

СТАНДАРТИЗАЦИЯ РУССКОГО Т_EX: УТОПИЯ ИЛИ НЕИЗБЕЖНОСТЬ*

С. В. ЗНАМЕНСКИЙ

Институт программных систем РАН, Переславль-Залесский, Россия
e-mail: znamensk@ipsun.msk.su

The article is devoted to the problems arising in attempts of working out the standards for information presentation in Russian computer systems based on T_EX russifications.

1. Постановка задачи

Система T_EX, разработанная Д. Кнудом в 1982 г., активно используется людьми, имеющими дело со сложными математическими формулами, в качестве профессиональной издательской системы; удобного средства подготовки книг, статей, препринтов; гибкого и ясного языка коммуникации специалистов по электронной почте; экономичного стандартного способа представления научной информации в сетевых архивах и базах данных.

Именно гармоничное сочетание этих качеств с изначально заложенными возможностями расширения функциональности (мощный язык программирования, открытость интерфейса с дополнительными утилитами) обеспечивает T_EXу эффективное выживание в мире бурно развивающихся программных продуктов, ориентированных на непрофессионалов.

К сожалению, люди, занимающиеся улучшением и разработкой утилит, далеко не всегда учитывают необходимость сохранения всех перечисленных качеств. В результате новые удобства для автора и новые возможности для издателя оборачиваются порой непреодолимыми трудностями для пользователя, пытающегося просмотреть или распечатать документ, полученный от коллег или из архива. Острота этой проблемы возрастает по мере разработки новых версий L^AT_EXа, а для российских пользователей, для которых T_EX, очевидно, останется основным средством общения еще на много лет, усугубляется распространением трудносовместимых вариантов русификации.

Российским фондом фундаментальных исследований поддержана идея фиксирования (не столько юридически, сколько фактически) стандарта представления информации в компьютерных системах России, базирующегося на русификации T_EXа. Реализация этой идеи предоставила бы исключительно благоприятные возможности для создания компактных архивов и баз данных научной информации с поиском не только по ключевым словам, но и по фрагментам формул и значительно облегчила бы коммуникацию ученых.

*Работа выполнена при поддержке Российского фонда фундаментальных исследований, грант № 95-07-19400в.

© С. В. Знаменский, 1997.

Предлагалось фиксировать набор русифицированных форматных файлов, очертив тем самым рамки стандарта. Выделяемые средства предполагалось направить на создание лицензионно чистой системы под различными платформами, поддерживающей фиксированные стандартом форматы, и организацию свободного доступа по сети. В ходе работ над проектом (см. www.botik.ru/PSI/RFBR_TeX) неожиданно выяснилось, что такое решение не имело бы ожидаемого эффекта по двум причинам: во-первых, мало кто согласится отказаться от тех или иных удобств, предоставляемых привычной ему установкой \TeX . В результате получилась бы еще одна трудносовместимая с уже существующими русификация. Во-вторых, существование и интенсивное развитие современных средств подготовки документов с непрерывно расширяющимися возможностями вскоре поставит на таком фиксированном стандарте клеймо архаичности. Последнее может произойти еще до распространения стандарта, и тогда стандарт умрет так и не родившись.

Трудно предположить, какие средства потребуются для решения этой проблемы, чтобы можно было привлечь сильнейших разработчиков со всего мира, мобилизовав их усилия на создание такой системы, которая бы в течение длительного срока оставалась бы вне конкуренции. Подобный проект под названием “ \LaTeX 3 ” осуществляется уже в течение трех лет, в работе задействованы сильнейшие специалисты по \TeX у, но до сложившегося стандарта ему еще далеко: идея поддержки многоязычных текстов, похоже, завела разработчиков в такой лабиринт проблем, что предлагаемые сегодня пути стандартизации отнюдь не кажутся вполне удовлетворительными, во всяком случае для русскоязычного пользователя.

Становится ясно, что при современных темпах развития компьютерной сферы если и возможно создать новый стандарт, превосходящий все существующие разработки, то это настолько дорого, что никогда не окупится. Ясно и другое: стандартизация в первую очередь должна ориентироваться на интересы всех групп пользователей \TeX а, а осуществляться лишь в той мере и в том виде, в каких ограничения свободы пользователей являются общепризнанно и бесспорно необходимыми. Сказанное означает, что правильное решение поставленной задачи должно опираться на детальный анализ используемых расширений \TeX а, проявлений несовместимости, их причин и путей устранения.

2. Проявления и причины несовместимости используемых расширений \TeX а

2.1. Классификация уровней совместимости

Идеальное воспроизведение документа зачастую требует наличия именно тех версий стилевого и форматного файлов, а также шрифтов, с применением которых он был подготовлен. По документу обычно невозможно определить, какие именно версии использовались и практически невозможно хранить и распространять все версии всех файлов. К счастью, идеальное воспроизведение документа практически никогда не является необходимым. По этим причинам удобно выделить следующие уровни совместимости при обработке документа другой системой:

идеальный — форма и положение каждой буквы или символа сохраняются;

корректный — не происходит ошибок и потерь информации при обработке файла, документ не имеет видимых дефектов;

приблизительный — не происходит ошибок и потерь информации при обработке файла (при этом строки документа могут иметь разную ширину или большие пробелы).

2.2. Форматные файлы

При запуске T_EXа для обработки входного файла обычно подключается тот или иной комплект настроек, макроопределений и алгоритмов переносов, называемый форматным файлом. Пример такого форматного файла был разработан Д. Кнудом и под названием “Plain” входит в большую часть установочных вариантов T_EXа. Следующий форматный файл, разработанный Л. Лампортом, называется “Lplain” или “L^AT_EX 2.09”. Он предоставляет автору богатый сервис автоматической нумерации формул, утверждений, заголовков, значительно облегчает составление обширной библиографии. В отличие от Plain, L^AT_EX базируется на том принципе, что в начале каждого документа следует подгрузить стилевой файл, определяющий, в каком из стилей (статья, отчет, книга, письмо или новый нестандартный стиль) будет оформлен документ. Стиль определяет используемые в заголовках шрифты и другие особенности дизайна документа. Файл, выполненный в L^AT_EXе, легко узнать по блочной структуре документа, оформленной рамками `\begin{нечто} ... \end{нечто}`.

К сожалению, в стандартных стилях L^AT_EXа 2.09 крупный шрифт заголовка диссонирует с мелким шрифтом формулы, которая в этом заголовке может оказаться. Этот дефект являлся трудноустранимым: поскольку L^AT_EX предусматривает много размеров текста, а для каждого из этих размеров текста требуется описать и подгрузить все шрифты, которые могут оказаться в формулах, включая индексы и индексы индексов, то устранение дефекта в рамках старого L^AT_EXа требовало чрезмерных машинных ресурсов.

Желание получать заголовки удовлетворительного качества привело к созданию новой схемы выбора фонтов — так называемой NFSS (New Font Selection Scheme), добавление которой к классическому L^AT_EX 2.09 приводит к формулам правильного размера как в заголовках, так и в примечаниях, рефератах и других элементах оформления. Кроме того, добавились команды `\shape`, `\size`, `\family`, `\series`, независимо меняющие форму, размер, гарнитуру и жирность шрифта, подлежащего загрузке, и команда `\selectfont` для подгрузки нового шрифта. При этом изменилось действие старых команд управления шрифтами, например, команда `\sf\bf` в классическом L^AT_EXе давала не полужирную версию рубленого шрифта, как в NFSS, а стандартный полужирный шрифт `\bf`. Такой L^AT_EX с NFSS распространен в России в составе оболочки TIS (T_EX Integrated System), разработанной В. Малышевым в 1993 году. Улучшение дизайна статей привело к следующему нарушению совместимости: файлы, подготовленные с использованием команд NFSS, не обрабатываются старым L^AT_EXом, а дизайн документов, подготовленных в старом L^AT_EXе, может отличаться от исходного вплоть до изменения количества страниц документа.

На мой взгляд, то обстоятельство, что новая система поддерживает все возможности старой, изредка внося минимальные искажения (как правило, улучшения на уровне корректной обработки), является вполне нормальным явлением, не создающим проблем несовместимости: пользователь всегда в состоянии обновить систему, получив новые возможности и ничего не потеряв. Как будет показано ниже, не все улучшения вносятся в должной осмотрительностью.

Американским математическим обществом разработаны шрифты и форматный файл *AMS*-T_EX, предназначенные для оформления математических статей со сложными формулами. В отличие от L^AT_EXа этот формат предназначен в большей степени для набора и издания готовых документов, чем для написания статей. Так, например, в нем отсутствует автоматизация нумераций формул, библиографических ссылок и т.д., но взамен имеется облегченный доступ к любой модификации элементов дизайна. Как и в L^AT_EXе, документ может начинаться командой `\documentstyle`, но в *AMS*-T_EX она не является обязатель-

ной. Файл, подготовленный в $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX e, обычно содержит строку `\input amstex`, хотя может быть корректно обработанным и без нее, если в командной строке указать `&amstex`. Блочная структура `\нечто ... \endнечто` присутствует в $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX, но является весьма изощренной: например, доказательство теоремы может начинаться в одном параграфе, а кончаться в другом. Несколько команд стандартного Plain (`\proclaim`, `\matrix`, `\cases` и др.) изменены в $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX e, что иногда исключает возможность обработки $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX ом файла, подготовленного в Plaine. На территории России $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX версии 2.1 используется в большинстве издательств, специализирующихся на литературе по математике. Предыдущая (1.1) версия до России не дошла, и проблема обработки файлов, подготовленных в $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX1.1, не является острой для российского пользователя. Она возникает лишь при желании обработать редкий старый файл из зарубежного электронного архива. По идее разработчиков, она решается заменой в начале обрабатываемого файла указания на стиль `amspppt` указанием на `amspppt1` и в простых случаях это срабатывает.

Стремление к объединению возможностей $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX а по элегантному оформлению математических формул с \LaTeX овской автоматикой привело к созданию двух форматных файлов: $\mathcal{A}\mathcal{M}\mathcal{S}\text{\LaTeX}$ (с синтаксисом \LaTeX) и $L\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX (с синтаксисом $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX). Первый из них (версия 1.1) основан на NFSS и получил столь же широкое распространение в России, как $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX, уступая лишь \LaTeX у. По второму нет доступных описаний и мне не известны случаи использования его в России.

Проблемы, связанные с русификацией, мы обсудим в следующем пункте, здесь же заметим, что поддержка языков на латинской основе также требует модификации форматных и стилевых файлов. Если большая часть языковой специфики может быть учтена введением дополнительных стилевых файлов, то алгоритмы переносов должны быть заложены заранее в форматный файл. Даже американский и британский английские языки, не говоря уж о многообразии европейских языков, требуют закладки индивидуальных алгоритмов переносов. Поддержка многоязычности должна предусматривать также возможность изменения кодировки дополнительных букв. Идея включить в одну систему команды смены шрифтов, кодировок и языков привела к возникновению нового форматного файла \LaTeX 2e, который наряду с новыми возможностями для пользователей создает и новые проблемы.

Все предыдущие варианты \LaTeX а объявлены устаревшими и не поддерживаются. Каждые полгода выходит новая версия \LaTeX 2e, которая действительна в течение года, после чего подлежит обновлению. Каждая из версий поддерживает старый \LaTeX 2.09 примерно на том же уровне, на котором его поддерживал \LaTeX с NFSS, но команды смены шрифтов снова изменены и команды NFSS не поддерживаются. Теперь NFSS переименована в NFSS1 в отличие от NFSS2, заложенной в основу \LaTeX 2e.

Новая версия $\mathcal{A}\mathcal{M}\mathcal{S}\text{\LaTeX}$ а (1.2) реализована уже не в виде форматного файла, а набором стилевых файлов над \LaTeX 2e и не поддерживает описанные в сопровождавшей старую версию $\mathcal{A}\mathcal{M}\mathcal{S}\text{\LaTeX}$ а документации по NFSS команды, перечисленные выше.

Что же остается пользователю, желающему обработать файл на английском языке, выполненный в \LaTeX е неизвестной версии? Можно попытаться обработать его в новом (2e) \LaTeX е, а если он не пойдет, то почти наверное нужен \LaTeX с NFSS1. Таким образом, пользователю, желающему обработать любой файл из FTP-архива (например, xxx.lanl.gov) нужно иметь одновременно как минимум Plain и две разновидности \LaTeX а, причем система должна менять пути поиска файлов в зависимости от выбранного форматного файла, поскольку стилевые файлы для этих разновидностей с одинаковыми именами не взаимозаменяемы. Для правильной настройки подобной системы обычно необходимы время и

высокая квалификация пользователя.

Удалось ли в новом L^AT_EX 2_ε решить проблему многоязычных текстов? В принципе, да: есть варианты установки, в полной мере обеспечивающие возможность подготовки документов на нескольких языках на латинской основе, причем в документ могут входить фрагменты на одном языке в разных кодовых таблицах. Для большинства европейских языков это не так бессмысленно как для русского, поскольку такие фрагменты на 80–90 % состоят из латинских букв и поэтому почти уверенно воспринимаются в любой разумной кодировке.

Может ли удовлетворить российского пользователя подход, при котором русские буквы не только не могут входить в командные слова, но и не появляются в протоколе обработки файла (*.log), что резко затрудняет верстку? Если добавить, что ни один из существующих сегодня на русском языке документов не может быть обработан без предварительного редактирования новым форматным файлом, а дополнительные строки не приведены ни в одном из существующих руководств по L^AT_EXу, то станет ясно, что скорого решения проблем массы российских пользователей от многоязычного L^AT_EXа 2_ε ждать не приходится.

2.3. Варианты русификации

Проблемы российских пользователей T_EXа складываются из проблем западных пользователей и проблем, порождаемых неоднозначностью русификации. Казалось бы, какие проблемы могут возникнуть, если в T_EXе начиная с версии 3.0 (т.е. вот уже почти пять лет) предусмотрена возможность закладки алгоритмов переносов для нескольких языков с последующим переключением и любой символ может быть объявлен буквой, абсолютно равноправной со всеми остальными буквами. Известен не один способ научить T_EX “говорить” по-русски, и далеко не все они равноценны. Возникающие при этом проблемы заслуживают более подробного рассмотрения.

Начнем с кодировок шрифтов. Первые, давно забытые варианты русификации семибитного T_EXа возникли на рубеже 70-х гг. почти одновременно и независимо в Вашингтонском университете (шрифты MСY*, замененные на WNCY*) и в Институте физики высоких энергий, Протвино (СМС*, переименованные в СМ*Z*). Естественно, русские буквы А, Б, Ц, Д и другие совпадали по кодировке с созвучными латинскими буквами. Что касается букв Ю, Я, Ъ и других, не имеющих созвучных латинских аналогов, то в Вашингтоне была использована привычная западным пользователям транслитерация или команды Ю→YU, Я→YA, Ъ→\Cprime и т.д. с вариациями Ц→TS=Ts=C, в то время как в Протвино использовалось однозначное соответствие Я→q, В→w, Е→e, напоминающее современную кодировку КОИ-8. Вашингтонский вариант был удобнее для набора на латинской клавиатуре, протвинский использовался со специальным перекодировщиком и обеспечивал в отличие от вашингтонского потенциально безупречные переносы. Тогда русские DVI-файлы были безусловно переносимы с компьютера на компьютер, независимо от кодировки входного файла. Качество шрифтов Протвино вскоре стало существенно выше вашингтонских.

Во многих обрабатываемых документах русские слова разбавлены английскими. Не хотелось переключать шрифт перед и после каждого слова из латинских букв. Насколько проще казалось иметь один шрифт с русскими и латинскими буквами одновременно, чтобы сообщения об ошибках не искажались, а печатались соответствующими буквами и не сбивалась автоматика формирования словарей, оглавлений и указателей. Так соблазнительно было объединить русский и английский в один общий язык с общим алгоритмом переносов

и общими шрифтами, названия которых так удобно оставить старыми. В довольно-таки разобщенной среде разработчиков появлялись десятки вариантов реализации этой исконно российской идеи. Появилось немало вариантов расширенных стандартных Т_ЕХовских шрифтов с сохраненными или измененными названиями. Если на недавнюю попытку изменить канонические для Т_ЕХа шрифты СМ*-семейства Д. Кнут отреагировал гневным письмом, то шрифты Протвино, известные также по именам создателей А. Самарина и Н. Глонти, лежат на каноническом архиве в нескольких местах: два разных варианта рисунков букв в файлах с идентичными названиями, два разных варианта названий у идентичных шрифтов и, наконец, разные варианты кодировок у шрифтов с одинаковыми названиями и рисунками... И довершают картину несанкционированные авторами улучшения их файлов с сохранением названий. От идеи Кнута о том, что при работе Т_ЕХа создается машинезависимый (DeVice Independent=DVI) файл, содержащий предельно экономное полное (благодаря тому, что информация о форме и размерах букв в документе представляется только названиями шрифтов) представление документа, фактически не остается ничего.

Двойственный статус шрифтов Самарина—Глонти (практически неограниченный сетевой доступ в сочетании с распространенными слухами об ограничениях возможности использования) спровоцировал разработку нескольких комплектов русифицированных расширений СМ*-семейства, из которых наиболее распространенным является семейство ЛН*, разработанное ассоциацией пользователей кириллического Т_ЕХа (CyrTUG) и официально объявленное свободно распространяемым. Поскольку основное начертание в шрифтах ЛН* (создатели О. Лапко, Н. Ходулев) проработано лучше, чем в СМС*, я в конце 1995 г. предложил включить их в установочный пакет Российского фонда фундаментальных исследований в качестве стандартных. Руководство ассоциации выставило перед РФФИ требование купить шрифты у разработчиков и издательства "МИР". Поскольку РФФИ осуществляет поддержку проектов и не может платить за уже проделанную работу, мне совместно с Л. Знаменской пришлось срочно разработать новое расширение СМ*-шрифтов, не уступающее существующим по всем основным показателям.

Если это вообще возможно — навести порядок в шрифтах — то, по-видимому, нельзя не пожертвовать менее распространенными и слабее проработанными. По моему мнению, заслуживает рассмотрения комплект, вошедший в TIS (Протвино, 1992 г.), ЛН*-шрифты из дистрибутива CyrTUG (1993 г.) и наше новое RF-семейство (1996–97 гг.).

Псевдоразнообразия шрифтов создает проблемы несовместимости не только в плане обработки DVI-файлов. Явное указание имени шрифта в обрабатываемом или стилевом файле делает их непереносимыми на другую машину. В перспективе эту проблему можно было бы решить введением макроса `\rusifont`, который по имени стандартного шрифта для Т_ЕХа вычислял бы имя присутствующего в системе расширенного шрифта и подгружал бы его. Такой макрос разработан автором, выверен на основных форматных файлах и доступен вместе с нашими шрифтами и правильной русификацией по FTP:

ftp.botik.ru/pub/msdos/Russian_TeX_97

Шрифты являются далеко не единственным источником несовместимости русификаций. Есть большое искушение сделать русские буквы не такими, как латинские, чтобы в математических формулах их легче можно было набирать прямыми, как это принято в изданиях на русском языке, или чтобы состыковать внутри Т_ЕХа разные кодировки входного и выходного файлов. Такое решение существенно усложняет поддержку множества самых разнообразных подгружаемых пакетов. Организованная А. Рожено сетевая дискуссия с участием А. Шеня и С. Львовского и обсуждение этого вопроса на конферен-

циях CugTUG выявили единодушное мнение специалистов: русские буквы должны быть абсолютно равноправны с латинскими. Старые (пока еще используемые) реализации T_EXa полноценно этого сделать не позволяют, отсюда рекомендация — не использовать русские буквы в командных словах.

Поскольку многоязычная поддержка русского языка в рамках L^AT_EXa 2_ε, очевидно, не будет в достаточной мере отработана в ближайшее время, то принят тезис минимальной русификации ВСЕХ форматных файлов: в них не должны отражаться никакие традиции русского книгоиздания (это — забота создателей стилевых файлов), но должны закладываться качественные переносы и равноправие русских букв с латинскими. С командой `rusifont` закладываются при необходимости русифицированные расширения шрифтов и коды дополнительных символов — номера и кавычек-“елочек”. При этом по причине изменчивости L^AT_EXa 2_ε важно не трогать исходные файлы, чтобы процедура обновления L^AT_EXa 2_ε осталась неизменной.

С алгоритмами переносов тоже связаны проблемы. Во-первых, они разные во всех наиболее распространенных пакетах. Во-вторых, ни один из них, строго говоря, не соответствует ни правилам русского языка, ни традициям книгопечатания.

Оказалось принципиально невозможным обеспечить правильные переносы в абзаце, содержащем предложения на русском и английском языках, если объединить алгоритмы для двух языков в одном. Дело в том, что только команда смены языка может разграничить части абзаца с разным значением `\righthyphenmin`, равным 3 для английского языка и 2 для русского, иначе на весь абзац подействует значение, имеющееся при выполнении команды `\rag`.

Сказанное позволяет однозначно заключить, что создать систему, *идеально* обрабатывающую все старые файлы на русском языке, в принципе невозможно, и пока файлы не будут содержать формализованные комментарии, с какими элементами русификации они создавались, речь можно вести лишь о *корректной* обработке.

2.4. Графические расширения

В T_EXe была предусмотрена, но не описана возможность включения в документ графических файлов. Поэтому способ подготовки графики и команды включения фатально определяются используемыми драйверами. В развитых странах и под UNIX чаще всего принято все файлы готовить в постскрипте, в остальных — под DOS/WINDOWS в форматах PCX или BMP. Сегодня для нас эта проблема вчерне решается обработкой программой DVIPS перед отправкой и программой GhostScript/GhostView при получении. Таков набросок круга возникающих здесь проблем, и я им пока ограничусь.

2.5. Стилиевые файлы

Серьезным источником несовместимости является доработка стилевых файлов. Например, разные версии `amsrpt.sty` для $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX2.1 дают различный дизайн и, следовательно, об идеальной обработке файла в этом стиле не может быть и речи. Бывает и еще хуже: разработчики заслуженно популярного пакета MFPICT в каждой новой версии меняют синтаксис языка. Пользователю остается либо держать все версии и подбирать соответствующую, либо составить словарь и пытаться перевести старый файл на новый язык. По нашему мнению, разработчикам стилей необходимо учитывать возможность наличия у пользователей старых файлов.

3. Стандарт как иерархия рамок

Итак, идея стандарта как документа, предписывающего, какие форматы и стили можно использовать, не приведет к желанной цели даже при условии разработки соответствующих программных продуктов под всеми существующими операционными системами.

Рамки стандарта никогда не избавят пользователя ни от подготовленных по устаревшим правилам документов, ни от привлечения новых средств их подготовки. Роль их видится в правильной ориентации пользователей и разработчиков. Поскольку интересы и тех, и других весьма разнородны, основой всей работы могли бы стать их (интересов) изучение и систематизация. Так как нет и не может быть устраивающей всех единой рамки стандарта, можно ли заменить ее *иерархией* рамок? Иначе не удастся ликвидировать проблему несовместимости разных Т_ЭХовских расширений, сохранить для пользователей богатство электронных архивов, созданных за последние годы в различных Т_ЭХовских форматах, избежать быстрого устаревания стандарта. Разумной альтернативы не видно.

Проблема, таким образом, вылилась в задачу приемлемой организации системы стандартных описаний стандартных ограничений, гарантирующих переносимость подготовленных документов. Результат должен реализовываться на двух уровнях: описание комплекта вариантов; разработка, документирование и свободное распространение сопровождающих программных продуктов.

Бесспорно, дать полное решение проблемы за короткий промежуток времени невозможно. Однако я считаю возможным выдвинуть стратегию развития, которая при ее планомерной последовательной реализации неизбежно приведет к желаемому результату. Единственный, на мой взгляд, способ это осуществить состоит примерно в следующем. Стандарт складывается поэтапно по уровням. Первый уровень включает в себя наиболее популярные исходные форматы: Plain и Л^AT_ЭX 2.09. Каждый следующий уровень обеспечивает 100-процентно правильную обработку всех файлов, включенных в предыдущий уровень. Постепенно уровень стандарта должен стать настолько высоким, чтобы обеспечить корректную обработку всех лежащих в популярных электронных архивах файлов.

В принципе каждый из этих файлов и сегодня может быть корректно обработан, для этого лишь надо взять из архива СТАН соответствующие файлы, расположить их в правильных директориях и иногда подправить конфигурационные файлы. Кроме того, стандарт не должен заставлять пользователя искать среди более чем 50 000 файлов СТАН нужные и разбираться в тонкостях конфигурирования системы. Подключение правильных файлов и конфигурирование должно быть заботой системы, а не пользователя. Этот принцип переносит трудности, связанные с многоплановостью развития Т_ЭХа, с пользователя на разработчика интерфейса запускающей Т_ЭХ оболочки. При таком подходе пользователю становится доступным все более широкий круг возможностей и сохраняется вся накопленная ранее информация.

Многие с недоверием относятся к этой идее, поскольку даже для человека, много лет работающего с Т_ЭХовскими файлами, порой чрезвычайно трудно подключить и сконфигурировать все, что нужно для обработки конкретного файла. Тем более эту интеллектуальную работу не может выполнить та или иная оболочка. Наверняка в ее алгоритме окажутся скрытые ошибки, которые вместо помощи пользователю создадут ему еще большие трудности. Под разными операционными системами используются разные оболочки, что усугубляет недоверие.

Необходимо заметить, что на первом уровне (Plain и Л^AT_ЭX 2.09) нет никаких трудностей в правильном определении формата: незакомментированная команда `\begin{document}`

однозначно указывает на L^AT_EX. На каждом следующем уровне задача определения форматного файла и используемых версий некоторых стилевых файлов будет усложняться, и для ее решения придется разработать соответствующий инструментарий, обеспечивающий легкую и быструю переносимость алгоритма распознавания на другие операционные системы и оболочки. Кроме того, оболочка должна оставлять пользователю возможность уточнения конфигурации для обработки отдельного файла.

Предлагаемый подход уводит описание конкретных уровней стандарта с первого этапа работы, поскольку прежде должны быть в принципе решены следующие задачи:

- создание хотя бы одной корректно работающей оболочки, автоматически конфигурирующей систему на основе анализа входного файла;

- разработка языка описания алгоритма анализа входного файла и интерпретатора этого языка, пригодного для включения в оболочку;

- разработка качественного варианта русифицированного T_EXа и размещение его на сети в свободном доступе.

Перечисленные задачи решены в рамках проекта РФФИ “Эффективный способ адекватного представления научной информации в компьютерных системах России” (www.botik.ru/PSI/RFBR_TeX), дистрибутив русифицированного T_EXа, автоматически различающий Plain, AmS_TE_X, L^AT_EX 2.09, AmS_IL^AT_EX 1.1, L^AT_EX 2e и AmS_IL^AT_EX 1.2, размещен на анонимном FTP, ftp.botik.ru/pub/msdos/Russian_TeX_97. Дистрибутив включает в себя утилиты (bm2font, mfpic), конфигурирование которых вручную обычно затруднительно. Он содержит также новые, качественные русифицированные шрифты, и на нем в значительной степени будет основан уровень стандарта, включающий русифицированный T_EX.

Следующим этапом работы должно стать расширение системы до придания ей способности к автоматической обработке 100 % T_EXовских файлов, находящихся в электронных архивах мира и серверах России. Параллельно предстоит анализ и классификация реально используемых стилей, пакетов и установочных комплектов с целью выработки расширяемого формального языка, позволяющего лаконично и однозначно описывать:

- средства, необходимые для приблизительной обработки документа;

- средства, необходимые для корректной обработки документа;

- средства, необходимые для идеальной обработки документа;

- средства, предоставляемые вариантом установки T_EX.

Такой язык позволит провести паспортизацию существующих документов и вариантов установки T_EXа и описать требования к подготовке файлов, рассчитанных на различные круги потребителей. Разработанный на его основе дистрибутив при полной установке должен будет давать не только возможность обработать практически любой T_EXовский файл, но и возможность установить режим работы, отсекающий нестандартные (относительно выбранного сервера или журнала) средства и дающий рекомендации по их замене. Дистрибутив не должен содержать закрытых, коммерческих или недокументированных элементов. Только тщательное соблюдение этого принципа открытости может заинтересовать разработчиков будущих программных продуктов в поддержании и развитии этого стандарта в новом понимании — развивающегося стандарта.

Поступила в редакцию 24 апреля 1997 г.