MathML-aware article conversion from LaTeX. A comparison study.

Heinrich Stamerjohanns, Deyan Ginev, Catalin David, Dimitar Misev,
Vladimir Zamdzhiev, Michael Kohlhase

Computer Science, Jacobs University Bremen
<first initial>.<last name>@jacobs-university.de

Abstract. Publishing in Mathematics and theoretical areas in Computer Science and Physics has been predominantly using TeX/LaTeX as a formatting language in the last two decades. This large corpus of born-digital material is both a boon — LaTeX is semi-semantic format where the source often contains indications of the author's intentions — and a problem — TeX is Turing-complete and authors use this freedom to use thousands of styles and millions of user macros.
Several tools have been developed to convert TeX/LaTeX documents to XML-based — i.e. Web and DML-compatible formats. Different DML Projects use different tools, and the selection seems largely accidental. To put the choice of converters for DML projects onto a more solid footing and to encourage competition and feature convergence we survey the market. In this paper we investigate and compare five LaTeX-to-XML transformers in three dimensions: *a*) ergonomic factors like documentation, ease of installation, *b*) coverage, and *c*) quality of the resulting documents (in particular the MathML parts).

1 Introduction

Publishing in Mathematics and theoretical areas in Computer Science and Physics has been predominantly using TeX/LaTeX as a formatting language in the last two decades. This large corpus of born-digital material is both a boon — LaTeX is semi-semantic format where the source often contains indications of the author's intentions — and a problem — TeX is Turing-complete and authors use this freedom to use thousands of styles and millions of user macros.

On the other hand there is a growing effort to make mathematical publications available on the Internet in formats that are more adapted to the Web than PostScript or PDF (which can readily be produced from TeX/LaTeX by standard tools). Even though there are competitors, MathML [ABC+03] seems to be the format of choice for rich mathematical content on the web, because it supports high-level services like aural rendering (e.g. in Internet Explorer with MathPlayer) or formula search [MM06,KŞ06].

Several tools have been developed to convert TeX/LaTeX documents to XML+MathML-based — i.e. Web and DML-compatible formats. Some tools

use the TEX engine to parse the original LATEX source, other tools try to reimplement a complete TEX parser to have full control over the document processing. Different DML Projects use different tools, and the selection seems largely accidental or governed by personal acquaintances. To name just two: the arXMLiv project [SK08,arX] at Jacobs University uses LaTeXML [Mil09] for converting the Cornell ePrint arXiv [ArX07] whereas the NUMDAM and CEDRAM projects [Bou08] use Tralics [Tra09].

In this paper we try to put the choice of converters for DML projects onto a more solid footing and to encourage competition and feature convergence by surveying the LATEX to XML+MathML transformation market and comparing five available systems. In Section 2 we briefly present the five tested systems and compare them on the ergonomic factors like documentation, ease of installation. In section 3 we investigate coverage, and quality of the resulting documents (in particular the MathML parts) on a corpus of sample scientific articles from the arXiv ePrint server [ArX07].

This paper is short version of [SGD+09] which contains additional details, will be kept up to date with new versions of the converter and will feature extended tests, system updates and further systems.

We are utilizing our experience, resources, and parts of the build system from the arXMLiv [SK08] project for these tests, but we are trying our best to give a neutral representation of the systems studied here and to avoid biases in the presentation. If we have misrepresented any parts of the systems, please feel free to contact the authors.

2 The Systems

In this paper we investigate and compare five LATEX-to-XML transformers that generate MathML output: Hermes, Tralics, LaTeXML, TeX4HT and TtM.

In addition to these five systems we have found other TEX-to-XML translators (see [WG09] for a relatively complete list and references to system homepages) which we could not not include in our comparison for various reasons.

1. the `BlaTeX`, `itex2mml`, `RiTeX`[1], MathMLStudio Lite only convert a subset of TEX/LATEX formulae to MathML, but do not seem to have a document level. They are more directed towards authors of mathematical documents on the web rather than born digital DML efforts. Therefore we have not included them into the current study. But our comparison methods should apply to them as well, so we may include them in future studies.
2. The `HeVeA`, and `LaTeX2html`, transform LaTeX documents to HTML, but do not seem to generate MathML output (only images of formulae).
3. The University of Western Ontario offers an online service [Wat09] to translate documents into MathML. The software is supposedly written in Java. We have asked for the code but have not received an answer.

[1] Not yet on [WG09], but see `http://ritex.rubyforge.org/`

4. `LXir`[Sci09] has been written by EDP Sciences under the GPL. It claims to to transform LaTeX to XML. The authors have been unable to compile the software, and since detailed instructions are only available in French, we also left this tool out.
5. `Omega`[PH] has been discontinued and seems to be merged into `LuaTeX` which is supposed to be an extended version of pdfTeX written in Lua. The project is at an early stage and aims to support the OpenType math of Microsoft.

In the following we will briefly give an overview over the systems, their state of development, their approach to the conversion problem and deployment and discuss installation and usability issues that may play a role in making a decision for a DML project.

Hermes Hermes is a grammar based translator from (AMS)LaTeX to Unicode(utf-8) encoded XML+MathML+metadata, however transformation of pure (AMS)TeX documents is not supported. MathML is the only valid XML vocabulary and is also the only output format implemented and supported currently by Hermes.

The system is available for download from Hermes' official site[Ang09b]. It is licensed under GNU GPL and is easy and straightforward to install from source. Hermes works on Linux, Windows and OS X. However, the latest version of the transformer 0.9.12 was released on 28 Nov. 2006 and development on it has been discontinued. The documentation that is available for Hermes is very scarce and contains only a very brief description of how Hermes works and its usage, installation requirements and a short description of the Hermes output document.

Hermes works, in theory, by semantically seeding a copy of the TeX source, then uses `latex` on it and parses the resulting `dvi` file to form the MathML output, which is a reflection of the TeX source. In practice this is achieved by using a binary `seed` that is obtained after compiling the source, using `latex` on the semantically enriched file, followed by using the second binary `hermes` and finally using an `xslt` style sheet. One stylesheet is available after the installation, but others can be alternatively used. The command-line interface is very user-unfriendly, no options can be specified to customize/control the conversion (not even a help option is available).

Hermes does not produce any information during the conversion except for the log produced by LaTeX, which however does not indicate whether or not the conversion of the file was ultimately successful as it can fail at the later stages. The conversion of a large number of files is impractical, because of the absence of appropriate log files which makes the task of evaluating the conversion harder, since the observations have to be based on other criteria.

The conversion is slow mostly due to the use of LaTeX, the Hermes binaries however perform very fast in most of the cases.

Tralics Tralics is designed to translate LaTeX sources into a custom XML representation with an outlook for a successive conversion to PDF or HTML. The software is licensed under CeCILL[CeC09], which is a GPL-like license, conformant with French law in particular. The original target of Tralics is

the conversion of annual activity reports[Gri03], with an outlook for ease of customization via configuration files.

The software is readily available online[Tra09] and is deployable both from source or a respective binary for either Linux, Mac OS or Windows. There is a separation between "main" and "extra" functionality, where the "main" package provides the conversion to XML, while the post-processing to HTML and PDF resides in an additional bundle.

An extensive documentation regarding use and customization is directly accessible online. The information is conveyed from a developer perspective, describing customization and extensibility. However, there could be more precise guidelines regarding the usability of TRALICS, building on the currently vague overview of the different option switches. As for the "extra" bundle, the auxiliary post-processing stylesheets are explained from a low-level perspective, yet no solid high-level use cases are given, making them hard to use out of the box. In its current form, the extensive documentation would be of great help to developers who are willing to extend or customize the processing, but it could be improved further to help users interested in using TRALICS.

TRALICS uses the TEX parser to expand the document recursively, stopping when the pages have been constructed. Consequently, the C++ engine constructs the XML document tree and converts the mathematics to MATHML, also integrating bibTeX in the resulting XML document. The conversion to the custom XML format supports MathML as a default representation for mathematics and can output images as an alternative.

LaTeXML LATEXML has been developed to support the creation of the *Digital Library of Mathematical Functions*[DLM09]. It is written in Perl and tries to emulate TEX. An additional post-processor converts the XML document into HTML or XHTML with MathML support.

LATEXML is freely available online[Mil09] and can be installed as a package or from source on Linux or Mac OS systems. Since it is written Perl it should run on Windows, but so far it seems that nobody has actually tried to install it on that platform.

A very detailed manual is either available online or as a 130-page PDF document. It provides detailed information about the architecture of this TEX emulator, as well as detailed chapters about customization, command usage and post-processing.

The LATEXML system consists of a TEX emulator, an XML emitter, and a post-processor. To cope with LATEX documents, the system needs to supply LATEXML bindings (special directives for the XML emitter) for the semantic macros in LATEX packages.

For the XML conversion, `latexml` processes a TEX or LATEX document. `latexml` loads the LATEXML bindings for the LATEX packages used in the document and generates a temporary LTXML document, which closely mimics the structure of the parse tree of the LATEX source. The LTXML format provides XML counterparts of all core TEX/LATEX concepts, serves as a target format

for LaTeXML, and thus legitimizes the XML fragments in the LaTeXML bindings.

In the post-processing phase, the LaTeX-near representation is transformed into the target format by the `latexmlpost` program. This program applies a pipeline of intelligent filters to its input. The LaTeXML program supplies various filters, e.g. for processing HTML tables, including graphics, or converting formulae to Presentation-MathML. Other filters like transformation to OpenMath and Content-MathML are currently under development.

The speed of the conversion is rather slow, for each document a new process needs to reload all needed bindings and perl modules.

TeX4HT TeX4HT is one of the TeX based transforming system which is mainly dedicated to output hypertext, though not restricted to that. Actually, one of the most interesting characteristics of the TeX4HT system is the multitude of output formats this system supports.

The system is available online on the official website[Tex09]. This is the recommended place to get it for installation, though it's also possible to install it from the repositories of some major Linux distributions (Ubuntu, Fedora).

In order to transform a file, the system modifies and compiles the .tex file with LaTeX, then runs the actual TeX4HT components in order to output a .xml file which corresponds to the MathML and XHTML standards. By doing that, the system actually supports all the LaTeX constructs that are available on the system and does not require any further bindings. The documentation is available on the website and is simple and well organized. Being neither overly technical, nor shallow, it provides ease of access and understanding to any type of user.

Biggest drawback of TeX4HT is the lack of debugging support. First of all, by running LaTeX, the system is highly dependent on the status of the LaTeX parsing and output, which makes batch testing rather difficult (in case LaTeX hits an error, the entire conversion process would stop). Even if the error in the LaTeX source would get corrected, the output still has chances of being broken (invalid XML is the most common, since TeX4HT does not guarantee that the output is correct and suggests an external XML validator for checking that). The log that is created corresponds to the LaTeX log file and is completed only by a small amount of information about the actual TeX4HT conversion.

Regarding the speed of the transformation process, the results are hindered because of LaTeX which is ran on a file three times, process which takes a lot of time, while the actual TeX4HT script is written in C, giving very good conversion times.

TtM TtM is a TeX to MathML translator that has essentially all of the capabilities of the TeX to HTML translator TtH, since it derives from the same code base. It supports all the complexities of TeX except for some features that do not translate readily into HTML. In most cases, TeX and LaTeX documents that conform to the appropriate standards will translate immediately.

TtM is available for Windows and Linux platforms (only the Linux version being free) online[TtM09].

TtM does not call the LaTeX or TeX programs at all by default (instead it tries to imitate how they work), and is not specifically dependent upon any other programs being installed on the translating system. Its portability is therefore virtually universal, and installation is as simple as extracting an archive. TtM is written using the flex language, from which a C executable is produced, making it extremely fast in default mode. Conversion of even large TEX files is a matter of a second or two, which makes it very suitable for use in an online script to output HTML directly from TEX source. A very well structured documentation comes bundled in the distribution archive. The manual is rather short and simple but covers pretty much all aspects of TtM, and is very easy to follow even for the not so experienced users.

Almost all of TeX's mathematics is supported with the exception of a few obscure symbols that are absent from the fonts normally available to browsers. LaTeX support includes essentially all mathematics plus most of the vital LaTeX constructs. Although macro definitions are fully supported, TtM does not understand TeX category codes (catcodes), therefore it will not work for some low-level TeX/LaTeX. In general, TtM will perform great on fairly vanilla TeX/LaTeX, but it will fail if many unusual packages or style files are being used.

3 Coverage and Processing Speed

In this section we will evaluate the coverage of the converters. For this we chose the arXiv corpus, since it can be considered as one of the most comprehensive sources of heterogeneous TeX/LaTeX documents. It contains more than a half million of scientific papers from fields including Physics, Mathematics, (computational) Biology, and Computer Science from two decades. In our experience, the corpus gives a good cross-section over the TeX/LaTeX in the wild. Of course we cannot run the converters over the whole corpus (a complete run of LaTeXML is in the order of a processor-year), therefore we have chosen a random sample of 1000 documents.

Addressing coverage of converters is not straightforward, since there are different degrees of failure and reasons for them. Following our analysis for the arXMLiv project [arX] we will concentrate on the three error classes that can be established for all systems.

incomplete The converter did not complete the conversion and crashed or signalled a fatal error. For some system we can identify subclasses like "timeout" or "fatal error".

complete with errors The converter completed conversions, but signalled errors. Some system give more indications on what the errors might be, a common one would be "missing macros", where some style file could not be processed.

success The converter completed the conversion with no problems or only "warnings" (i.e. problems the converter classifies as minor).

Note that the assessment in this section only relies on the problems reported by the systems themselves — apart from processing speed and system crashes that can be measured objectively. Of course system diagnostics may be inaccurate; for instance a system may report success and only produce an empty result file. But for the large number of documents needed for statistical validity in the coverage tests, we cannot run manual quality controls. We have made quality evaluations for the generated MathML on a sample in the next section.

Hermes Coverage		
Results	Count	%
incomplete	653	65.3
with errors	0	0
success	347	34.7

The statistics were obtained by writing a script to run Hermes over all of the files. 65.3% of the conversions resulted in producing an empty output file, while 34.7% of them were considered successful. Hermes has very poor debugging features so the statistics are not based on Hermes's report, but by running Validator[Val09] on the generated output. According to Validator[Val09], all of the files that Hermes produced which are not empty are well-formed. In most cases, after running latex on the seeded copy of the TEX file there are many warnings and some errors produced by it, which is overcome by running latex in batch mode, and out of the 653 unsuccessful conversions, 362 failed at this point. Running Hermes on all of the files took approximately 20 minutes.

Tralics Coverage		
Results	Count	%
incomplete	0	0
with errors	984	98.4
success	16	1.6

Due to its originally narrow scope of coverage, Tralics encountered a substantial amount of undefined command sequences, giving it a success rate of only about two percent. Considering that the conversion is based on configuration files that define the XML translation of the different LATEX commands, using it to process a general selection of scientific LATEX articles could not lead to an immediate success. Many of the documents of the arXiv corpus use specialized packages and classes, either imposed by community standards, or as personal convenience for macro definitions. Tralics met 50 undefined commands on the average document, being properly setup and aware of its standard configuration files. On the other hand, the conversion was performed in only three minutes and produced an XML output for each input article, regardless of the encountered errors. A check for well-formedness showed 93% of the files were proper XML, hence even an erroneous conversion could be potentially utilized. For such batch jobs, one can choose to run in a verbose or quiet mode, to save the details in a log file and whether to use MathML for trivial math formulas, to name a few useful options.

LaTeXML Coverage		
Results	Count	%
incomplete	103	10.3
with errors	357	35.7
success	540	54.0

It has to be mentioned that LaTeXML has an advantage here. Because of the ongoing arXMLiv effort many binding files have been written to support specifically style files that are typically found in scientific articles. Therefore LaTeXML does a nice job, 89% of the documents have been successfully created as XHTML. However for 35.7 % of the documents LaTeXML cannot guarantee

that the XHTML will be rendered fully correctly. Most of these converted documents rather have problems with layout oriented packages, which is irrelevant for XML.

The time to process these 1000 documents is rather long: more than 1.5 hours are needed to convert all documents.

TeX4HT Coverage		
Results	Count	%
incomplete	414	34.33
with errors	332	27.53
success	450	38.14

As mentioned before, the TeX4HT system should be versatile because it actually runs LaTeX on the files and only afterwards calls the actual transformation scripts that work on the DVI and lg files. However, LaTeX is called three times in order to ensure that the references are correct, action that, in the batch testing that we made, takes a lot of time and, because of errors, has the tendency to interrupt the actual conversion process. This issue has been overcome by using a BASH script that sets a timeout for the conversion of each file. One important note about the TeX4HT system is that, in case of large files, it splits the actual file in multiple files (thing which was nowhere to be found in the documentation), thus resulting a larger number of files than in the other systems that were tested.

Running the system over all the documents, with a timeout of 60s for the transformation of each individual file lasted, on average, for 90 minutes (result mainly influenced by the LaTeX parsing).

TtM Coverage		
Results	Count	%
incomplete	270	27
with errors	650	65
success	80	8

In 40 seconds TtM managed to successfully convert only 8% of the 1000 files. The low success rate is mainly due to the fact that TtM doesn't understand category codes and \usepackage, which makes it unsuitable for converting general scientific papers.

This run clearly shows that TtM is an extremely fast tool which needs about 0.04 seconds on average to convert a TeX file to XHTML+MathML. On average there were 100 warnings per file, most of which related to converting unknown commands/environments, unknown bibitems and missing bibcites. 651 errors in total were reported, 30 of which fatal (fatal errors result in an immediate termination of the conversion producing an incomplete XHTML output).

4 MathML Quality Evaluation

In this section we will evaluate the quality of MathML formulae generated by the five tested converters. The overall test methodology is to establsh a "Formula Quality Test Corpus" (FQC) consisting of a small set of non-trivial formulae randomly chosen from the arXMLiv corpus.

Here we only have space to report the highlights of this evaluation on a single example, the full results can be found in Appendices **A** to **E** of [SGD+09]. Generally, the quality of the generated XML is judged in terms of the XHTML+MathML quality i.e. CSS usage, presentation vs content MathML,

$$
\begin{aligned}
4r^2 \int_0^{\pi/2} \cos^2\theta \, d\theta &= 4r^2 \int_0^{\pi/2} \frac{1}{2}(1+\cos 2\theta)\, d\theta \\
&= 2r^2\theta \Big|_0^{\pi/2} + 2r^2 \int_0^{\pi/2} \cos 2\theta \, d\theta \\
&= \pi r^2 + 2r^2(\sin 2\theta) \Big|_0^{\pi/2} \\
&= \pi r^2
\end{aligned}
$$

Fig. 1. An example formula to check the quality of the converters

and formula tree quality. In particular addressing questions like: Is the resulting XML valid? Can we revert back to LaTeX? Are formulae like $x + y^2$ disambiguated, etc.?

4.1 The Eqnarray* Environment

Here we examine particular features of the LaTeX to MathML conversion of the example in Fig.1, which either make a rendering difference or reveal an interesting design choice of the conversion.

Representing Eqnarray As a multi-line equation environment, `eqnarray` requires a table representation in XHTML. The solutions here vary from using a MathML ⟨mtable⟩ (Tralics,TeX4HT,TtM), through using only HTML ⟨table⟩, each cell of which has a separate math construct (LaTeXML), to a combination of both (Hermes). Using only ⟨mtable⟩ allows to obtain an equivalent mathematical fragment to the original and should be the long term goal of all converters. However, pragmatic Al reasons, such as missing browser support for references (via the ⟨mlabelledtr⟩ element), justify the HTML ⟨table⟩ approach, in the context of the current state of art. The mixed solution employed by Hermes, however, embodies the worst of both approaches with an additionally crippled alignment of the rendering.

Operators and Symbols To represent mathematical operators, the converters start with a plain use of the ⟨mo⟩ element (Hermes) and enhance this representation with additional attributes, whenever possible. TeX4HT uses the "class" attribute to achieve a better rendering of the symbols, while Tralics uses the "form" attribute for their better positioning. Additionally, LaTeXML makes use of "movablelimits", to achieve a deterministic rendering of scripts. It is interesting that every converter has come with its own disjoint enhancement, hence giving an outlook for an improved rendering by making use of different attribute combinations, as each attribute contributes to the quality of display.

Math spaces Spacing seems to be a delicate issue to most converters. Hermes and LaTeXML forget any spacing information when converting to XHTML,

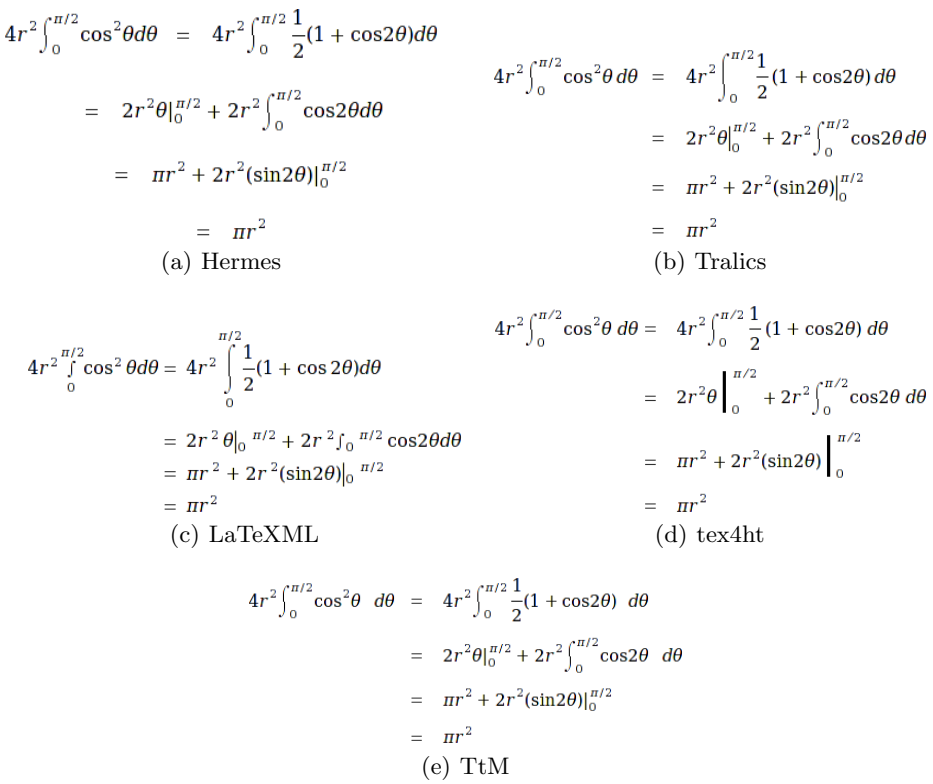

Fig. 2. Rendered representation of produced output in firefox

which leads to a very poor display of densely stacked symbols. However, LaTeXML tries to avoid this by adding its own heuristics to the processing, adding an &ApplyFunctions; math operator, wherever it considers appropriate. This leads to a satisfactory result when guessed correctly, but being a semantic enhancement, it is not always present when needed. This leaves an impression of an inconsistent LaTeXML rendering of spacing. On the other hand, Tralics and TeX4HT sharply preserve the spacing from source, in the form of ⟨mspace⟩ elements. TtM uses a more generic approach to math spacing, using only ⟨mi⟩ ⟨/mi⟩.

The | symbol We have the chance to analyze a symbol that seems to be a mystery for all discussed converters in terms of matching its LaTeX representation in MathML. The common solution is to utilize an ⟨mo⟩ element with a following ⟨msubsup⟩, which is, however, a completely different concept, and hence rendering, to the one of the LaTeX original. The only alternative solution comes from TeX4HT, where | is an "open" attribute to an ⟨mfenced⟩ element. Additionally, this element is wrapped with ⟨mstyle⟩, adjusting the size of | using the "mathsize" attribute. This lands closest to the originally produced LaTeX display, yet it is still not a perfect solution.

Integrals As standard mathematical constructs, integrals receive a largely similar representation among the converters. Using a ⟨mi⟩∫⟨/mi⟩ for the symbol itself is uniformly adopted, followed by a ⟨msubsup⟩ element encapsulating its range. Only LaTeXML makes use of ⟨munderover⟩ to capture the range, yet this differs from the original LaTeX rendering and is additionally causing an inconsistently bad rendering of the integral symbol.

Noise A problem common to all converters is the creation of empty elements such as ⟨mrow/⟩,⟨mo/⟩, ⟨mo⟩ ⟨/mo⟩ etc. On average, 10 or more of these noisy bits and pieces occur in each converted file, whether due to lack of coverage or erroneous processing. What we want to highlight is that such empty elements clutter the conversion and harm follow-up applications, such as search. To avoid this, converters should evolve to be aware of such noise and remove it in due time.

5 Conclusions, Recommendations and Further Work

In this paper try to support the DML community in raising the treasure of born-digital materials encoded in TeX/LaTeX by bringing order into the zoo of LaTeX-to-XML transformers. Concretely, we have studied five programs that transform TeX/LaTeX sources into XML and have the option to create MathML. We compared the systems in three dimensions: *a*) ergonomic factors like documentation, ease of installation, *b*) coverage, and *c*) quality of the resulting documents (in particular the MathML parts). To obtain an objective measure of *b*) and *c*), we tested all systems on a set of 1000 articles randomly picked from the arXiv ePrint server. The result can be summarized in Figure 3.

System	Documentation	Installation	**Coverage**	% incomplete	% with errors	% success	Quality	Speed	Ease of Use	Extra
Hermes	–	++	–	65	0	35	–	o	–	–
Tralics	+	++	–	0	98	2	o	+	– –	o
LaTeXML	++	+	**+**	10	36	54	+	–	+	+
TeX4HT	++	++	**o**	34	28	38	++	–	++	++
TtM	++	++	–	27	65	8	o	++	+	–

Fig. 3. Comparison Table for the systems

We hope that our survey helps put the choice of converters for DML projects onto a more solid footing and also encourages competition, collaboration, and feature convergence — what project wants to have – – marks in Table 3.

Of course the work reported in this paper is just a beginning, we need to incorporate feedback from the author's of the systems, and extend it to incorporate more systems, in particular the formula-only transformers `BlahTeX`, `itex2mml`, `RiTeX`, MathMLStudio Lite mentioned in in the introduction to section 2.

References

[ABC+03] Ron Ausbrooks et al. Mathematical Markup Language (MathML) version 2.0 (second edition). W3C recommendation, World Wide Web Consortium, 2003.

[Ang09a] Romeo Anghelache. Hermes discontinued. project page at `http://humanist.roua.org/2009/01/01/hermes-paused/`, seen May 2009.

[Ang09b] Romeo Anghelache. Hermes website. project page at `http://hermes.roua.org/`, seen May 2009.

[arX] arxmliv build system. `http://arxmliv.kwarc.info`.

[ArX07] `arXiv.org` e-Print archive, seen December2007. web page at `http://www.arxiv.org`.

[Bou08] Thierry Bouche. Cedrics: When cedram meets tralics. In Petr Sojka, editor, *Towards Digital Mathematics Library, Proceedings of the DML 2008 workshop*, pages 153–165. Masaryk University, Brno, 2008.

[CeC09] Cecill license. `http://www.cecill.info/`, seen May 2009.

[DLM09] Digital library of mathematical functions. project page at `http://dlmf.nist.gov/`, seen May 2009.

[Gri03] Jose Grimm. Tralics, a latex to xml translator, 2003.

[KŞ06] Michael Kohlhase and Ioan Şucan. A search engine for mathematical formulae. In Tetsuo Ida, Jacques Calmet, and Dongming Wang, editors, *Proceedings of Artificial Intelligence and Symbolic Computation, AISC'2006*, number 4120 in LNAI, pages 241–253. Springer Verlag, 2006.

[Mil09] Bruce Miller. LaTeXML website. `http://dlmf.nist.gov/LaTeXML/`, seen May 2009.

[MM06] Rajesh Munavalli and Robert Miner. Mathfind: a math-aware search engine. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 735–735, New York, NY, USA, 2006. ACM Press.

[PH] Plaice and Yannis Haralambous. Omega website.

[Sci09] EDP Sciences. lxir website. `http://www.lxir-latex.org/`, seen May 2009.

[SGD+09] Heinrich Stamerjohanns, Deyan Ginev, Catalin David, Dimitar Misev, Vladimir Zamdzhiev, and Michael Kohlhase. A comparison study of mathml-aware LaTeX converters. Kwarc report, Jacobs University Bremen, 2009.

[SK08] Heinrich Stamerjohanns and Michael Kohlhase. Transforming the arχiv to xml. In Serge Autexier et al., editors, *Intelligent Computer Mathematics, 9th International Conference, MKM 2008 Birmingham, UK, July 28 - August 1, 2008, Proceedings*, number 5144 in LNAI, pages 574–582. Springer Verlag, 2008.

[Tex09] TeX4HT website. `http://www.cse.ohio-state.edu/~gurari/TeX4ht/`, seen May 2009.

[Tra09] Tralics website. `http://www-sop.inria.fr/miaou/tralics/`, seen May 2009.

[TtM09] TtM website. project page at `http://hutchinson.belmont.ma.us/tth/mml/`, seen May 2009.

[Val09] Validator website. `http://homepage.mac.com/rcrews/software/validator/`, seen May 2009.

[Wat09] Stephen Watt. Mathml at ORCCA. project page at `http://www.orcca.on.ca/MathML/`, seen May 2009.

[WG09] W3C Math WG. Mathml software - converters. `http://www.w3.org/Math/Software/mathml_software_cat_converters.html`, seen May 2009.