
An Earthshaking Announcement

Donald E. Knuth

Ladies and gentlemen, distinguished guests, dear friends: How appropriate it is for us to be meeting here in the city where Steve Jobs has made so many dramatic announcements. Today I have the honor of unveiling for you something that, in Steve’s words, is “truly incredible” — a successor to T_EX that I’ve been working on in secret for quite some time.

All of us know that computers and the Internet have been changing the world at a dizzying pace. Consequently few, if any, of the assumptions that I made when I first got T_EX to work in 1978 are valid today. Day after day I’ve been becoming more and more convinced that a totally different approach is now needed. Finally I woke up one morning with the realization that I couldn’t be happy unless I came up with a new system that rectifies my former mistakes — a system that leads to real progress.

Thus I’ve decided to scrap T_EX78 and T_EX82 and to start over from scratch. Of course the first thing that I wanted to fix was the most egregious design error that I’d made in the early system: The old T_EX was internally based on binary arithmetic, although its user interface was entirely decimal! Thus one could write, for example,

```
\ifdim .4pt = .39999pt \message{yes} \fi
```

and get the response ‘yes’. Furthermore a construction like

```
\dimen0=.4pt \multiply \dimen0 by 10
\showthe\dimen0
```

would give the answer ‘3.99994pt’; how ridiculous can you get? Are mathematicians supposed to like this? Are computer scientists supposed to like this? Is anybody supposed to like this?

(By the way, I apologize for using handwritten overhead-projector slides in this presentation, instead of making PowerPoint points. Some unexpected problems arose with my computer at home, and I didn’t want to risk security leaks by putting any of this material on someone else’s machine.)

Returning to my story, many of you may recall that the old T_EX represented all dimensions as integer multiples of a so-called “scaled point”, defined to be 1/65536th of a printer’s point, where a printer’s point was defined to be exactly 1/72.27th of an inch. How bizarre and anachronistic! Nobody remembers or cares about the old-fashioned units that printers used in the pre-Internet era. The graphic designers of today all know that there are

exactly 72 points to an inch, as specified by Adobe Systems; why should T_EX insist on calling that now-universal unit a “big point”? Indeed, what relevance will points of *any* kind be to anybody, ten years from now?

Moreover, T_EX never has allowed dimensions to exceed `\maxdimen`, or 16383.9999847412109375pt, which is roughly 18.8921175 feet (5.7583 meters). Today’s graphic devices make posters and banners much larger than this, so T_EX cannot cope.

At the other extreme, advances in nanotechnology mean that T_EX’s minimum dimension of one scaled point is far too large to accommodate 21st-century applications: A scaled point is huge, more than two farshimmelt potrzebies; it’s more than 53 Ångstrom units, way bigger than a hydrogen atom.

Thus I’m pleased to say that my new typesetting system finally gets it right: Dimensions can be arbitrarily large or arbitrarily small multiples of internationally accepted units. They can be expressed as exact rational quantities, like 3/7 of a yard; they can also be expressed in terms of irrational numbers like π and $\sqrt{2}$, so that circles and other objects can at last be rendered with perfect accuracy.

My design from 32 years ago was heavily influenced by what we used to call “efficiency”. I didn’t understand the implications of Moore’s Law. I didn’t realize that, in a few years, I wouldn’t care whether *The Art of Computer Programming* could be typeset in half a second, rather than waiting five seconds.

Examples of my tunnel vision abound, on almost every page of *The T_EXbook*. For example, I used backslashes and other strange characters to define what I called “control sequences”. Does any other system you know have control sequences? Of course not.

With my old rules people never knew whether or not a blank space really meant a blank space.

Therefore the basic input language for my new system is entirely a subset of XML, a widely accepted standard. However, XML is really only necessary at the lowest level, and most users won’t need to be aware of it, because we’ll see in a moment that there are many other ways to provide input.

Of course the character set for my new system is Unicode, so that there is 100% support for all of the languages and metalanguages of the world. Automatic spelling and grammar correction are built in for each language, as well as automatic correction to page layout and design. Different languages can freely be intermixed at will, always with appropriate ligatures, kerning, and hyphenation.

The old \TeX was limited to left-to-right typesetting; and some of its extensions also now handle the right-to-left conventions of many languages that my original implementation didn't consider. But my new system has been designed from the beginning to produce output in any direction whatsoever, whether horizontally or vertically or diagonally or along any kind of curved lines.

In fact, since 3D printing technology is now widespread, I decided at the outset that there was no reason to limit my new system to only two dimensions. Three dimensions are now standard in the new system; in other words, we deal with voxels instead of mere pixels. I've also provided hooks to allow future extensions to four or more dimensions, in case the string theorists prove to be right.

From a virtual standpoint, the notion of hypertext already gives us the equivalent of unlimited dimensionality, and the production of hyperdocuments and web pages will be one of the chief thrusts of my new system. \TeX 's old principles of boxes, glue, and penalties turn out to yield fantastic new ways to create multimedia documents, including animated videos and stereophonic sound.

Indeed, the input and output aspects of the new system aren't confined to traditional forms of text. Audio input and camera input are now seamlessly integrated, as well as sensor devices of all sorts. The system uses your GPS coordinates intelligently, if you are a mobile user, and senses your motions and gestures with accelerometers, etc. Complete support of haptics is also fully implemented. Instead of "what you see is what you get," we now also have "what you hear is what you get," and "what you feel is what you get."

There really is little difference between input and output in the new system, because any input can be output; conversely, any output from one hyperdocument can be input to another, or to itself. For example, music can be input from one or more MIDI devices, then either output to a conventional printed score, or to another MIDI device or group of devices—optionally segmented into individual parts, or transposed, or whatever you want. Going the other way, a printed score can be used as the input to a synthesizer, etc. I'll say more about these dynamic aspects later.

Does my new system have macros? No. Macros are *passé*; they're *so* mid-20th-century. Nowadays no one really needs macros, which we all know can be difficult to write and even dangerous. Everything in the new system is menu-driven, somewhat in the style of "Microsoft Word" but considerably enhanced: Experts have prepared recipes for everything you'll ever want to do, and these features

keep growing and getting better and better. The menus needn't appear on your computer screens in traditional pull-down or pop-up form; my system also responds to spoken commands and to gestures. And it quickly learns your preferences, so that it's customized to your own wishes, thereby making document preparation almost instantaneous.

You may have noticed that I've been referring a lot to my new system, but I haven't yet told you its name. I had to explain some of its characteristics before you could fully understand the name. But now I'm ready to reveal it, and more importantly to show you its logo:



How does one properly pronounce this name? Listen carefully: "i-téx" ["éé-técks"]. In the first place, you'll notice that it should be said musically, with *tones* as in Mandarin. (The first vowel is spoken with a dipping tone, "i", where the pitch falls and then rises. The second vowel has a strictly rising tone, "é", almost as if you're asking a question.) In the second place, you'll notice that I've also rung a bell when saying the name. The bell is also part of the logo:



It reminds us that $\text{\textcircled{TEX}}^*$ is not limited by obsolete conventions, not hampered by the days when documents were only seen but not heard. (However, the bell is optional, and it is omitted in documents that have no audio. Conversely, the logo is actually three-dimensional in a 3D document.) In the third place, did you notice that I said "tecks" instead of "techhhh"? I've decided to go with the flow, since almost nobody outside of Greece has ever pronounced 'TeX' with the correct 'X' sound.

Some of you may recall that I wrote the entire program for $\text{\TeX}78$ and $\text{\TeX}82$ all by myself, and you may be wondering whether I've done the same for $\text{\textcircled{TEX}}^*$. Don't worry: This time around I'm having the job done by people who know what they're doing. After many years I've finally come to realize that my main strength lies in an ability to delegate work and to lead large projects, rather than to go it alone. Programming has never really been my *forté*—for example, I've had to remove 1289 bugs from \TeX , and 571 from \METAFONT .

I made a very fortunate discovery during the summer of 2006 when I visited the Academy of

* [A bell rings at this point.]

Sciences in Armenia. There I learned that a huge amount of highly sophisticated but classified defense work had been done secretly at a large institute in Yerevan during the Soviet era. I met many of the people who had participated in those activities, and found that they were extraordinarily good programmers. Moreover, they were anxious to apply their skills to new domains. So they were a perfect match for my desire to make \TeX^* a reality.

I had long envied my colleagues at Stanford who had started up their own companies and gotten rich. Now it was my turn, and without much difficulty I formed a clandestine group called *Project Marianne*, comprising more than 100 of the top programmers in the world.

A few weeks ago some of you apparently discovered our website portal at `projectmarianne.com`. I've also seen blogs that wondered about the Armenian letter M on that page (Unicode #0544, "Men"). But as far as I know, nobody outside of our group has yet been able to penetrate the firewall that we built into that site, nor to look at any of our planning documents or initial demos. Needless to say, I'm pleased at this success, because the ability to create secure documents is another feature of \TeX^* .

Furthermore, I believe that nobody else has realized until now that 'Marianne' is an anagram of 'Armenian'.

After some deliberation, our group decided that all of the code for \TeX^* should be written in Scheme. We also decided to guarantee success by using all of the silver bullets that have been discovered by software engineers during the past decades: Information Hiding, Agile Software Development, Extreme Programming, Use Case Modeling, Bebugging, Look Ahead Design, Waterfall Modeling, Unit Testing, Refactoring, Rapid Prototyping, the whole shebang. We're going beyond ordinary Object-Oriented Programming to Aspect-Oriented Programming. But we're not using any formal methods, because everybody knows that formal methods are strictly academic. And we're abandoning the old notion of "literate programming" that I used when developing $\text{\TeX}82$, because documentation has proved to be too much of a pain.

Naturally it's out of the question for a system like \TeX^* to be freely available and essentially in the public domain, as the old \TeX system was. These talented programmers certainly deserve to be paid handsomely for their hard work. We have therefore devised some innovative pricing strategies, so that I'm sure you will consider \TeX^* to be an unbeatable bargain, considering the enormous value of its new features.

Here's the way it will work: Payments will be by monthly subscription, which will entitle you to unlimited use of \TeX^* on one or two of your own computers, or up to 40 hours \times 16 gigabytes of computing in our cloud of approved service providers. During the first year we're offering a one-month free introductory trial; thereafter your costs will depend on the quality of Internet access that is available in your area. For example, California users will pay \$99 per month, and German users will pay €69; but the monthly fee in Armenia will be only $\text{\$}1999$.

There are substantial discounts for senior citizens and for children under five years of age, as well as educational discounts for students.

Moreover — and this is the main innovation — you get a 10% discount for every new member that you can convince to join, lasting as long as you and that person are both enrolled in the plan. Thus if you can sign up just ten new subscribers, your access to \TeX^* will be free; and if you bring in eleven, you've essentially garnered a lifetime income.

My new enterprise operates by monthly subscription, instead of actually selling copies of the software, in part because the software is proprietary, but mainly because \TeX^* will change every day, due to constant improvements and upgrades to the system. Once upon a time I took great care in order to ensure that $\text{\TeX}82$ would be truly archival, so that the results obtainable today will produce identical output 50 years from now. But that was manifestly foolish. Let's face it: Who's going to care one whit for what I do today, after even 5 years have elapsed, let alone 50? Life is too short to reread anything anymore; in the Internet Age, nothing over 30 months old is trustworthy or interesting. We're best off just enjoying each moment as it happens.

\TeX^* will benefit the entire world's economy, because it will lead to tens of thousands of new jobs. For example, independent developers will be able to design and sell plugins that are distributed online and available for only a few pennies per week. Any \TeX^* user will be able to sell his or her own documents online, without leaving the \TeX^* system, because \TeX^* includes facilities for ordering, billing, manufacturing, and shipping. You can, for instance, write a blog, and others can package as many chapters of that blog as they wish into a customized book that is nicely printed and bound. \TeX^* will collect the appropriate payments from each customer and divide them fairly between you, the printer, the binder, and the shipper; the finished book will then arrive promptly at the customer's

residence. The operation will be something like the old Sears and Roebuck catalog, but now each item will be custom-tailored to an extent never before seen.

More importantly, there will be a large network of certified \TeX^* consultants, at various graded levels of certification. \TeX^* has no user manual, in the old sense, because the system changes daily. But it does have three varieties of online help: There's online help for dummies, online help for wizards, and personalized online help—in which you get to chat one-on-one with a certified \TeX^* helper. (Your membership fee entitles you to an hour's worth of one-on-one help each month.) Such helpers can arrange to work part-time for the \TeX^* consortium, out of their own homes and with flexible hours, in order to supplement their other income.

Let me conclude by describing a few more of \TeX^* 's features, so that you can begin to get a glimpse of how truly revolutionary it is. I've already told you that dimensions can be specified as arbitrary multiples of standard units; but that's just a tiny part of the story. \TeX^* actually is able to do arbitrary symbolic calculations, with polynomials and power series and matrices and partial differential equation solvers and convex optimization, etc., all integrated with graphics for automatic curve plotting and statistical charts, together with maps and satellite photographs of the world. When combined with \TeX^* 's synthesized voice output, you can do things like find a shortest route and navigate your car, all as part of an \TeX^* hyperdocument. If you're a professor like me, you can write math texts in which the formulas are changeable by each individual reader, who can evaluate them and plot their graphs interactively. (Incidentally I've changed math mode so that formulas must now be specified unambiguously, in such a way that they can be evaluated as well as printed; think MathML. This makes the formulas longer and more difficult to type, but that's a small price to pay for the added functionality.)

The hyperdocuments of \TeX^* can have any number of users, who can interact with each other and render images of themselves as avatars. This capability goes beyond the traditional kinds of virtual reality that are offered by systems such as Second Life[®], not only because of \TeX^* 's haptics but also because \TeX^* uses hyperbolic geometry—in which exponentially many avatars can be within a bounded distance of each other.

Such interactive documents obviously enable videoconferencing as a simple special case. I mentioned earlier that \TeX^* can receive input from all

kinds of sources: news feeds, webcams, traffic and weather sensors, heart monitors, seismographs, astronomical observations, you name it. All of these can be captured, mixed, and/or converted to other forms, such as audio or video or both. World-class tools are provided for photo retouching and image processing, computer-aided design, character and face recognition, as well as sophisticated filters for all sorts of data—including, for example, audio tracks and email. Output can be automatically formatted for lasercutters, embroidery machines, 3D printers, milling machines, and other CNC devices ... and shipped directly to consumers, as mentioned earlier.

One of our early plugins will feature an interactive cookbook that interfaces directly to your kitchen stove, oven, pantry, and refrigerator, so that you can prepare meals automatically with the ingredients that you already have on hand, and/or replenish your supplies by online ordering.

\TeX^* naturally incorporates extensive facilities for social networking. You can easily read the hyperdocuments prepared by others, and it's even easier to send and receive "tweets". (Your tweets needn't be limited to 140 characters of Unicode; the actual limit is a parameter. For example, you can set things up so that you receive only tweets of 50 characters or less.) With \TeX^* your entire life can be encapsulated into a dynamic hyperdocument, downloadable by anybody you designate.

I had intended to give you a live demonstration of \TeX^* today, instead of merely talking about its features. Indeed, \TeX^* was supposed to have provided all of my slides for this lecture, because the illustrations for a technical talk are among the simplest of all documents to create. Unfortunately, however, that has turned out to be impossible, because of hardware glitches and breakdowns in communication that I had no way to anticipate. (You can well imagine how difficult it has been to get all the pieces of \TeX^* to work together.)

But my coworkers assure me that the system is almost ready for its first major release, and we plan a worldwide press conference when \TeX^* is officially launched—hopefully next month.

Well, I've got to stop now: I can't tell you any more until our patent applications have all been filed. But I'm sure that, once you've tried \TeX^* , you'll immediately want to become a charter member of iTUG*.

◇ Donald E. Knuth
 Founder of *Project Marianne*