DML-CZ Metadata Editor

Content Creation System for Digital Libraries

Miroslav Bartošek, Petr Kovář and Martin Šárfy

Institute of Computer Science, Masaryk University, Brno, Czech Republic
bartosek@ics.muni.cz, kovar@ics.muni.cz, sarfy@ics.muni.cz

Abstract. The aim of the DML-CZ project (2005–2009 – Czech Academy of Sciences, Masaryk University in Brno, Charles University in Prague, Czech Republic) is to investigate, develop and apply techniques, methods and tools that would allow the creation of the Czech Digital Mathematics Library.
The most important tool developed and used in the course of the project is the Metadata Editor – a complex web-based system supporting all essential steps in the development of the article oriented digital library: integration of scanned pages (journals, proceedings, monographs) into hierarchical structures, article building, detailed metadata description up to the level of articles and book chapters, article bibliography references processing and linking, name authority management, inclusion of born-digital material, automated metadata verification, and generation of the resulting PDF papers. The rights management in combination with the remote access enable to distribute the work of a digital library among many people with different levels of expertise. Building a library of more than 15.000 articles proved the soundness of the Metadata Editor architecture and implementation. An overview of the system architecture and functionality is briefly revealed in our paper.

1 Introduction

In the Czech Digital Mathematic Library project[1] (aimed to digitize and present relevant Czech mathematical literature since the 19th century) we focused on both restoring the original look-and-feel of the historical materials and delivering content-rich database with features required by today's scientific world like extensive hypertext linking or powerful searching mechanism [1].

There are many steps neccessary to achieve this goal [4]. An overall schema of the DML-CZ workflow is depicted in Figure 1. Scanned images of journals, proceedings and monographs are imported into hierarchical structure and identified with the article reference metadata harvested from reference databases Zentrallblatt-MATH or Mathematical Reviews (step 1).

[1] Project 1ET200190513 – funded by the Academy of Sciences of the Czech Republic Programme "Information Society" (National Research Programme, 2005–2009)

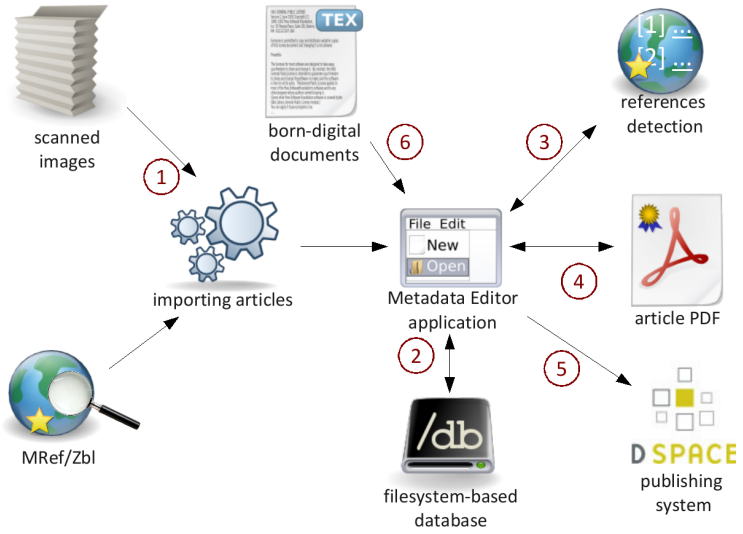

Fig. 1. An overall picture of the workflow used in the DML-CZ project

Detailed article descriptive metadata is then reviewed, corrected and completed by mathematicians (step 2). Processing of the article bibliographical references consists of reference block detection in OCR sources, splitting the block into particular references and structuring of individual reference. Mathematical reference databases are then queried to identify corresponding records and to make hypertext links (step 3).

The scanned images are enriched by OCR [5] and they form, together with the generated cover page, the resulting article PDF file (step 4). These PDF files as well as all descriptive, structural and administrative metadata are then sent to our publication system based on the DSpace repository system (step 5).

There is also support for incorporating born-digital documents in our workflow (step 6). Publishers can use the tools we developed for collecting, managing and publishing the digital libraries, greatly reducing their overhead costs.

It can be observed that to achieve the DML-CZ goals we needed a tool powerful enough to handle all the workflow processes. During the last three years, we gradually developed such a system. Metadata Editor (ME) is a web-based application that allows users to effectively manage digital library content creation.

In this article, we briefly describe ME essential features and give a technical overview to our solution.

2 Metadata Editor

Metadata editor (editor.dml.cz) is a client-server application consisting of a web interface, a suit of supporting scripts and an internal database. In the following

we describe the Metadata Editor workflow used in our project. The main steps of the worflow are as follows:

- loading the input data into the ME internal structures;
- article building – defining the logical structure of digitized publications;
- metadata editing – creating descriptive metadata records from journal/ proceedings series/monograph levels up to the article or book chapter level;
- bibliographical references processing - creating, harvesting and linking lists of references;
- automated metadata verification;
- final PDF compilation and export to the publication system.

2.1 Input data and its structure

Metadata Editor works with data and metadata prepared in previous phases of the DML-CZ workflow from different sources, such as:

- digitized old printed documents (created in the scanning phase);
- materials already existing in some digital form (retrieved in the *retro-born-digital* conversion phase);
- born-digital publications inserted to ME on-line by publishers (newly published journal issues created automatically 'as byproduct' of a publishing process).

Metadata editor focuses primarily on scanned documents, but it can handle other sources as well (usually by using simplified or slightly modified workflow).

All the data obtained from the scanning/conversion phases (page images, initial structural and page description metadata) are validated with respect to their completeness and consistency, page order correctness, duplicities, etc. The data is then restructured, stored in the hierarchical directory structure suitable for further processing and enriched by metadata gained from OCR and the mathematics reference databases.

The Metadata Editor organizes objects in the following hierarchical structures:

- **serials** – journal/volume/issue/article,
- **proceedings** – proceeding series/proceedings volume/article,
- **monographs** – collection/monograph/chapter.

Each object in the Metadata Editor is managed using a unique identifier (e.g. `serial/AplMat/12-1967-2/#4`) which reflects the path inside the directory structure where it is stored (the identifier also forms a part of the object's URL).

2.2 Article building

Combination of several methods is used to automatically create the initial structure of articles of a journal issue (proceedings volume), and also to minimize

the manual workload in the article building step. This includes exploitation of pagination information from reference metadata and localization of beginnings and ends of articles in OCR-ed texts.

Tieing the pages automatically into the article structure is not always reliable and a manual check of the structure is still necessary. Sometimes the pages are badly assigned to articles, or some articles are not detected at all. It is then necessary to move pages, to create new articles or to delete false ones. This problem applies to scanned documents only – born-digital articles are well-structured implicitly.

Metadata editor provides effective ways to handle the article building task. The most interesting tool is the visual article editor: the human operator works with the page thumbnails on a screen arranged tabularly like cards laid on the desk, as can be seen in Figure 2. This allows an easy visual inspection of pages, verification of the page ordering, reshuffling pages within an article and/or between articles, cancellation of badly identified articles and constituting the missing ones, removing blank pages, etc. By clicking on a thumbnail a large page image is open in a new window, allowing the operator to examine details of a given page.

Page thumbnails are grouped into blocks of two types/colours: green blocks represent individual articles, red blocks consist of pages excluded from article processing (blank pages, front- and back-matter, advertisements, etc.)

A set of auxiliary functions is available to handle a non-standard structuring of old printed journals (interleaving articles or page numbering schemas, articles crossing issue boundaries, etc):

- page cloning (allowing to clone pages belonging to more than one article),
- download/upload of page images (allowing for local corrections/improvements in images),
- page number editing,
- page reshuffling within an article and/or journal issues,
- grouping of articles in named sections and subsections,
- and other.

Three different numbering schemas are used to identify pages in the Metadata Editor:

- **physical page numbers** – numbers printed on original sheets of paper,
- **logical page numbers** – the unique identifiers of pages within an issue/proceedings/monograph,
- **sequential page numbers** – keys defining order of pages within an issue/proceedings/monograph.

Logical numbers are always decimal numbers and are derived from the image file names assigned during the scanning phase. Sequential numbers keep pages in the right order; they are not directly visible to the operator.

If the automated article building process fails significantly, it may be time-consuming to create an article structure manualy; in these cases a batch process

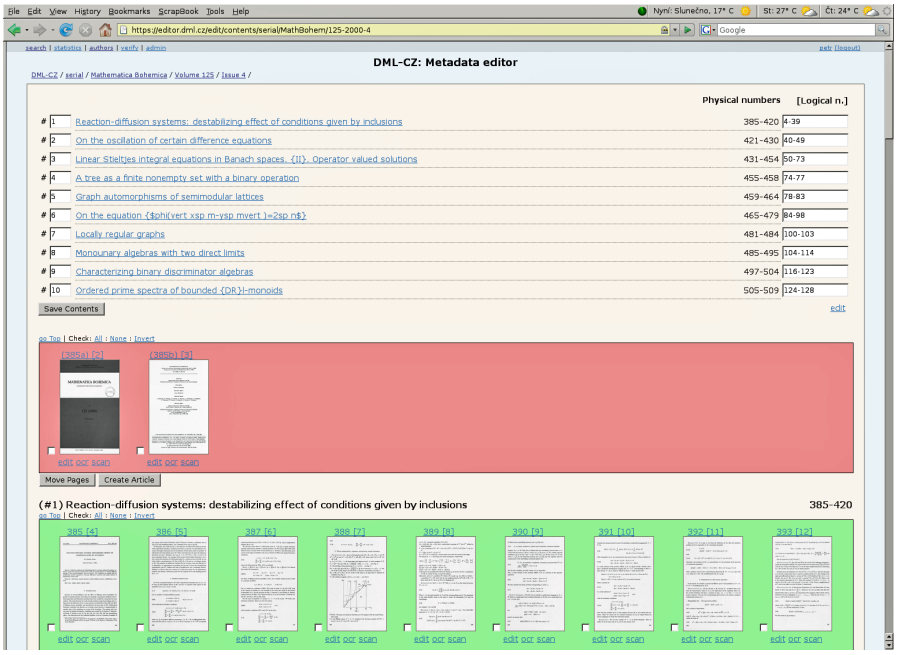

Fig. 2. Contents structure editing page

of article building can be used to create all the articles in an issue at once, leaving the visual article editor for final visual checking only.

Building articles (defining structural metadata) is permitted only to operators with appropriate structure editing rights.

2.3 Metadata editing

After the pages are grouped into articles and the document structure is created the article metadata editing step is unlocked. The metadata record is usually pre-filled with the reference metadata by the automated process. The operator is typically required just to check it and/or to make minor changes only.

The article metadata editing screen is split into two parts: the metadata editing form and the preview area. The left part of the screen consists of a form where the metadata is edited. The preview area displays the first page of the article, so the operator can easily access it while editing the article metadata. It is possible to flip through the pages using the list of the page links or keyboard shortcuts.

The following metadata are assigned to an article:

- **Title** – article title in several languages (title in the original language and its translation into English is required at least),
- **Author** – author's name verified against the name authority database,

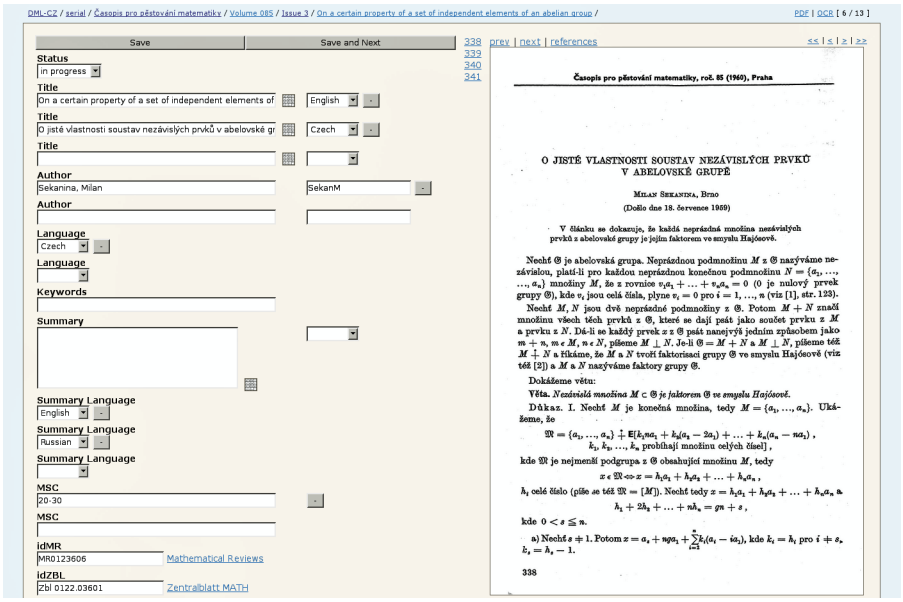

Fig. 3. Article metadata editing page

- **Language** – language of the article,
- **MSC** – MSC codes specifying the topics of the paper,
- **Summary Language** – language of the article summary,
- **Article Type** – type of article: math, physics, editorial, table-of-contents, history, ...
- **Accessibility** – can the article be made publicly visible?
- **idMR, idZBL, idJFM** – identifiers to external databases,
- **Status** – metadata record processing status: untouched, in progress, completed.

The operator can display the OCR text of a page in a separate window and use it for copy-and-paste editing. The author field is connected to the Name Authority database. By writing down just first few letters of the author name the operator is offered a list of matching names, allowing to choose the author name correctly without mistakes. The MSC field offers similar functionality: when writing a code into the MSC field, the Metadata Editor displays the corresponding code description. Data in idMR/idZbl/idJFM fields serve as links to reference databases MathSciNet, Zentralblatt-MATH and Jahrbuch über die Fortschritte der Mathematik. By clicking on these anchors, the appropriate record of the given database is openned in a new window allowing to check visually the correctness of the identifier assignment.

Link. The linking mechanism is used for binding related articles. There are several different types of article relations: continuation articles, derived work,

article review, suggested relevant papers, etc. Information about related articles might be suitably presented to users in the publication system. Currently, we use this feature to bind continuation articles only.

References. Reference processing is proposed to be semi-automated with the human operator intervention in fixing errors from OCR processing and marking the basic reference structure.

References are automatically identifed in the OCR fulltext using methods similar to those used in the CiteSeer project[2]. Simple markup characters are added by the system to the article OCR text (newline characters for identifying individual references within the block of references and '//' characters for marking borders between authors and title). In the next step, the result of automatic reference pattern detection is reviewed and corrected by the operator, if necessary.

Using these markups, a structured record of reference metadata is then generated. Finally, reference mathematical databases are queried to identify referenced articles and to establish hypertext links.

Although the reference processing is highly automated, its resulting quality heavily depends on the quality of OCR. We are labouring nontrivial effort to achieve a compromise between the quality and extent of manual interventions required by human operators.

Processing status. There are three different processing states assigned by the Metadata Editor to all objects (articles, issues, volumes, journals, etc.):

- **untouched** – (grey) object and all its nested items were just imported into ME internal structures and has not been edited yet;
- **in progress** – (red) object or at least one of its nested items were already edited;
- **completed** – (green) object and all its nested items were completed and checked by an operator; the object is prepared for PDF-generating step and export to the publication system.

The status is displayed as a colored bullet in front of the object title in the Metadata Editor.

2.4 Authority Base

The name authority base was introduced to handle author names ambiguities appearing in DML-CZ articles correctly. The concept of authority database is inspired by the one used in traditional library management systems. An authority database record consists of author personal data (at least in the extent sufficient to distinguish between persons with the same name) and a set of *name forms* appearing in articles in the DML-CZ.

[2] based on regular expressions for typical textual reference patterns

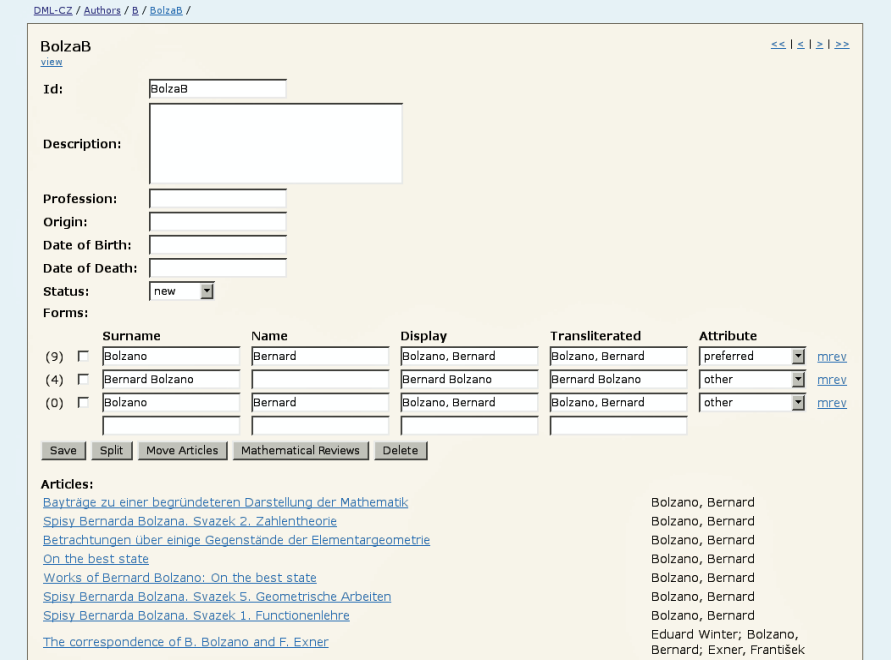

Fig. 4. Authority database management screen

In the article or in the reference metadata, we store a name in the identical form as it appears in the original printed document as well as an optional internal identifier of the corresponding authority record. This approach allow us to manage several scenarios:

- one person has several name forms (name with initials or full first names, pseudonym, transliterated forms of the name, etc.);
- two (or more) persons have the same name and we want to distinguish among them in granting correct article authorship;
- two (or more) persons have the same name, but we are uncertain who of them is the author of the article.

The name authority database collecting a broad spectrum of different name forms is taken into account in the publication system DML-CZ[3]. Searching for a particular name form results in displaying all articles written by the given author regardless of the author's name forms used in the articles.

This model can also be extended by detailed personal information (like curriculum vitae, photograph,...) for more famous authors.

Thanks to the name authority records, locating and correcting spelling errors in names is quite an easy task. Authority database is getting bigger and bigger

[3] or at least we are working on that

as the extent of the digital library grows continuously, so keeping authority database clean is a never ending process.

We also plan to connect our authority base to another mathematical authority databases like EuDML or WDML when they will be established to allow interchange of authority records.

2.5 Searching and batch update mechanism

The browsing capabilities of the Metadata Editor are supplemented by an easy to use searching module allowing operators quickly search for specified objects.

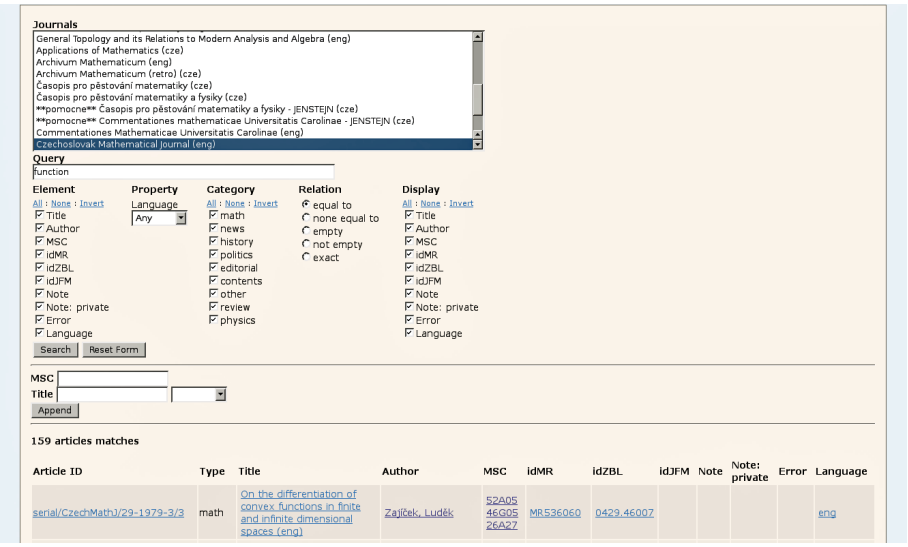

Fig. 5. Advanced search tool

The searching mechanism can be used to search for the specified term in the selected metadata record element(s), to search articles by specified language, document category, and so on.

It is possible to set one of the following relations between elements in a query:

- **equal to** – metadata records containing the search term somewhere in the selected elements,
- **none equal to** – none of the selected (possibly repetitive) elements contains the searched term,
- **exact** – value of the selected element equals exactly to the searched term,
- **empty** – metadata records with all the selected elements empty.
- **not empty** – metadata records where at least one of the selected elements is not empty.

The search tool can also be used for automated batch metadata update. When the search result set is displayed, privileged users can specify metadata element and a value that should be added or updated in all records in the result set.

This simple batch update tool addresses most of the Metadata Editor operator's needs without the necessity of the system administrator's intervention.

2.6 Automated metadata verification

To keep data consistent and of a high quality, we developed a powerful and extensible verificiation mechanism within the Metadata Editor with a set of useful tests. So far, the following verification tests were implemented:

- test for a missing mandatory metadata elements,
- data storage integrity tests (data completeness and coherence, XML validation,...),
- test of page ordering based on OCR data,
- test of article language based on OCR language detection,
- syntax of the TeX expressions used in metadata (titles),
- syntax of markups used for reference identification,
- statistics of the work progress (per individual document or per individual operator).

Each verification test consists of an executable plug-in and a formal description of input parameters and output format. Formal description is used to build appropriate user interface for particular verification test and to display results interactively. It is also possible to specify only a subset of documents to be verified. Each test can be executed directly from Metadata Editor or scheduled to run on a regular basis notifying system administrators by an e-mail. This feature is used to permanently (daily) monitor the data storage integrity.

2.7 Final PDF compilation

The final article PDF file consists of a cover page generated automatically using BibTeX and fine-tuned TeX style from descriptive metadata; and a set of two-layered PDF files representing individual pages generated previously by the OCR software.

In the next step, PDF file is optimized (by `pdfopt` tool) and digitally signed by a DML-CZ certificate authority (using `iText` library).

This PDF file together with all metadata are then imported into our publication system DML-CZ, based on D-Space repository system, as described in [2].

3 Technical Description

Our Metadata Editor is a three-tier web-based application based on Nitro (Ruby) framework. It uses a *Model-View-Controller* design pattern that clearly separates the visual appearance from the application logic and underlying data model (Figure 6).

As a primary data storage, a regular filesystem with a very simple and self-explanatory schema is used. Utilizing filesystem as an "API" turned out to be a great advantage during integration of the specialized external tools developed by independent programmers. Import/export scripts, reference linking tool, similarity searching algorithms [3], OCR, backup, etc. are all written in a real mixture of available programming technologies.

Snapshot of metadata indices are also stored in an internal MySQL database for effective metadata browsing in Metadata Editor. This approach combines the advantages of both storage technologies – flexible administration and quick access.

Application user interface is, due to its client-server architecture, accessible from anywhere using any web browser[4]. The network communication is encrypted using HTTPS and the proposed authorization model takes into account the user role with respect to accessed data collection. The changes are logged, that among others allows us to fairly reward hired students.

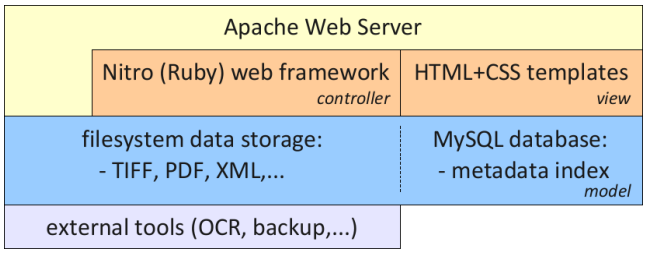

Fig. 6. System architecture of the Metadata Editor

The server is backed-up and monitored, in the case of break down or internal error the system developer is informed by e-mail.

4 Conclusions

During the building of a digital library, consisting of more than 200.000 pages and 15.000 articles, Metadata Editor has proven its unique and mature ability to effectively manage all labor intensive tasks. We have also optimized all painful

[4] with a special focus on *Mozilla Firefox* for which we provided keyboard shortcuts and some other special features.

bottle-necks of the system, so now all operations are quite responsive, even on a regular PC server (3GHz dual-core, 2GB RAM).

The architecture of the remote access with thin client and robust authorization schema enables us to distribute the work among several people with different levels of expertise. Students are hired to make all regular tasks such as input data verification, page assembly, article building, reference tagging and basic metadata completing. In second stage, mathematicians work on advanced tasks like translation of the titles to English, assigning missing MSC codes, authority base cleansing,... Apart these, there is a Metadata Editor supervisor assuring the metadata perfection, final checking, managing the whole process and distributing labour among operators.

We found that the overall time of article metadata processing depends heavily on several factors:

- article metadata coverage in referential databases
- internal complexity of document structures
- quality of scanned print-outs used for OCR (for references,..)

In general, the newer the material the better[5].

Our next plan is to release this software under an Open-Source license. In this moment ME is primarily designed to satisfy our needs, but during the oncoming final year of the DML-CZ project we want to generalize it for wider use and also make some minor improvements. In general, we need to finalize the shift from the set of home-grown tools to a production quality software suite, including installation package and a proper documentation. There is no doubt that this is a great challenge.

References

1. Bartošek, M., Lhoták, M., Rákosník, J., Sojka, P., Šárfy, M.: *DML-CZ: The Objectives and the First Steps.* In Borwein, J., Rocha, E.M., Rodrigues, J.F., eds.: CMDE 2006: Communicating Mathematics in the Digital Era. A. K. Peters, MA, USA (2008) 69-79.
2. Krejčíř, V.: *Building the Czech Digital Mathematics Library upon DSpace System*, (submitted to the workshop "Towards Digital Mathematics Library 2008").
3. Radim Řehůřek, Petr Sojka: *Classification of Multilingual Mathematical Papers in DML-CZ.* In: Proceedings of Recent Advances in Slavonic Natural Language Processing- RASLAN 2007, Karlova Studánka, Czech Republic, Masaryk University, Brno (2007) 89-96
4. Sojka, P.: *From Scanned Image to Knowledge Sharing.* In Tochtermann, K., Maurer, H., eds.: Proceedings of I-KNOW '05: Fifth International Conference on Knowledge Management, Graz, Austria, Know-Center in coop. with Graz Uni, Joanneumm Research and Springer Pub. Co. (2005) 664-672.
5. Petr Sojka, Radovan Panák, and Tomáš Mudrák: *Optical Character Recognition of Mathematical Texts in the DML-CZ Project.* June 2006. Submitted to "Communicating Mathematics in Digital Era, CMDE 2006", August 15-18, 2006. Aveiro, Portugal.

[5] Processing journals from 19th and beginning of 20th century is quite messy.