DVI-based Electronic Publication

Laurent Siebenmann
Matématiques, Bâtiment 425, Université de Paris-Sud, F-91405, Orsay, France
lcs@topo.matups.fr

Abstract

This article offers an introduction to three proposals that would make TEX's original device-independent output format "dvi" more useful for electronic publication of scientific documents.

The first is a set of small auxiliary 'atomic' fonts to complement Knuth's (atomic!) Computer Modern CM fonts so as to serve all European languages with a Latin (non-Cyrillic) alphabet. Here is a (last?) opportunity to make Knuth's original CM fonts part of an efficient and durable international system. Only in the English speaking world, would these fonts serve directly during TEX typesetting; beyond it, typesetting would use virtual fonts with composite accented characters based on these atomic fonts, which in turn would undoubtedly be extracted from the rather comprehensive EC font system; then a utility such as Peter Breitenlohner's "dvicopy" would convert the resulting "dvi" files to the more portable form using only the atomic fonts. The strong points of this scheme are unrivaled efficiency, and compatibility with the core of TEX users who have no reason to abandon Knuth's CM fonts.

The second proposal is the notion of a 'multi-standard' graphics object for inclusion in a TEX document. This notion was initially designed to solve the old problem of providing portable previewing of "eps" (EPS = Encapsulated PostScript) graphics. But it promises much wider compatibility: a single "dvi"-based posting should be able to serve all the multifarious graphics formats, and all the abilities and debilities of TEX viewers and printers in all computer environments. At least one programmer of a "dvi" viewer has been developing very much the sort of mechanism I recommend, namely Hippocrates Sendoukas in his viewer "DVIWin" for PCs. The near coincidence of our independently thought-out ideas is a good omen suggesting that in the present instance there may be essentially one way to skin the cat neatly.

The third proposal can be viewed either as a very general approach to implementing any standard for special commands or, alternatively, as a way to achieve "dvi" portability without such a standard. On each of the major platforms, one creates a freely available utility program converting a (candidate) standard for special's to the native format of any selected TEX implementation on that platform. The onus of implementation of a standard can thereby be entirely shifted from the implementors of TEX screen and printer drivers to the publishers of "dvi"-based postings. A low profile utility project will be discussed in some detail.

Several other developments will play a major role: notably universal availability of CM and related fonts in Adobe Type 1 format suitable for screen viewing, notably the BSR and BaKoMa series.

Granted the sum of these developments, TEX's own "dvi" format—enhanced by additives like PostScript "eps" graphics, or MPEG's "mpg" movies, *Hyper*TEX features etc.—should provide a remarkably efficient, stable, and polyvalent format for electronic publication in the mathematical sciences. This prospective "dvi_etc" posting format will be briefly measured against other formats such as Adobe Acrobat's portable document format "pdf".

E-PRINT: June 1996 [Master posting: ftp://matups.math.u-psud.fr/pub/TeX+/e-PubTools.dir/dvi_etc.dvi. . .]

Introduction

The race is on to establish formats for publishing the world's scientific literature in purely electronic form. Let us first recall the technological circumstances that have set this race in motion.

It is widely believed that the Internet now has the capacity necessary to deliver scientific articles in just a few seconds to any computer on the Internet—via www (World Wide Web), and also older systems like ftp (File Transfer Protocol). Hopefully, the costs for Internet transmission will remain quite negligible in spite of commercialization. Rapid access (hard disk) storage cost per page in an electronic library is already negligible. Provided efficient storage formats are used, the cost is now less than one cent per page for the 4 year life of a hard disk—and it is still rapidly falling.

However, there are serious traffic congestion problems that currently nearly freeze the Internet for key portions of the 24-hour day throughout considerable portions of the world. Thus a couple of complications should be kept in mind. (i) Although delivery via Internet is the favored vehicle, alternative delivery formats may have to take on an important slice of the action, notably compact read-only optical disks, known as CDROMs. This delivery format is slow and chunky, but its cost is low, and its capacity is probably orders of magnitude greater than those of Internet. (ii) A congested Internet will be more tolerable if we have a particularly efficient standard for science such as the "dvi".

Optimists expect that public fascination with bulky imagery available on the Internet will push data transport speeds and cost efficiency on the Internet *ahead of and beyond* the needs of the browsing scientist. But again this assumes that scientists use efficient norms insulated from technological inflation.

The leading candidates among electronic formats for the mathematical sciences seem to be:
—"tex": TEX's ASCII typescript input format defined by Knuth and elaborated by 'TEX markup formats' such as Plain, LATEX and others. This format usually must be enriched by special-ly included graphics objects. Hence it should perhaps be called the "tex_etc" format.
—"dvi": TEX's binary device-independent output format. Similarly, we will talk of a "dvi_etc" format that evolves with the 'special' additives and their attendant software.
—"ps": the PostScript page description language of Adobe Systems Inc. accepted by a majority of Xerographic printers, i.e. 'Laser' printers.
—"pdf": Adobe's format for its (presently) freeware viewer Acrobat, nourished by its commercial "ps" to "pdf" converter Distiller.
—"html": the www Internet format used by most browsers; mathematical material has been skimpily served, except through numerous bitmaps, and these have proved cumbersome. However, direct support for simple mathematics is being implemented under version 3 of "html".

How do these formats measure up to expectations of users of the scientific literature? For me, the major fault today of all of these formats (judged along with the software and hardware that support them) is 'low bandwidth' for browsing. Indeed, one still assimilates far less of interest, in an hour of electronic browsing, than in a well-run physical library. The hypertext features and net references (via URLs) are worthy innovations, but they do not fully compensate for slowness.

I believe that this is a sign that browsers (also called viewers or previewers or screen drivers) are in their infancy. Their speed should improve with increased microprocessor power—provided system complexity does not outstrip the power increase. The 10-fold increase in processor clock speed from 8 to 80 mhz in about 10 years augurs well. As does the doubling of bus width from 16 to 32 bits. One *bad* omen is that the resolution of images will lag behind—if the progress from 70 dpi to 100 dpi in 10 years is any indication.

Equally worrying is the widely heard prediction that Internet services will be too expensive for third world countries and also for any scientists without robust financial support. It is based on the observation that the Internet is being inexorably commercialized.

One can hope that a combination of growth and competition will make the necessary services affordable. However, there is a grave problem casting its shadow on this optimism—it can be stated as follows. Prices tend to be as high as a market can bear[1], while science teachers and researchers have relatively little financial clout (and students still less). Thus, we would be wise to ask whether scientists will not be playing a losing game in the same Internet league as business and professional groups.

My answer is the same as for transmission speed. To assure excellent service from the Internet in spite of lacking financial clout, scientists should adopt their own particularly *efficient* methods of

[1] This occurs particularly with monopolies and oligopolies, which are widespread in the world communications industry.

electronic publication. So efficient that even third world scientists will have adequate access to the electronic literature.

In the struggle for effective communication, the scientists' secret weapon is TeX. Secret because I believe that TeX's inherent complexity will restrict its use to a small circle centered on the scientific community in which it was developed. Donald Knuth paid extraordinary attention to efficient operation of TeX: the input "tex" typescripts and the binary output "dvi" page description files have information density comparable to that of ordinary ASCII prose (to within a factor rarely worse than two or three). And like prose they are compressible to well under 50%.

To some of you, the idea that electronic publishing with TeX should be done 'on the cheap' is repulsive. I claim that flexibility of design can allow the same TeX systems to do electronic publishing involving sophisticated typography and high tech—for example including color MPEG animations. Better, I want the same posting to serve both poor and rich scientists: a poorly equipped reader might see a still, black-and-white image instead of the color animation, but he should be able browse the same posting.

In prospective summary, our goal should be to extend the domain of extraordinary efficiency of TeX from the paper publishing world of the 1980's to the electronic publishing world of tomorrow—and without sacrificing high-tech ambitions.

Return to the more specific question: In what format should science be electronically published? Probably no one format will prove superior in all respects. I believe scientists should reckon carefully where their best interests lie, and be ready to provide some development effort at crucial points.

The "tex" format (in LaTeX dialect) is currently the most popular on the pioneering preprint server ftp://xxx.lanl.gov that was initiated by Paul Ginsparg. My article [7] can be interpreted as an attempt to make Plain TeX as efficient and as archival a markup system as LaTeX but by a radically new approach: macro compilation. In my view, however, "tex" format is best for author and publisher archives; I suspect it is too complex and fragile for non-experts.

The "dvi" format, although intended for the job by Don Knuth, seems to be in a checkmate position where browsable publication is paramount. It could lose out entirely if not further developed as I now explain.

For example, the "pdf" article format is browsable since both text and graphics are more-or-less instantly viewable (as well as printable). Exotic languages can be handled because "pdf" tends to include necessary fonts or the relevant parts thereof.

In contrast, a "dvi" article format is, as of today, probably only printable at best. Here is a quick rundown on the difficulties which it faces and which are addressed by my three proposals. The printability of "eps" graphics inclusions seems to depend on future implementation of a proposed standard for 'special' syntax recently published by T. Rokicki [5]. The browsability of "eps" graphics is a largely unsolved problem except for the lucky few who have a "dvi-to-ps" converter and an exceptionally good "ps" viewer. Finally, where one of the more exotic European languages such as Polish and Czech is involved, both printing and browsing will probably depend on the weighty EC font collection currently being completed (outside of Europe, expect to be disappointed when you call for it!).

On the other hand, this "dvi" format, which was designed by Donald Knuth's collaborator David Fuchs, has one signal advantage. It is by far the simplest format for the representation of TeX output; indeed it is presented in [3] along with helpful commentary in less than 10 pages. Alternative formats like "ps", and "pdf" are easily ten times as complex. Therefore "dvi" is comparatively easy to interpret and transform. For example, "dvi" is readily converted to "ps", while the (possibly useful) inverse conversion "ps" to "dvi" plus "eps" seems never to have been attempted.

I will focus successively on the three proposals intended to help the "dvi" format to avoid checkmate and then, just possibly, to go on to surpass all competitors! Originally, I explained these innovations on Internet discussion lists, and my often eminent interlocutors made significant contributions. Each remains tentative in the sense that it is just a sketched plan for action waiting for support. Developers can immediately examine many details by consulting the list archives indicated below, or by contacting me personally.

An 'atomic' complement to CM fonts

One problem with "dvi" postings is specifically a European one. No single font encoding quite manages to cover the needs of all European languages using a Latin-based alphabet. It is well known that Knuth's CM fonts are not sufficient for the Poles, and Czechs. Even the basic EC font encoding (the Cork norm) with numerous accented letters is not sufficient for the Latvians, Welsh and others. There are quite simply more than 256 characters in all.

The auxiliary text font series “tc” of the EC collection [4] will probably have to be combined with virtual fonts to bring them under the common EC roof. This will tend to render non-portable the “dvi” files involving such languages.

My suggestion for the Latin alphabets is to supplement Knuth's CM fonts by a single parallel series for auxiliary ‘atomic’ characters. It would contain chiefly separate accents, in distinct versions for lower case and upper case letters. But it would also include some notoriously troublesome characters, for example the Slovak character ľ (typed as `\v l`), which is difficult to render reliably and beautifully from atomic characters. Such troublesome characters are to be treated as atomic.

Note that the intervention of virtual fonts makes it possible to assemble composite characters from ‘atoms’ in distinct fonts; thus the CM fonts need not be modified in any way in order to be the core of such a system.

To achieve optimal typographic quality the author will use virtual fonts—but the virtual fonts are to be subsequently ‘atomized’ and so eliminated from the posted “dvi” file, for example by use of P. Breitenlohner's “dvicopy”. There is a significant typographical benefit to be gained from the use of virtual fonts: accent positioning can be adjusted to suit any national typography.

This scheme is compatible with the (core) EC fonts in the sense that virtual EC fonts (hopefully with the true EC metrics) can be based on CM and the envisaged ‘atomic’ complement to CM. This imposes inclusion of some miscellaneous characters that made their TeX début in the Cork encoding.

To be widely adopted for browsable electronic scientific postings, such complementary atomic fonts will have to be made available in Adobe Type 1 format, matching the BSR and BaKoMa Type 1 font series.

A somewhat similar proposal (independenent of mine) was made for Adobe-distributed Type 1 fonts by Pierre Mackay. It influenced the May 1995 encoding named 8r (for 8bit raw) used by the PSNFSS font system of LaTeX fame. But 8r is not atomic—if it had been made atomic (as the sadly lacunary Adobe Standard Encoding nearly was) that would have denied access to numerous characters present in the Type 1 “afb” files and reduce them to extra baggage. In my view, the CM fonts of Knuth are unique and deserve unique treatment.

The window of opportunity for acting on the proposal above is opening up with the approaching completion by Jörg Knappen of the EC fonts.

The main discussion of this proposal took place on the Math Fonts list [2] from March through May 1994; hopefully its archive will remain available. Technical problems related to the notion of ‘drift’ are discussed there.[2]

EPS graphics integration

The most glaring defect of “dvi” format has hitherto been the lack of a standard ‘special’ syntax for inclusion of graphics objects via TeX's `\special` command—for it tends to make “dvi” files non-portable.[3]

Where “eps” graphics (i.e. Encapsulated PostScript graphics) are concerned, a specific proposal for standard special syntax has recently been put forward by Tomas Rokicki [5].

Unfortunately TeX screen viewers are, more often then not, unable to directly view “eps” graphics. Prompt to recognize this problem, Adobe provided standard enhanced formats for screen viewing on Macintosh, and DOS (or MS Windows) systems.[4] On Macintosh, the enhanced viewer file is a PICT resource number 256 in the resource fork of the “eps” file; it contains either a “packbits” compressed bitmap or (more recently) vectorized PICT graphics. On PCs, the file has a binary format sometimes called EPSP; this has a special header, a segment for the “eps” file, and a segment for the bitmapped preview; the latter may be a simple “tif” bitmap with “packbits” compression, or a vectorized “wmf”. Here, “tif” means Tagged Image Format File (TIFF is the official acronym), and “wmf” means Windows MetaFile.

If the enhanced PC and Macintosh viewer formats had been one and the same format, it would surely have been adopted overnight as a standard. But as matters stand the enhanced “eps” files pose

[2] These problems (rather minor ones I believe) could be entirely bannished by a TeX Users Group standard for “dvi” interpretation.

[3] The “tex” format does not have an acute problem here. My “boxedeps.tex” (Feb. 1991), a TeX macros package (on CTAN), first made “tex” typescripts using “eps” graphics portable in spite of this lack of `\special` standardization. LaTeX has recently adopted this approach (basically one of systematically accommodating all extant syntaxes) and extended it to `\special` commands governing color, rotations etc.

[4] Adobe also provided a device-independent preview format EPSI (I for Interchange). Alas, its bulk is high and its quality low, and it is almost never used.

an annoying portability problem—and neither seems to have been widely adopted on UNIX systems—which tend to rely on postscript interpreters for graphics preview.

Thus, in practice, the standard proposed by Rokicki will assure portability between printers but often not between TEX viewers on different platforms. This puts DVI format in grave jeopardy as a publication format.

The portable viewing problem for "eps" graphics could (as noted above) be solved by seting up a viewing standard common to the Mac and PC platforms. Here is an obvious approach: have the "eps" text file `myfig.eps` accompanied by a parallel "tif" bitmap file `myfig.tif`. With modest effort, viewer builders on all platforms could derive the wanted preview from the "tif" bitmap. Such bitmaps can be displayed as fast as surrounding text. If the bitmap is scaled downwards (not upwards!) to its screen size the image quality is usually good—at least for the linefigures that dominate in the mathematical sciences. For best quality, figure labels should be put in a TEX overlay. The syntax proposed by Rokicki is adequate; so one merely enhances the functionality to provide preview using the "tif" bitmaps.

This solution seems worth implementing since it would solve an urgent problem. The basic functionality, i.e. use of a parallel "tif" file to give a viewer image corresponding to an "eps" file is already present in the two MS Windows viewers "DVIWin" [6] and "DVIWindo" [9]. They are by Hippocrates Sendoukas and Y&Y Inc. respectively. Unfortunately, such a development seems unlikely to come about spontaneously…

Multi-standard graphics integration

The above potential solution for "eps" graphics integration with preview is a very particular case of the use of several graphics standards to represent a single graphics object. Would it not be worthwhile to have a general scheme that goes well beyond the viewing problem as Adobe understood it ten years ago? We need a multi-standard scheme that can provide viewers with the very best screen representations available for any given platform!

This seems to me a bold enough idea to command the respect of a standards committee. I am hopeful that any number of distinct graphics norms can be handled by one and the same protocol so as to allow the same DVI file to serve on all platforms under all circumstances. Hippocrates Sendoukas was perhaps the first to envisage this possibility; see the documentation of his DVIWin [6].

Fortunately, the sort of special syntax proposed by Rokicki in [5] seems adequate for the functionality we want. For example:

```
\special{:: object multifig=myfig
                         width=200 height=120}
```

might be be interpreted as follows: there is a graphics object whose name root is `myfig`; its representation norm is to be selected by the driver among those available; the image is to be linearly scaled to fit exactly onto the rectangle 200bp wide and 120bp high located with its lower left corner at the TEX insertion point (all mentioned dimensions are thereafter corrected by TEX's magnification factor). Thus, in this most convenient case, the image rectangle (on paper or screen) are specified in the special's argument string viewed in isolation.

More specifically, consider the integration of a graphics object "myfig" that is represented as a high resolution color bitmap "myfig.jpg" of JPEG norm for viewers that have such wonderful[5] capabilities, and by a simple "tif" bitmap file[6] "myfig.tif" designed to serve as a lowest-common-denominator format for viewers without JPEG display capabilities. We explain the integration in terms of this example, which is one that possibly no TEX viewer yet handles today. Indeed, I believe it is fair to say that the one-file "eps"-with-preview offered by Adobe blinded most developers to the virtues of the more general multi-standard scheme.

Only the article author's intended experience, and that of the browsing reader needs to be described, for we have already indicated the sort of special command TEX and the viewer will manipulate behind the user's back.

The user will exploit a macro package for integration of multi-standard graphics; this package will have syntax similar to one of the several integration packages for "eps" graphics.[7] (The TEX user rarely wants to have to work with explicit figure dimensions!) Thus figure placement and size will be adjusted using conventional macro commands such as `\ForceWidth{0.75\hsize}`.

[5] Notably unrivaled compactness and nearly perfect scalability.

[6] Simple "tif" means black-white, 'packbits'-compressed, and with strip structure but no tiles. This is the sort of preview that Adobe Illustrator can include inside "eps" files to permit preview on PCs.

[7] It can be derived from the existing version by relatively minor generalizations.

Much as with LaTeX's cross-referencing, the result of a first production cycle is likely to be disappointing; typically the figure is crammed into a squared-out box. This is because TeX is quite unable to find out the shape or size of the figure since both graphics files mentioned are binary and TeX reads only text files. But the viewer can conveniently take on that task,[8] and the author is instructed to preview and expect good results on second and later runs.

If the viewer is able to accept JPEG files, preview is in color, and, if not, the image is the black and white familiar for "eps" preview.

To post the "dvi" file with its graphics inclusion, the author packages it with the "jpg" and "tif" files in a directory—whence the posting format name "dvi_etc". He should omit an ephemeral ASCII file "myfig.gdf" (say) a temporary 'graphics description file', which is created by the viewer or alternatively by the utility mentioned in the last note, and which serves to pass on to TeX the 'bounding box' information on the graphics object that TeX initially could not read.

What happens when a user browses this "dvi_etc" posting with a different viewer supporting the proposed multi-standard? The user will immediately see the best image his viewer can offer. But let us imagine that the user sees only the modest "tif" and is unhappy because he knows that his viewer can handle good quality "gif" images (but not "jpg"). Here the multi-standard again comes to the rescue: it suffices to convert from "jpg" to "gif" and the high resolution images appear![9]

Popular PICT and WMF scalable graphics become portable. The above JPEG example of multi-standard graphics integration is forward-looking. Here is an application that gives an immediate payoff. One creates a preprint posting using the Macintosh's native vectorized PICT graphics (suffix "pct") and the multi-standard special. A notice with the electronic posting recommends that users on other platforms convert[10] the PICT graphics to their favorite local norm, which for PCs means a Microsoft Windows MetaFile with extension "wmf". The multi-standard graphics integration using a single special will assure that the graphics appear on the end-user's screen as soon as a version in his native norm is available. Successful viewing of vectorized graphics almost always implies optimal printing, but not conversely.

In summary, with multi-standard graphics integration, one can confidently use and post with TeX the native vectorized (scalable) linefigures of Macintosh and Microsoft Windows. The viewing is then of optimal quality at all scales; this includes the labels if they are based on hinted outline fonts. A high degree of portability will be assured.

Since there is, at long last [5], a technical working group devoted to special standards, let us wish them the intelligence, openness, stamina and goodwill necessary to deliver a powerful standard. The long term health of Knuth's "dvi" format requires such a standard.

The Internet discussion of the above graphics multi-standard proposal can be consulted on www as a hypermail archive:

http://math.albany.edu:8800/hm/emj/

Assisted portability of DVI files

The classic retort to pious predictions concerning future standards is: "Don't hold your breath while waiting!" If the past is a measure of the speed of progress with special standards it could be another decade before an official standard deals frontally with multi-standard graphics. This educated skepticism drives me to propose an immediately applicable scheme, one which is similar in spirit to the "boxedeps" utility that I developed in the period 1989–1991 to make "tex" typescripts with "eps" graphics immediately portable.

This last proposal can be viewed either as a very general approach to rapidly implementing standards for special commands (graphical or not!) or,

[8] Alternative: It would be more conventional but somewhat less efficient to assign this task to a utility that is to be run first off; on the other hand, one utility per platform would suffice, and a disappointing first run would be avoided.

[9] There is a regrettable tendency among users to convert "jpg" files at an early stage to "eps"; this often means the file is encysted in the "eps" in a bulky form that moreover usually cannot be exploited for screen viewing. A best possible format, such as "jpg" for color bitmaps, deserves to be posted.

[10] Powerful conversion utilities are becoming available on most platforms. For the many users who refuse to become involved in conversions, one can hope that www servers will soon automatically supply to any www-connected viewer the graphics norm(s) it requests. Hippocrates Sendoukas recommends an alternative: viewers and printer drivers could ally themselves with conversion utilities so that they become able to exploit a broad spectrum of graphics norms.

alternatively, as a way to survive without such standards.

Zapping specials I first present a bite-sized form of assisted portability. We consider a small but embarrassing problem that has conjecturally held back the dissemination of "dvi" files containing *Hyper*TEX specials. The shortage of viewers that support the specials is not the culprit. In principle, electronic science journal postings in "dvi" format should *all* include *Hyper*TEX specials, since these potentially offer the revolutionary viewer features that are making "html" and "pdf" postings so attractive. The presence of these specials would be a strong incentive to TEX viewer modernization and a payoff to those who have installed *Hyper*TEX viewers. So why are *Hyper*TEX "dvi" files so rarely posted? Mostly, I claim, because so many existing viewers and printer drivers complain loudly about unrecognized specials, and sometimes go so far as to halt processing. This is quite incompatible with my own interpretation of Knuth's intentions for the `\special` command. See *The TEXbook* pages 228-229. I would say that he recommends a driver execute only specials it understands; and that, for those specials it does not understand, the driver should do exactly nothing, gracefully. Clearly many programmers have a different interpretation and espouse a less tolerant policy; and unfortunately, [5] does not clarify this issue.

Thus my introductory challenge is: Find a way to let existing *Hyper*TEX specials circulate in "dvi" files without wreaking havoc where *Hyper*TEX is not supported. There is a practical response. In view of the relative simplicity of the "dvi" format, one can make available on all platforms a small utility to automatically delete all specials from any "dvi" file. Thus specials that cause a fuss can be suppressed whenever necessary. In reality this 'zapping' would probably be just one among several functions of a utility.

There is a related development that should be mentioned here since it inspired the proposal for assisted DVI portability. Geoffrey Tobin has defined a new ASCII version "dvl" of "dvi" format along with converters in both directions [8]). As it stands, Tobin's system allows electronic publishers to adjust or update special syntax using a programmable editor. On the other hand, any autonomous and fully automatic utility for users would probably better operate directly on the binary "dvi" files.

After the above 'teaser' featuring 'special zapping', let us consider more important roles for utilities that propose to adapt "dvi" based postings to the idiosyncrasies of viewers and printer drivers.

There are a great many possible profiles; I mention two:

A) High profile utility Given any "dvi" file, this utility would convert the specials in it to those required by a driver to be specified (within bounds) by the user. Reasonable bounds might mean: to suit any of the screen and printer drivers present on the platform where the utility operates. Such a utility would certainly be worthwhile, but it would be of considerable size, and tiresome to construct, because specials and their syntaxes are numerous and various. There would also be a big maintenance problem since specials are proliferating; thus one might frequently encounter messages to the effect that certain specials listed in the log file could not be understood, and that, just possibly, a more recent version of the utility would do a more thorough job.

B) Low profile utility This would be designed to serve only quite restricted "dvi" files. For example those adhering to an ad hoc specials standard. This seems very suitable for a group of electronic publishers.

Let us consider a hypothetical but plausible instance. A group of electronic scientific journals decides that, for their current needs, a limited set of specials will suffice—say those governing color, "eps" integration with bitmapped preview, basic hypertext, and net references (URLs). They post "dvi" files plus graphics files adhering to their ad hoc standard and they build the corresponding utilities for all platforms and post them conspicuously. I provisionally call both the format and the utility "dvi_etc".

I believe that such a low profile scheme is ripe for implementation. So let us enquire how some obvious difficulties can be overcome.

Problem (i) The posted "dvi" had better not contain the ad hoc special standard in an active form since many users will be assaulted by complaints from touchy drivers.

Solution: The posted dvi can have its ad hoc specials removed to an external module, the special locations being remembered by a system of pointers. Then the "dvi" file contains no specials whatever, and there will be no complaints if the uninformed user directly views the "dvi". In that case, there should be a clear notice on a cover sheet (the first "dvi" page displayed!) indicating that all specials are currently inactive and should be activated by use of the "dvi_etc" utility. When this utility is used, the cover sheet changes to indicate for which driver(s) the "dvi" is now suitable and how versions for other drivers can be obtained.

Problem (ii) It should be possible to arrange for automated reception and viewing of downloaded documents.

Solution: Supposing (for example) that the reception platform is Macintosh, Mime conventions can assign to the auxiliary module "brouwer.etc" a type and creator so as to launch the "dvi_etc" utility. If this utility has been parametered for a specific "dvi" viewer, and put on alert status, then conversion of the posting to the format of the driver will proceed automatically. And if the viewer has been put on alert status, the downloaded "dvi" posting will automatically appear in a window of the viewer.

Problem (iii) Graphics objects in viewer-ready bitmapped form never have the same format for MSWindows and Macintosh systems. But, for simplicity, it is highly desirable to have one and the same graphics module for all platforms. What to do?

Solution (for b/w bitmaps): Since the conversion of simplest "tif" to a bitmapped Macintosh PICT resource is very easy to program, the external module need only contain the "tif" bitmap. The utility thus should (as necessary) combine "tif" plus "eps" files to create the Macintosh or PC preview-enhanced "eps" file format of Adobe.

Thus an article "brouwer" can be posted in two parts: "brouwer.dvi", a classical "dvi" file (without specials), and "brouwer.etc", a standard auxiliary module containing 'special' material. There is also a one-file alternative mentioned below.

Problem (iv) The "ps" and "pdf" formats present an article as a single file including the graphics. To compete successfully, a "dvi"-based posting should do as much.

Solution: This problem more or less disappears if each article is presented as a 'zip'ed directory. Zip is an interplatform standard for packaging and compressing files and/or directories. It attends to such niceties as directory structure, the text versus binary distinction, Macintosh types and creators, and UNIX permissions, etc.

Improved Solution: Anselm Lingnau ⟨lingnau@tm.informatik.uni-frankfurt.de⟩ pointed out on the EMJ list that, at the risk of violating the letter of the law for "dvi" format, it is in practice possible to insert an arbitrarily long segment of material into a "dvi" file without influencing its behavior under any known driver. (He proposed this device to include "pk" fonts in the "dvi" file.)

More precisely, in the terminal segment: *post* ⟨postamble⟩ ⟨font definitions⟩ *post_post* $q[4]$ 2 223's[$\geq$4], one is free to insert extra stuff before the four byte pointer $q[4]$ to the file offset of *post*.

Thus, all the special and graphics material necessary for an article called (say) "brouwer" can alternatively be located in a single file "brouwer.dvi_etc". This file will itself behave as a "dvi" file; but it should have a cover sheet that tells how it is to be exploited fully.

Notice that, in this low profile "dvi_etc" scheme, the material a user fetches from a server is always independent of his computing platform; thus it lends itself to email delivery and to exchanges among colleagues, i.e. to situations where the sender has no knowledge of the addressee's platform.

The politics of assisted "dvi" portability This new strategy—particularly in the above low-profile form—seems to have many advantages: (a) the total amount of work involved is hopefully less than if individual driver builders were all asked to retrofit a new standard; (b) by creating such a coherent system of utilities one can provide a universally valid distribution standard without having to extract assent and effort from every last driver builder; (c) upgrading such a standard is a pay-as-you-go affair, which can be transparent to the end-user and to the driver builders; (d) competing standards can coexist peacefully; (e) a small cross-platform group of like-minded developers is able to implement the system.

Scorecard

Here is a quick ranking of "dvi" format measured against its two closest rivals "ps" and "pdf".

- reliability: "dvi" and "ps" are very reliable, whereas "pdf" is still having numerous very serious teething problems. I will not discuss them except to point out that they are providing a few years of grace for the TEX community to come up with a competitive "dvi" based format.
- viewer availability: "dvi" and "pdf" rank about equal for scientists. And "ps" ranks last since "ghostscript" based "ps" viewers are not widely installed outside the UNIX world. There is still no "ps" viewer on Macintosh computers that is suitable for browsing. However a remarkable viewer PSVIEW has recently appeared in the PC world (contact address ⟨tdieting@iicm.tu-graz.ac.at⟩). Rankings are always provisional!

- format portability: The straggler is the "dvi" format: Only printed English text with mathematics is currently portable in "dvi" format; elaborate graphics or other special enhancements are non-portable. That is not good enough "dvi" format to hold its own as a publication format. But the proposed standard [5] and the three proposals of this article are aimed at extending this portability to cover essentially all scientific publication in electronic form.
- print quality: "dvi" and "pdf" offer optimal quality. The "ps" format is slightly inferior in current practice; for, by tradition, non-scalable "pk" fonts of 300 or 600 dpi resolution are used.
- print convenience: "ps" files often pose problems for lack of a parent application to manage the printing operation. In particular, the printing of a part of a "ps" file is too often out of reach of the casual user.
- graphics preview quality: "ps" and "pdf" beat "dvi" whenever the "dvi" viewer uses bitmapped preview. However, provided figure labels are put in a TEX overlay, the bitmaps are usually quite satisfactory.
- text (and math) preview quality: "ps" appears to be the loser. On PC and UNIX platforms, "ghostscript" based viewers are available. But there is a quality gap since the Type 1 hinting that makes ATM (Adobe Type Manager) font rendering so remarkable is still poorly supported by the viewers. It is said that "ghostscript" based viewers are learning to cooperate with ATM, or else to use grayscale blending (known as anti-aliasing) to replace Type 1 hinting. If ATM and Type 1 fonts are available, "dvi" matches "pdf" quality, and both provide best quality. If not, "pdf" is far superior at most magnifications.
- viewer speed: "dvi" viewers are faster with few exceptions, often dramatically faster. Their bitmapped graphics are also fast. I consider slowness the worst failing of the "ps" and "pdf" viewers.
- file size: "dvi" is the winner: typically only 2 Ko per page once compressed. Even when compressed, "pdf" is bulky because it is currently necessary to include (partially downloaded) Type 1 fonts; 20 Ko per page is typical for articles of moderate length.
- public acceptance: Only time will tell; the race has only begun. A big piece of the pie is sure to go to "pdf" since it is being actively promoted by a powerful corporation aiming at a multibillion dollar market: the totality of electronic publishing.

Conclusion

I have an undisguised bias in favor of "dvi" format for scientific postings, and with some good reasons:
—the TEX community is undisputed master of the "dvi" standard.
—"dvi" format is output by TEX; thus, so long as TEX dominates the composition of scientific documents, "dvi" format will hopefully[11] be conveniently convertible to all page description formats, including newcomers.
—the comparative simplicity of "dvi" format makes possible significant developments in return for moderate effort.
—TEX viewer development has been the cutting edge of TEXnology for many years. If "pdf" format gains a near monopoly position for browsable postings, I fear there will be a dramatic wilting back of TEX system development, particularly in the public domain and shareware realms. When a viewer falters, the linked TEX user interface also falters. Moreover, a monetary hurdle to electronic science publication may then become a fact of life—one that is, ironically, substantial for an individual but quite negligible for an organization. (Acrobat Distiller costs several hundred dollars, as do commercial TEX systems.)

The future role of the "dvi" format for electronic publication of science articles has yet to be decided by programming and subsequent competition. I believe the three developments informally proposed in this article could help TEX's own "dvi" format win a role that is viable, and indeed enviable.

[11] This not so certain that an electronic publisher would dream of discarding the "tex" source files!

References

[1] Electronic Math Journal List, archive at http://math.albany.edu:8800/hm/emj/; wais index at http://nyjm.albany.edu:8000/SF/emjsearch.html; this list was organized by Mark Steinberger and the author.

[2] Math font discussion list, math-font-discuss@cogs.sussex.ac.uk; see its archives on ftp cogs.sussex.ac.uk; this list was organized by Alan Jeffrey, alanje@cogs.susx.ac.uk).

[3] D. Knuth, "TEX the program", Addison Wesley, 1986.

[4] Jörg Knappen, "Release 1.2 of the dc-fonts: Improvements of the European letters and first release of the text companion symbols", *TUGboat* **16(4)** (December, 1995), pp. 381–387.

[5] Tomas G. Rokicki, "A proposed standard for specials", *TUGboat* **16(4)** (December, 1995), pp. 395–401. (Compare earlier proposals by Don Hosek (1987), and Nelson Beebe (1990) archived on CTAN in dviware/driv-standard/papers/.)

[6] Hippocrates Sendoukas, *DVIWin*, a DVI viewer and printer driver for MSWindows exploiting PK fonts, on the CTAN archive, in dviware/dviwin/.

[7] L. Siebenmann, "Occam's Razor and Macro Management", (Proceedings, EuroTEX 1995, Papendaal, Netherlands); included in electronic form in the distribution of Occam, see CTAN archive macros/generic/ and ftp://matups.math.u-psud.fr/pub/TeX/.

[8] G. Tobin, *DVL a DVI Text Language*, 1995, a portable package found on CTAN in dviware/dtl.

[9] Y&Y Inc., *DVIWindo*, a DVI viewer and printer driver for MSWindows exploiting Adobe Type 1 fonts; commercial distribution: see http://www.YandY.com.