>.

Starting the MATE workbench

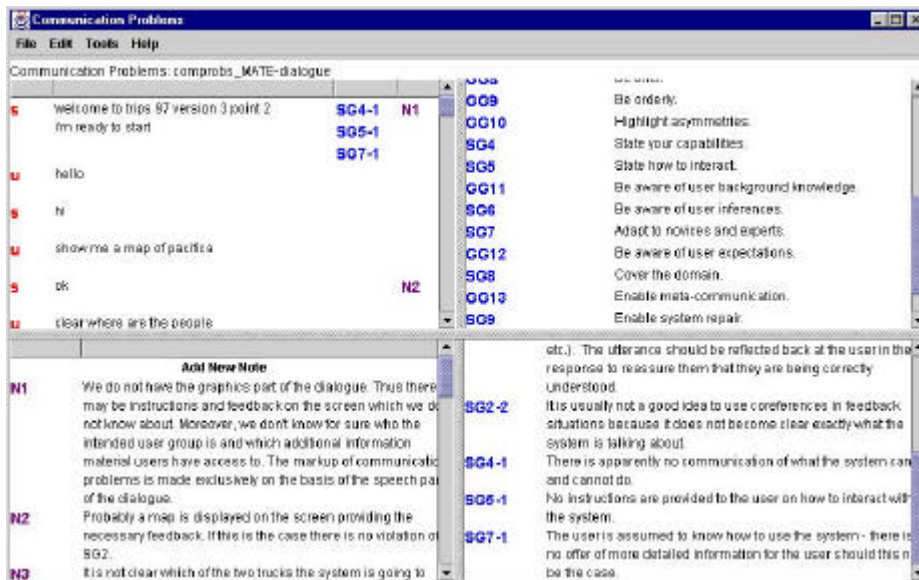
Upon start-up the following two windows will appear (the control window and the projects window).



Annotation

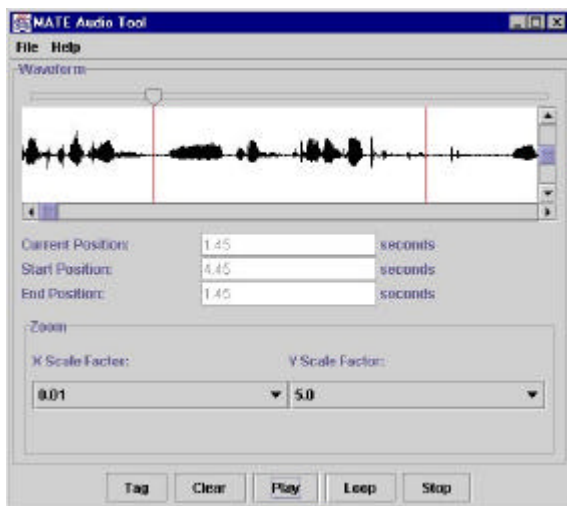
Annotation can be done using one of the MATE best practice coding schemes or users may add their own coding schemes. The two windows below show annotation based on the MATE MapTask scheme and the MATE communication problems scheme, respectively.



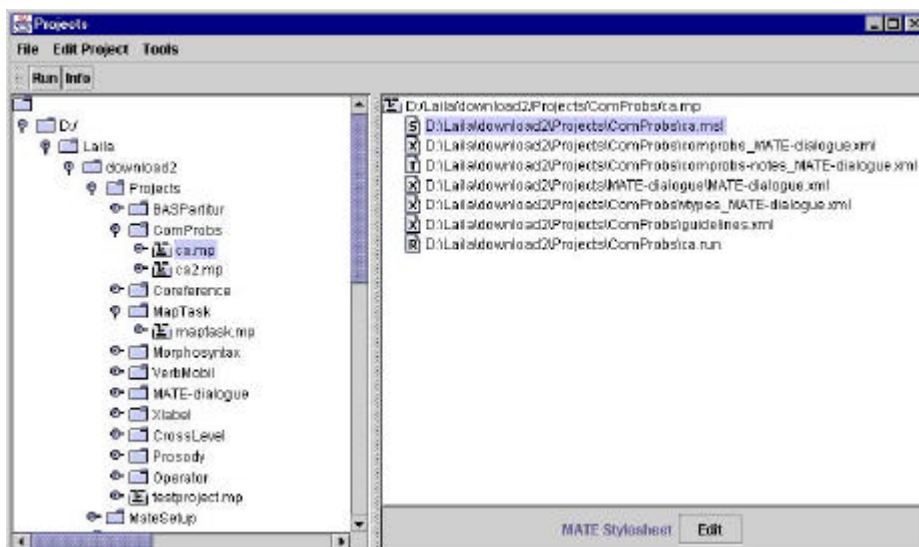


Listening to audio files

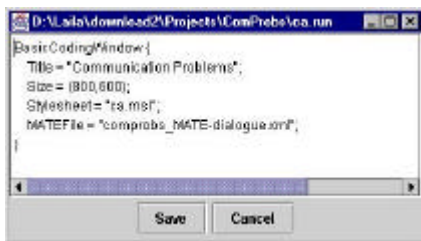
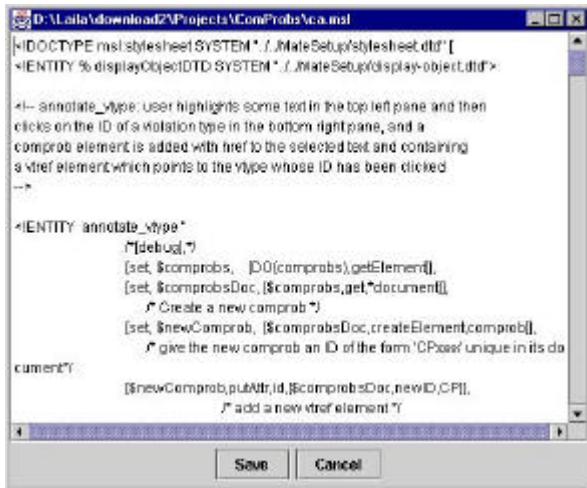
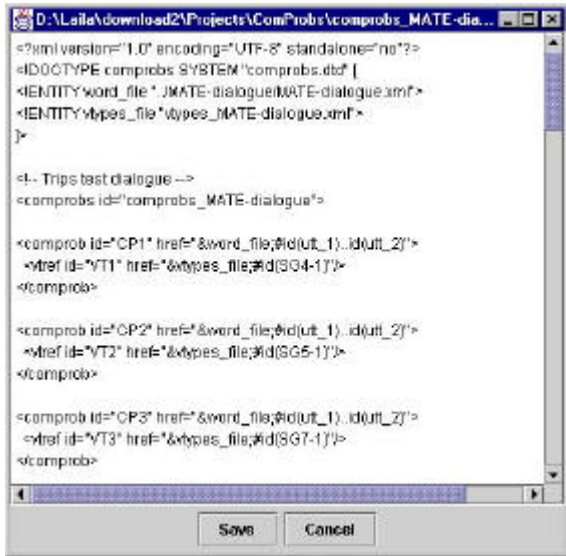
Audio files can be selected in the projects window and played. The window below will then appear.



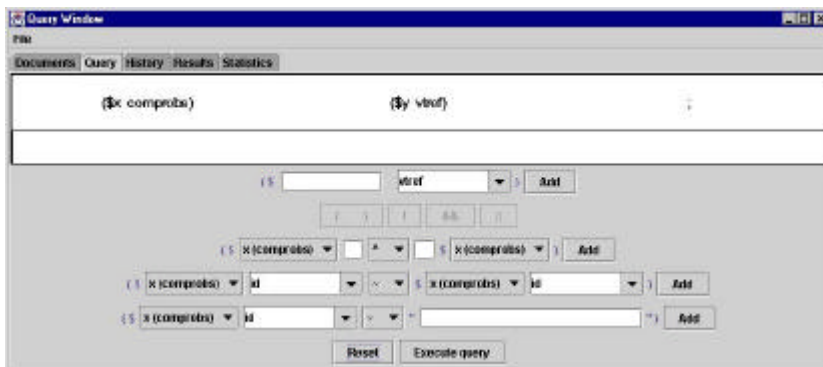
Editing a file



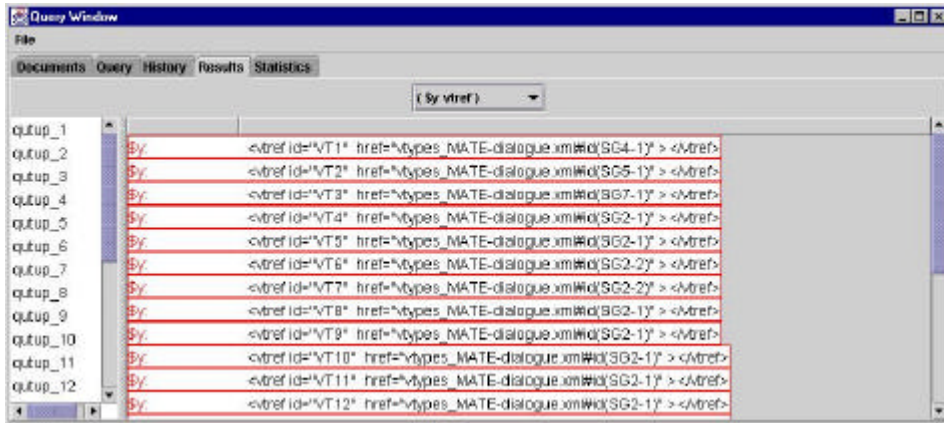
MATE XML, MSL (stylesheets), and run files in MATE project folders can be edited by selecting the file and clicking Edit in the projects window (see window above).



Extracting information from annotated corpora

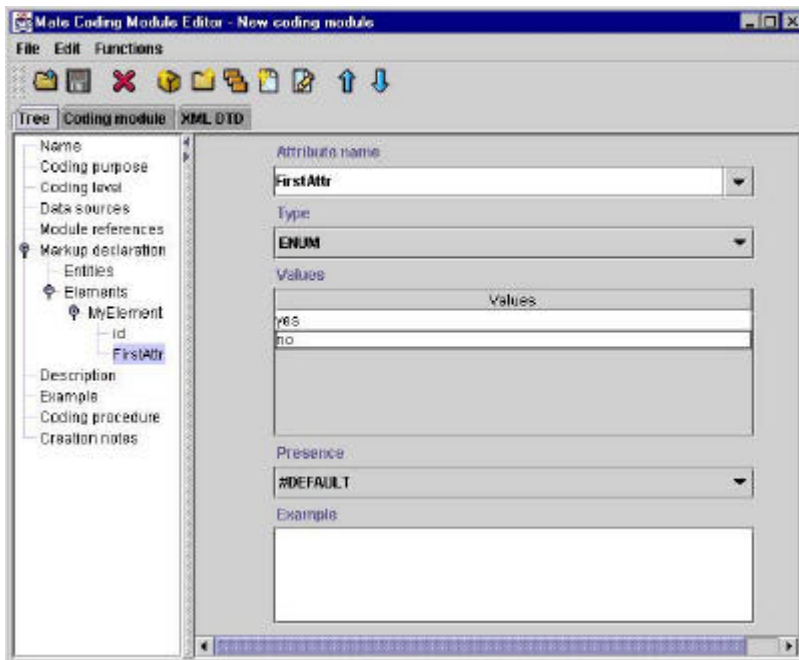


From the query window one or more MATE documents can be selected and queried (see window above). Results are shown as exemplified below.



Adding a new coding module

Users may add new coding modules (coding schemes) themselves via the coding module editor shown here:



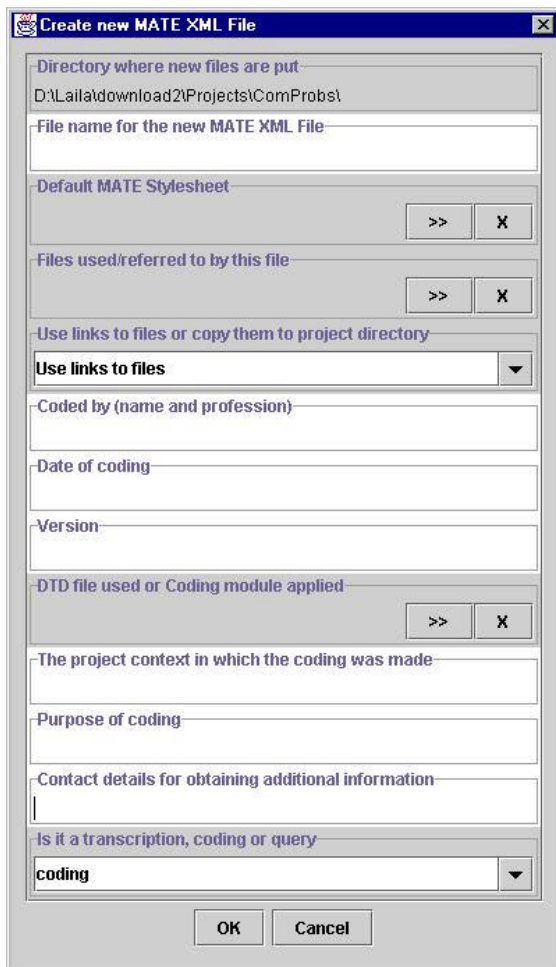
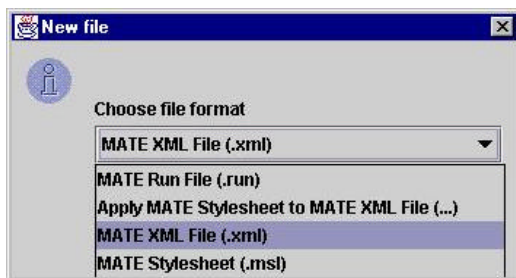
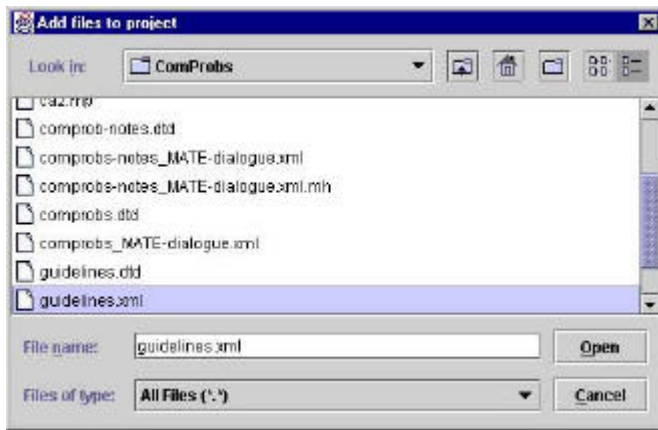
Adding a new project

A new MATE project folder can be created as follows:



Adding a new file

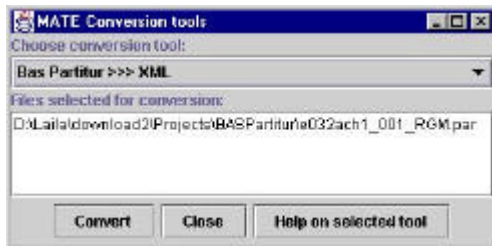
Existing files may be added to a project folder or new files may be created.



As a minimum, the new file must be given a name, and an applicable DTD or coding module must be specified. However, it is recommended that additional header information is provided as well.

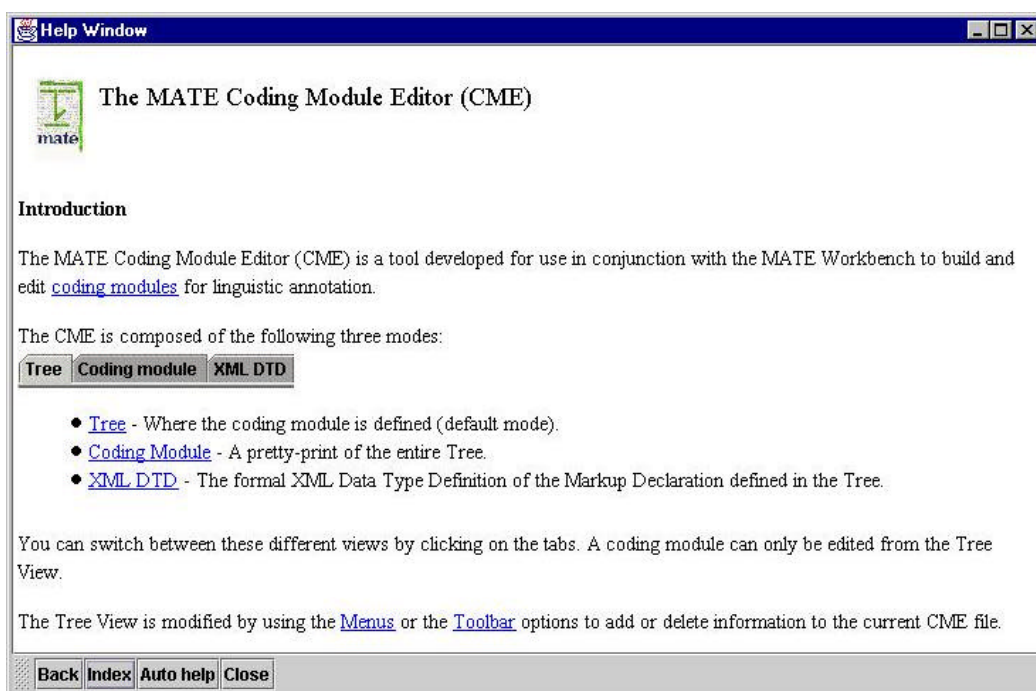
Import from, and export to, other file formats

Files may be imported from other formats to XML. Below is shown import from BAS Partitur.



Getting help

A help window can be opened which provides information on various workbench issues. Below is shown the topmost help page for the coding module editor.



3. Hardware and software requirements

We will bring a portable PC with the MATE workbench installed.

However, we would very much like the local organisers to provide us with either a data projector or a large screen. If the demonstration is a, say, 20 minute presentation in a lecture hall in front of a large audience, a data projector is needed. If the demonstration is on exhibition- or poster- like conditions where people come and go and ask questions and discuss, we would prefer a large screen.