

Conference logistics

- The **conference location** is DeBartolo Hall on the campus of the University of Notre Dame near South Bend, Indiana. Note that there is also a DeBartolo Center for Performing Arts in the same area of campus (south side). DeBartolo Hall is building 1144 on the campus map and is just west of the football stadium and just north of Mendoza College of Business. It is east across the quad from McKenna Hall.

The conference sessions will take place in Room 126 DeBartolo and the workshops will be taught in 118 and 119 DeBartolo (first floor). Conference breaks will be outside Room 126.

- **Registration** will take place in McKenna Hall (the Center for Continuing Education) in the main lobby. McKenna Hall is on Notre Dame Avenue east across the driveway from the Morris Inn. At registration, participants will receive a meal card for the South Dining Hall, instructions for Internet access, directions for obtaining parking tokens if driving to campus, and program booklets. McKenna Hall Information Desk phone number is 574-631-6691.
- **Internet access:** The guest account you will receive is *only* valid to connect your personal computer and/or handheld device to the `nd_guest` wireless network. That guest account will not allow you to log into any lab computers. It is also not valid for any classroom, lectern, or presentation computers; it doesn't work on any hallway email kiosk stations; you will not be able to print with it. Guests at the Morris Inn will also have Internet access.
- **Driving directions** to the Morris Inn and campus parking from the intersection of 933 (Michigan Street) and Angela Boulevard: Turn east on Angela Boulevard and go to the first traffic signal. This is Notre Dame Avenue. Turn left and look up. If you see the Golden Dome, you're there. At the four-way stop, Morris Inn guests should proceed straight and the Inn will be on your left, almost to the head of the avenue. If you go too far, you'll enter a roundabout and return to the Inn immediately on your right. Parking is free to Morris Inn guests in the adjoining lot on the south side of the Inn.

Participants driving to campus should turn right at the four-way stop. Drive past the huge building on your right (DeBartolo Performing Arts Center). Note that there is a stop sign directly in front of DPAC. Proceed to the next four-way stop, and your lot is just past the stop on the right. This is a gated access lot which requires \$2 to exit. However, parking will be free if you pick up a token at the McKenna Hall registration desk.

- An **opening reception** will take place Tuesday, July 28, from 5–7 p.m. in the Irish Courtyard, a large white tent near the Morris Inn. Hors d'oeuvres and nonalcoholic beverages will be served. Please check in at the registration table, pick up your name tag, conference booklet, and other items. If you're not able to attend the reception, you can register on Wednesday morning at 8 am at the McKenna Hall registration desk.

TUG members meeting

After the regular session on Thursday, we will hold a TUG user group meeting for anyone interested. Several TUG board members will be present. We will report on TUG's current status and future outlook.

We invite discussion of any TUG-related business at this time: ideas for outreach to additional communities, additional initiatives to undertake, existing projects to support, or any other user group-related topics.

Banquet & soapbox

The conference banquet will be held in the Donors Room of the Morris Inn at 7 pm on Thursday, July 30. We will have a few door prizes at the banquet (as usual). If you haven't signed up for the banquet, it's not too late. Just let us know (email tug2009@tug.org).

We will hold a 66–99 second soapbox at the banquet, where anyone can speak for a minimum of 66 seconds and a maximum of 99 seconds on a T_EX-related topic:

- You can report a success, gripe about a problem, lament a failure, share an insight, ask a question, or explain a solution.
- No intros, no questions, no hacking on earlier speakers; just you, the mike, and the audience . . .
- . . . and a moderator with a timer who will cut you off when your 99 seconds are up.
- No slides, overheads, whiteboards, blackboards, flipcharts, chalk, markers, or other props.
- Come prepared or make it up on the spur of the moment — no experience necessary.

Tim Arnold

Getting started with plasTeX

We discuss plasTeX, a software package for converting L^AT_EX documents to other markup languages. We will cover typical usage with examples of how to create HTML and DocBook XML from L^AT_EX sources.

We finish with a demonstration of converting a simple L^AT_EX source file and a brief overview of how to extend the package to handle custom commands and environments.

Kaveh Bazargan

TeX as an ebook reader

An important advantage of ebook readers is their ability to modify text size and page orientation, for the most comfortable reading configuration. The ebook reader has to reformat the text on the fly and with minimum delay. Current ebook readers (e.g. Stanza on the iPhone) can do this reformatting, but cannot deal with complicated text such as mathematics. We have been experimenting with using T_EX as the formatting engine. Of course it can handle complex mathematics, but it also creates the best line breaks of any ebook reader, e.g. Stanza. We will report our experiments with using T_EX as an ebook reader on the iPhone.

Nelson Beebe

BIBTeX meets relational databases

After giving some background and comments on the BIBTeX bibliographic database system, we discuss the problem of searching large collections of such data. We briefly describe how relational databases are structured and queried.

Portable new programs, bibtosql and bibsql, are introduced and their use is illustrated. The first handles the conversion of BIBTeX data to input for three free, popular, and portable, database systems. The second provides a uniform and simple user interface for issuing search queries to any of the supported backend databases.

We finish with a discussion of the contributions of the two late computer scientists to whom this article is dedicated.

Karl Berry and David Walden

TeX People: The TUG interviews project and book

We present the history and evolution of the TUG interviews project. We discuss the interviewing process as well as our methods for creating web pages and printed output from the interviews, using m4 as a preprocessor targeting either HTML or L^AT_EX. We don't claim great generality for what we have done. Nonetheless, some of our experience may be educational.

Jin-Hwan Cho

Extended DVI formats and DViasm

DViasm is a T_EX utility program that is designed for editing DVI files with three steps: disassembling, editing, and assembling.

The first stage of DViasm [1] supported the standard DVI file format as in DVItypex and DTL, but in a much more flexible way than those utilities. In the second stage [2], DViasm made it possible to handle two-byte characters, CJK and Unicode characters. The extended DVI formats generated by Omega, Japanese pT_EX were all supported, as well as ordinary L^AT_EX packages with subfont scheme such as CJK-L^AT_EX and Korean koT_EX. The final stage of DViasm confronts the two advanced T_EX engines, LuaT_EX and X_YT_EX, both of which can handle OpenType and TrueType files in a direct way without TFM files.

In this talk we introduce DViasm with a few interesting applications to the T_EX world, and discuss how DViasm handles the extended DVI formats generated by LuaT_EX and X_YT_EX.

- [1] Jin-Hwan Cho, *Hacking DVI files: Birth of DViasm*, *The PracT_EX Journal* (2007), no. 1, and *TUGboat* **28** (2007), no. 2, 210–217. <http://tug.org/TUGboat/Articles/tb28-2/\tb89cho.pdf>.
- [2] Jin-Hwan Cho, *Handling Two-Byte Characters with DViasm*, *The Asian Journal of T_EX* **2** (2008), no. 1, 63–68. <http://ajt.ktug.kr/\assets/2008/5/1/0201cho.pdf>.

Hans Hagen

The LuaTeX project: Halfway to version 1

We're close to releasing version 0.50 of LuaT_EX. What's old (and stable), what's new (and experimental) and what is on the agenda (but can get off). In this talk I will give an overview of what has happened so far, what is currently being done and where we expect to end up. If time permits, I'll also do a quick update on the MPlib project.

Hans Hagen

Plain TeX and OpenType

I will start this presentation with a quick overview of what OpenType brings us. I will use a (T_EX-based) previewing tool to show what such fonts provide. I will also demonstrate how you can use OpenType fonts in plain LuaT_EX, using a few generic modules that ship with the ConT_EXt distribution.

Hans Hagen

Why I still use TeX

Those familiar with ConT_EXt will have noticed that we're in the middle of a rather drastic rewrite tagged MkIV. This activity is related to the

LuaTeX project. In the process I occasionally ask myself the question if such an effort makes sense. In this talk I will discuss a few reasons why I consider it still worthwhile to invest time in TeX, in what direction we're heading, how the expectations evolved, and how ConTeXt can be deployed in coming years.

Idris Hamid

Dynamic Arabic: Towards the philosopher's stone of Arabic-script typography

We discuss the present status of the Oriental TeX project, particularly the problem of Arabic-script microtypography. This includes glyph substitution and hz parameterization.

Idris Hamid

Arabic typography: Past, present, and TeX

Accommodating the classical Arabic script to print typography has always been a challenge. In this talk we go over some of the history of this effort — including TeX-based solutions — with a view to providing a backdrop to the Oriental TeX Project and its progress.

Jim Hefferon

Gyre fonts for the easily confused

The TeX Gyre fonts are modelled after a group of fonts that are in widespread use. They bring to TeX users a number of very nice choices for making documents. But they have not been as quick to catch on with end users as they deserve. We'll cover some of the history and goals of the project, and explain how easy this option is to incorporate in your work.

Morten Høgholm

Next steps for breqn

The next version of the breqn package for automatic linebreaking of displayed equations is underway. We will discuss different areas of math typesetting: some things breqn handles well, some areas have room for improvement, and some areas are simply not covered. We will spend some time talking about the technical challenges posed by these requirements.

Morten Høgholm

Consolidation of expl3

The expl3 language used as the foundation of LaTeX3 has gone through a consolidation phase where almost each and every concept has been questioned, taken apart and put back. Sometimes in the same form as it was and sometimes in radically different forms. We will go through some of the most interesting changes and highlight the areas

where special effort has been made to ensure simple and natural interfaces.

Klaus Høppner

Introduction to MetaPost

MetaPost is a program strongly related to Knuth's original METAFONT. It uses nearly the same graphics language and syntax, but instead of bitmap fonts it produces PostScript output. So it can be used to create high quality graphics.

In MetaPost, points and paths may be described by a set of linear equations that are solved by the program. Thus, MetaPost becomes unique among other tools like PSTricks or commercial applications (e.g. CorelDraw). Additionally, the PostScript subset created by MetaPost can be interpreted by pdfTeX. So MetaPost figure can be directly included with e.g. the standard graphics package, while normal EPS images have to be converted first to be usable with pdfLaTeX.

The talk will present the basic features of MetaPost on a beginner's level, introducing the definition of points, creation of paths and their transformations, and show some examples how MetaPost solves linear equations.

Aditya Mahajan

ConTeXt math: Integrating Unicode and OpenType

LuaTeX provides the ability to process Unicode input and to work with OpenType fonts. These features are used in ConTeXt to fundamentally alter the handling of math typesetting. The user can type math using Unicode symbols and use OpenType math fonts.

In MkII, the pdfTeX version of ConTeXt, users could use Unicode math symbols. The mapping of Unicode symbols to math glyphs was done using catcode trickery. In MkIV, the LuaTeX version of ConTeXt, Unicode symbols are mapped to the corresponding location in a Unicode-aware OpenType font. To maintain the quality of math typesetting expected from TeX, while making ConTeXt Unicode and OpenType-aware we must ensure that

- math spacing is handled correctly;
- extensible characters (delimiters and operators) are handled correctly;
- traditional TeX macros for various symbols are mapped to correct OpenType locations;
- virtual OpenType fonts can be created from traditional TeX fonts.

In this talk, I will explain how ConTeXt MkIV handles these issues.

Aditya Mahajan

LuaTeX: A user's perspective

With LuaTeX, TeX has access to Lua. Since Lua is a normal programming language, implement-

ing algorithms in Lua is much easier than the usual macro jugglery required in traditional \TeX . \LuaTeX has been used in \ConTeXt to simplify many internal macros. But, for most part, the ease of writing a macro package in \TeX does not interest a normal user. How does \LuaTeX affect a normal user? In this talk, I take some \TeX programming examples from day to day usage and show how \LuaTeX simplifies them. For instance: writing loops without worrying about expansion; parsing input without exploding your brain; doing calculations without using an abacus.

Andrew Mertz and William Slough

A TikZ tutorial: Generating graphics in the spirit of \TeX

\TikZ is a system which can be used to specify graphics of very high quality. For example, accurate placement of picture elements, use of \TeX fonts, ability to incorporate mathematical typesetting, and the possibility of introducing macros can be viewed as positive factors of this system. The syntax uses an amalgamation of ideas from \METAFONT , \PSTricks and \SVG , requiring its users to “program” their desired graphics — which can be either a curse or a blessing, depending on one’s outlook. The latest revision to \TikZ introduces many new features to an already feature-packed system, as evidenced by its 560 page user manual. In this talk, we will present a tutorial overview of this system, suitable for both beginning and intermediate users of \TikZ .

Ross Moore

Towards tagged PDF

This talk will demonstrate recent work done by the author and Hàn Thế Thành, to enrich \pdfTeX with the primitives to allow the production of “tagged PDF”. As this is still very much work-in-progress, the talk will concentrate on presenting various aspects of tagging that allow the advantages of tagging to be easily appreciated. These advantages include, but are not limited to:

- substitution of Unicode characters, for glyph combinations from fonts that use encodings other than Unicode, via CMap resources and other techniques;
- alternative text, to be read by screen-readers;
- extraction of text from PDFs in XML format;
- extraction of mathematical content, in MathMLformat.

Each of these aspects will be illustrated by examples constructed using an enhanced version of \pdfTeX .

Also, I’ll try to explain the extra complexity of internal PDF structures required for generating properly tagged structure and content. If there is sufficient time, this may be followed by a discussion

of the requirements needed to adjust the \LaTeX format and packages, to facilitate the automatic production of properly tagged PDF, to become conformant with the ISO-32000-2 standard — also known as PDF/UA (Universal Accessibility, <http://pdf.editme.com/PDFUA/>). This standard includes MathML tagging of mathematical content; I wish to acknowledge Neil Soiffer (Design Science Inc., <http://accessiblemath.dessci.com/>) for motivation and much helpful advice, and testing, concerning this aspect.

Bob Neveln and Bob Alps

Writing and checking complete proofs in \LaTeX

\ProofCheck is a system for writing and checking mathematical proofs. Theorems and proofs are contained in a plain \TeX or \LaTeX document. Parsing and proof checking are accomplished through Python programs which read the source file. A general explanation of the use and structure of the system and programs is provided and a sample proof is shown in detail. The work done by the authors has been based on standard sentence logic, a non-standard predicate logic and set theory with proper classes. Theorems and proofs based on other foundations may be checked if external data files are modified. Four such data files and their possible modifications are described. In addition, the extent to which the formal language can be shaped to accommodate an author’s preferences is discussed.

Frank Quinn

\EduTeX : A source format for self-scoring tests

I’ll describe an NSF-funded project to use \TeX to produce self-scoring and otherwise interactive PDF documents for educational use. D. P. Story’s \AcroTeX Education Bundle already has a great deal of the necessary functionality and will serve as the starting point.

The objective is to produce customized documents — including tests — that are essentially independent of a testing system or other restricted environment. This is needed to make content more portable and less dependent on specific systems. PDF provides satisfactory functionality for the end product. The missing link is a source format that complies to PDF, complete with forms, links, and embedded JavaScript, and that can be written both by software problem generators and (in principle anyway) humans. An enhancement of \TeX seems ideal.

Boris Veytsman

Class writing for wizard apprentices

There are a number of excellent articles explaining \LaTeX classes writing for beginners [1, 2, 3, 4]. Of course, true wizards do not need any instructions.

This talk is intended for those T_EXnicians, who are no longer beginners, but do not (yet) qualify to be wizards. It discusses some tips and tricks of the trade: packages you may want to use and packages you must be compatible with; why the first two pages of a book take 60% of your time; what is wrong with L^AT_EX sectioning interface and why you better stick with it, etc. It is based on the experience of the author in writing L^AT_EX classes for various customers [5].

- [1] Flynn, Peter. “Rolling your own Document Class: Using L^AT_EX to keep away From the Dark Side”. *TUGboat* **28**(1), 110–123, 2007. <http://tug.org/TUGboat/Articles/tb28-1/tb88flynn.pdf>.
- [2] Hefferon, Jim. “Minutes in less than hours: Using L^AT_EX resources”. *TUGboat* **26**(3), 188–192, 2005. <http://tug.org/TUGboat/Articles/tb26-3/tb84heff.pdf>.
- [3] Mansfield, Niall. “How to develop your own document class—Our experience”. *TUGboat* **29**(3), 356–361, 2008. <http://tug.org/TUGboat/tb29-3/tb93mansfield.pdf>.
- [4] Pakin, Scott. “Good things come in little packages: An introduction to writing .ins and .dtx files”. *TUGboat* **29**(2), 305–314, 2008. <http://tug.org/TUGboat/tb29-2/tb92pakin.pdf>.
- [5] Veytsman, Boris. “Observations of a T_EXnician for hire”. *TUGboat* **29**(3), 484, 2008. <http://tug.org/TUGboat/Articles/tb29-3/tb93abstracts.pdf>.

Boris Veytsman and Leila Akhmadeeva

Medical pedigrees: Typography and interface

At the last T_EX meetings we presented our work on medical pedigrees [1, 2]. In this talk we report our newest advances in this field.

First, we improved the look of pedigrees based on the study of classical and modern illustrations in

biological and medical papers. Second, we worked on the interface to the program and incorporated the comments and suggestions received from the users.

We discuss the new challenges and unsolved problems.

- [1] Veytsman, Boris and L. Akhmadeeva. “Drawing medical pedigree trees with T_EX and PSTricks”. *TUGboat* **28**(1), 100–109, 2007. <http://tug.org/TUGboat/Articles/tb28-1/tb88veytsman-pedigree.pdf>.
- [2] Veytsman, Boris and L. Akhmadeeva. “Medical Pedigrees with T_EX and PSTricks: New advances and challenges”. *TUGboat* **29**(3), 484, 2008. <http://tug.org/TUGboat/Articles/tb29-3/tb93abstracts.pdf>.

David Walden

Self-publishing: experiences and opinions

Once a book is typeset (for example with L^AT_EX), you have to get it printed and distributed. If you have a traditional publisher, it takes care of many of the remaining steps. It also typically controls the product more or less in perpetuity. There can be circumstances where it is better to handle the publishing yourself.

Over the past several years I have gained considerable experience with self-publishing, using both traditional printing and print-on-demand. This presentation summarizes that experience and a number of the trade-offs, building on the sketch given in a companion presentation on the TUG Interviews Project.

As other conference participants are professionally involved in various aspects of publishing, I will also solicit (as part of the presentation) audience input on self-publishing possibilities. My final written presentation submitted to *TUGboat* will be augmented by what we learn during the presentation.