

ADEPT

Analysis and Development of Electronic Publishing Technologies Project
Unit for Scientific Information and Learning, KTH, Stockholm, and
Department of Computer and Information Science, Linköping University

Erik Sandewall

Support for Managing IPR and Parallel Publishing in
the MADMAN Research Author Support System

This series contains technical reports from the joint KTH/Linköping project
Analysis and Development of Electronic Publishing Technologies (ADEPT).

The present report can persistently be accessed as follows:

Memo persistent URL: <http://piex.publ.kth.se/reports/adept/003/>

Date of manuscript: 2009-01-17

Related information can be obtained through the following www sites:

KRF website: <http://piex.publ.kth.se/krf/>

The author: <http://www.ida.liu.se/~erisa/>

Purpose of the MADMAN System

The MADMAN system ("Management of Articles, Datasets, Messages And Notes") is a software system for assisting a researcher in those aspects of his or her work that concern the communication of research results in writing. In principle this covers a broad range of document types, from journal articles to e-mail messages and to grant applications and reports to sponsors. However at present the system addresses research articles, technical memoranda and webpages.

The need for an effective service of this kind is probably evident to every productive researcher that uses information technology for his or her writing. The MADMAN system is intended as a vehicle for experimenting with possible system designs and as a proof of concept; it is not a goal of the project to produce a software product. It is designed by, and actively used by the present author, but does not have any additional user community. However, parts of MADMAN have been incorporated into the web-based KEPS system (Knowledge Editing and Publication System) which is being made available to users on a trial basis.

The MADMAN system relies on a number of existing software tools, in particular, tools for text editing and formatting (Word, emacs + latex, etc.). It is designed in such a way that the user can continue to use these like he or she has done before. The following are the major services provided by MADMAN:

- Maintaining a database of bibliographic data and other metadata for each article, including both finished articles and those that are in various stages of preparation.
- Using these metadata for the formatting of the article.
- Managing alternative versions of an article, for example, the version submitted to a journal, the final version that was sent to the publisher, and the publisher's formatted version.
- Producing publication lists, both as webpages, as separate pdf documents, and for inclusion in larger documents.
- Delivery of articles to institutional archives and other websites.

An additional service which is specifically a topic of the present report, is the service of identifying, based on the metadata of a particular article, whether parallel publishing of this article is consistent with the IPR agreement of the journal where this article has been, or will be published. This service is integrated with some of those mentioned above, since it is common that publication contracts that do permit parallel publishing also impose requirements on the appearance of the article on the parallel publication site. These requirements may include a rule about which version of the article may be used; it may also include requirements about specific information that must be included in the text of the article, or in the webpage containing a link to the full text of the article.

The IPR management part of MADMAN is being developed as a part of the PARPUB project ⁽¹⁾ which is sponsored by the `OpenAccess.se` research

¹Domänmodellering av rättigheter och bivillkor vid parallellpublicering av

program ⁽²⁾ at the Swedish National Library (KB). Another part of the PARPUB project develops a server-based service for providing information about what IPR requirements apply for an article given its metadata. The MADMAN system will send inquiries to this server, although for testing purposes at present it keeps a preliminary version of the IPR database in its own computer.

Design Approach for the MADMAN System

MADMAN is not designed as a self-contained system containing all the required services. Instead, it assumes that the user will wish to continue, for example, the same text editor and the same ways (if any) of formatting text and of producing reference lists as he or she is already used to.

The perception is that MADMAN operates on an overarching level compared to those other softwares: it specifies layouts, administrates versions, specifies aspects of their graphic appearance, and so on, whereas other softwares are used for the more detailed work. Therefore, the connection between MADMAN and the jointly used softwares is implemented in two ways: through a set of conventions for the directory ("folder") structure and for the structure and naming of document files that must be used consistently, and through the possibility of invocation so that MADMAN is able to start up some of the other softwares with appropriate parameters.

Directory and file structure

The assumptions for the directory structure are as follows. The user is assumed to maintain one or a few directories that are dedicated to containing research articles and reports, including both finished ones and those that are in the preparation stage. Each such directory will be called an *article collection*. Each article collection has a number of subdirectories, one for each article, called *article directories*. The user may choose to have one single article collection for all articles, or he may introduce one collection for each of a few projects, or for each year, for example.

It is recommended to use short names for article collections and article directories, either a short mnemonic name or a number (e.g. "014" or "ms-014"), but not the full title of the article. In its database, MADMAN considers the names of article directories as database entities and associates a variety of information with it, including in particular a phrase that is used e.g. when displaying the contents of an article collection.

The main assumption about the structure of files in an article directory is the *separation of front matter from contents*. This means that there shall be one or more files that contain the contents of the article, but without the introductory matter such as the title, the author and affiliation information, and so forth. If the article is formatted with a front page, then no part of the front page shall be in the contents file(s). It is recommended to also maintain the abstract as a separate file.

vetenskapliga artiklar (*Domain modelling of rights and conditions for parallel publication of research articles*). Please see the project website at

<http://www.ida.liu.se/ext/parpub/>

²<http://www.kb.se/OpenAccess/>

The primary representation of the contents of the front matter are in the MADMAN database. It is used, together with the contents file(s), for generating the full article in one or more *variants*. For an article that has been published in a journal there is likely to be at least three variants: the variant that was submitted to the journal, the variant that was produced by the publisher and that appeared in the journal, and the variant that is to be used for parallel publication. For example, in some instances the third variant may be equal to the first variant except for the addition of bibliographic information about the journal version and other information that the journal publisher requires to be present there.

Notice that although different variants may have different contents, the primary distinction that is made here is the use of different graphical appearance. The notion of variants is to be distinguished from the notion of *versions* of the contents of the article, which result as the author(s) make successive changes in the manuscript as it is being written. MADMAN handles both versions and variants and they are independent concepts.

It is recommended, although not required, to use standardized names for the major files in an article directory. Component files may be called e.g. **abstract** for the file containing the abstract, and **body** for the file containing the contents (without the abstract) if the contents is just one file. Full article files may be called e.g. **submitted**, **revised**, **journal-variant** and **parpub-variant**, with the obvious meanings. The choice of these standardized names are specific to each installation or user. If the user wishes to use other names for these which are specific to each article, then the MADMAN database will contain metadata information specifying what e.g. the abstract file is called for this particular article. Regardless of how the naming is done, it is necessary to organize the files that represent the article along these lines, since MADMAN relies on, and operates on this structure.

The Appendix contains a detailed description of how these conventions can be implemented in the context of specific document preparation softwares.

Modes of usage

MADMAN offers a choice of usage modes in two dimensions: the type of text editor, and the use of command-line mode vs graphical mode. We shall discuss them in turn.

Mode of text editor

There are two alternatives for text editing mode: the direct editing of marked-up "source" text using a text editor such as **emacs**, and editing of the final graphical form of the text using a wysiwyg editor such as **Word** or **OpenOffice**. The **LaTeX** system is the only formatter of practical use when the former approach is used.

The initial implementation of MADMAN is for marked-up source mode. The corresponding implementation for **Word** and/or **OpenOffice** has not yet started.

When MADMAN is used in marked-up source mode, the standard method is to work with files with names such as **body.tex** and **abstract.tex** and

then to process them using **LaTeX**. However, there is an additional option of using a MADMAN-specific markup language called **TSL** for the contents files. There are generators from **TSL** to **LaTeX**, **HTML** and **RDF**, and **RDF** files can easily be imported into **Word**. The advantage of **TSL** over **LaTeX** format is that it makes it possible and convenient to include commands that fetch information from the MADMAN database and incorporate it into the document, both for single items of information and for tables.

Mode of access

There are three alternatives for access to the services of the MADMAN system:

- The user runs the MADMAN system on his or her computer, and operates it in command-line mode. Specific commands are used for invoking text editors, formatters, etc., and for updating database contents and for exporting documents to outside destinations.
- The user accesses the web-based subset of MADMAN, which is called **KEPS**. All information pertaining to articles is maintained on the **KEPS** server.
- The user runs the **KEPS** server on his own computer, which makes it possible to use the graphical interface locally.

The command-line mode and the local **KEPS** server can be used alternately since all information is located and represented in the same ways in both modes.

Preparation of reference lists

Several systems for the preparation of reference lists are in widespread use, and the choice of system may depend on the choice of text editor. For **LaTeX** users, **BibTeX** is the natural choice. For **Word** users, there are a number of commercial systems, such as **Endnote**.

At present, MADMAN supports the direct use of **BibTeX**. Metadata for the user's own documents can be exported from the MADMAN database to **BibTeX** files. It is also possible to store metadata for articles by others in the MADMAN database and use them for reference lists (via **BibTeX** as well as for other purposes).

Website generation

Besides the processing of the document files themselves, MADMAN also supports the creation of a static website structure consisting of **HTML** files together with the **PDF** files of the articles themselves. In addition, the **KEPS** system provides an implementation of similar services using dynamic webpages.

The pros and cons of the static versus the dynamic implementation are as follows. In the static implementation, **HTML** pages are generated once and for all and are copied to the webserver, and in the dynamic implementation

they are generated anew from the database contents each time there is a request from a website visitor. The dynamic implementation makes it possible to implement a wider set of services, including search facilities, but the disadvantage is that these services will only function as long as the particular implementation works on the server computer. This is a problem in the long-range perspective: what will be the possibility of using the KEPS server, or any other dynamic server for that matter, twenty years or fifty years from now? With static webpages using a minimal set of very basic HTML markup commands, it is likely that the structure can be effectively usable almost indefinitely.

The organization of MADMAN generated (or operated) websites is similar to the one used when manuscripts are prepared. It is primarily organized to serve the case of static webpages, but dynamic services can also access it. It is assumed that there is one *website article directory* for each article that is published, and that website article directories are grouped into collections. There is however no assumption that the name of the website directory for an article shall be the same as its name in the preparation stage; the article obtains a new "name" when it is added to the website structure.

A website article directory contains normally at least two files: an *index file* whose name is standardized to be `index.html`, and the PDF file for the full text of the article. Additional files may also be included, for example for an annex to the article, an errata page, an exchange of opinions about the article, or research data pertaining to the article.

An article index file, i.e. the index file of an article website directory, is intended to be used as a reference for the article. Users are therefore recommended not to link to the actual fulltext files (PDF) for an article, but only to link to the index file which in turn contains links to the full text. This is done so that whoever follows the link will obtain the fullest possible information about the article, and so that the full text can be moved to other locations if and when this should be desirable for some reason.

A systematic and well thought-out use of website article directories and article index files makes it unnecessary and redundant to use an additional layer of locators; the URL for the article index file is sufficient and it can be implemented with all the necessary flexibility.

The operations that MADMAN provides with respect to the website structure include in particular the generation of index files for website article directories, the generation of contents files containing a list of articles with links to their respective index files, and copying full-text files to their website article directory. MADMAN maintains a working copy of the website structure for its own use, and the copying of this information to the actual website structure may be made either using MADMAN commands or using other software for website maintenance.

Use of IPR information for parallel publication

The facilities for managing IPR and other contractual information for articles have been implemented using the structuring conventions that have now been described. There are two aspects to the IPR management (we use this as a brief term for the entire set of issues) for a specific article given its bibliographic metadata: (1) obtaining the IPR information that pertains

to the article and identifying its consequences for parallel publication; (2) implementing these consequences in the actual handling of the article.

For obtaining the IPR information, MADMAN will rely on the use of a server that is being developed in the PARPUB project. This web-based server receives bibliographic metadata for the article, including the ISSN number of the journal where the article is published and the month and year of publication, and furthermore the type of server (e.g. institutional, private, external but non-profit) and identifiers for the agency/agencies that sponsored the research. It responds with a composite, formal expression that specifies what consequences the client must observe, in our case, the MADMAN system.

These consequences are of several kinds. First of all, of course, there is the question whether parallel publication is consistent with the publication contract, and on which date. This will determine whether KEPS copies the full text of the article from the article directory to the website article directory. It also determines whether the article index page will contain a link to that full text.

Secondly, there is the consequence as to which variant of the article is to be used for parallel publication. Some publishers require it to be the publisher's variant, others require it to be the author's own variant and explicitly forbid using the publisher's variant. The KEPS system observes of course these conditions.

Third, there are often requirements of including particular information in the parallel-published variant of an article (typically, on its first page), or of including it in any surrounding webpage, which in our case will be the index page and possibly also in web pages containing a list of articles in a particular collection. The required information may include the identity of the journal and its publisher, the exact bibliographic information for the journal instance of the article, the DOI identifier, or a weblink to the location of the article on the publisher's website. It may also include a requirement for the use of specific phrases which are different for different publishers. These requirements are coded in the information that MADMAN receives from the PARPUB server.

The use of this information for generating website pages is straightforward. The index page of an article is regenerated when additional information comes in, for example, when the exact volume, issue and page number for the article in the journal become available.

As for the full text of the article, the MADMAN database may contain information about the URL (or other identifier) whereby the article is accessed at the publisher's website. This information can be used both for downloading a copy of the article to the article directory, and for inclusion in the index page if appropriate. If the publisher requires their own variant to be used for parallel publication then this is easily respected.

In the opposite case, where the author's own variant is to be used, it will be necessary to reformat the article in order to obtain a new variant that conforms to publisher requirements with respect to what bibliographic information is to be included on the first page of the article, or elsewhere in it. The separation of front matter from the contents of the article that was described above is important since it makes it easy to reformat in order to respect new requirements.

An interesting additional type of requirement occurs when a publisher permits prepublication of an article, but *requires* that the prepublished variant shall be removed at the time of publication in the journal and replaced by a final author's variant containing linking information to the publisher's instance of the article. This means that the transfer from the local copy of the website structure in the MADMAN system, to the target website, can not merely consist of copy-and-overwrite; it must also accommodate the replacement requirement (at least if it is supposed that the preprint variant and the postprint variant shall be distinguished by different filenames, which would be appropriate but not necessary).

The use of the DSL scripting language

An important part of the implementation of MADMAN is to arrange for the automatic generation of several types of files: HTML files for index pages, LaTeX files for the front-matter part of each variant of an article, and so forth. For these purposes MADMAN uses a scripting language called DSL (Document Scripting Language) which at present is set up to generate HTML code and LaTeX code; other target languages are forthcoming. DSL is highly readable and highly flexible due to its procedural character, and makes it convenient to combine fixed text with information obtained from a database. DSL is closely connected to TSL (Text Scripting Language) which was mentioned above as an optional facility for the document preparation function.

The customization of the MADMAN system to a particular user or group of users will often require the design of specific webpage layouts, for example for preparing lists of articles in a particular way. The DSL language is used for such customization. DSL is likewise used for defining the special additions to article fulltext files that are required by specific publishers. The DSL/TSL language combination is documented in a separate report which is referenced below.

References

1. Erik Sandewall: OA-publicerade domänmodeller avseende vetenskaplig publicering och gruppstruktur. Organizational Memo number 4, Division of Publication Infrastructure, KTH, Stockholm.
<http://piex.publ.kth.se/reports/rapp/004/>
2. Preben Hansen, Gunnar Eriksson and Oscar Täckström: Steps towards automatic acquisition and recognition of IPR conditions for parallel publishing. In preparation as a report from SICS.
3. Erik Sandewall. Delrapport för OpenAccess.se-projektet Domänmodellering av rättigheter och bivillkor vid parallellpublicering av vetenskapliga artiklar.
<http://piex.publ.kth.se/reports/rapp/007/>
4. Erik Sandewall. KRF Scripting Languages for Documents and Texts.
<http://piex.publ.kth.se/reports/krf/007/>