СТАНДАРТИЗАЦИЯ РУССКОГО TEX: УТОПИЯ ИЛИ НЕИЗБЕЖНОСТЬ

С. В. ЗНАМЕНСКИЙ

1. Постановка задачи

Система TEX, разработанная Д. Кнутом в 1982 г., активно используется людьми, имеющими дело со сложными математическими формулами, как

- профессиональная издательская система;
- удобное средство подготовки книг, статей, препринтов;
- гибкий и ясный язык коммуникации специалистов по электронной почте;
- экономичный стандартный способ представления научной информации в сетевых архивах и базах данных.

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

К сожалению, люди, занимающиеся улучшением и разработкой утилит, далеко не всегда имеют в виду сохранение и упрочение 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 нарастает в связи с непрерывной разработкой новых версий LATEX, а для нищих российский пользователей, для которых TEX очевидно останется основным средством выживания еще на много лет, усугубляется еще и распространением трудно совместимых вариантов русификации.

Российским Фондом Фундаментальных Исследований предложена и поддержана идея фиксирования (не столько юридически, сколько фактически) стандарта представления информации в компьютерных системах России, базирующегося на русификации TEX. Реализация этой идеи создала 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.

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 должного эффекта по

РАБОТА ВЫПОЛНЕНА ПРИ ПОДДЕРЖКЕ РФФИ, ГРАНТ 95-07-19400В

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, что мало поможет в решении стоящей проблемы. 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.

Сколь большие средства потребовалось бы выделить 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? Трудно сказать. Подобный проект под названием “LATEX3” осуществляется уже в течение трех лет, в работе задействованы сильнейшие специалисты по TEXу, 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: при современных темпах развития компьютерной сферы даже если и возможно создать целенаправленно разработать 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 TEXа, 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, проявлений несовместимости, их причин и путей устранения.

2. Проявления и причины несовместимости используемых расширений TEX

2.1. Классификация уровней совместимости. Идеальное воспроизведение документа зачастую требует наличия именно тех версий используемых стилевого, форматного файла и шрифтов, с которыми 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 TEX 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. Кнутом и под названием "Plain" входит в большую часть установочных вариантов TEXа. Следующий форматный файл, разработанный Л. Лампортом, называется "Lplain" или "LATEX2.09". Он предоставляет автору богатый сервис автоматической нумерации формул, утверждений, заголовков, значительно облегчает составление общирной библиографии. В отличие от Plain, LATEX 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, выполненый в LATEXе легко узнать по блочной структуре документа, офрмленной рамками `\begin\{нечто\}` ... `\end\{нечто\}`.

К сожалению, в стандартных стилях LATEX2.09 крупный шрифт заголовка диссонирует с мелким шрифтом формулы, которая в этом заголовке может оказаться. Этот дефект являлся трудноустранимым: поскольку LATEX 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 LATEXа требовало чрезмерных машинных ресурсов.

Желание получать приличные заголовки привело к созданию так называемой NFSS — (New Font Selection Scheme), добавление которой к классическому LATEX2.09 приводит к формулам правильного размера как в заголовках, так и в примечаниях, рефератах и других элементах оформления. Попутно 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, `\sf\bf` в классическом LATEXе давало не жирную версию рубленного шрифта, как в NFSS, а стандартный жирный шрифт `\bf`. Такой LATEX с NFSS распространен в России в составе оболочки TIS (TEX 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 LATEXом, а дизайн документов, подготовленных в старом LATEXе, 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 с должной осмотрительностью.

Американским математическим обществом разработаны шрифты и форматный файл $\mathcal{AMS}$-TEX, предназначенные для оформления математических статей со сложными формулами. В отличие от LATEXа 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 LATEXе документ может начинаться командой `\documentstyle`, но в $\mathcal{AMS}$-TEXе она не является обязательной. Файл, подготовленный в $\mathcal{A}\mathcal{M}\mathcal{S}$-TEXе, обычно содержит строку `\input amstex`, хотя может корректно быть 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 $\mathcal{A}\mathcal{M}\mathcal{S}$-TEXе, что иногда исключает возможность обработки $\mathcal{A}\mathcal{M}\mathcal{S}$-TEXом 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 с зарубежного электронного архива. По идее разработчиков она решается заменой в начале обрабатываемого файла указания на стиль `amsppt` указанием на `amsppt1` и в простых случаях это срабатывает.

Стремление к объединению возможностей $\mathcal{A}\mathcal{M}\mathcal{S}$-TEXпо элегантному оформлению математических формул с LaTEXовской автоматикой привело к созданию двух форматных файлов: $\mathcal{A}\mathcal{M}\mathcal{S}$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 LaTEXу. По второму нет доступных описаний и мне не известно случаев использования его в России.

Оставляя до следующего пункта проблемы, связанные с русификацией, замечу, 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 LaTEX2e, который щедро одаряет пользователей новыми возможностями и новой головной болью.

Все предыдущие варианты LaTEXа объявлены устаревшими и не поддерживаются. Каждые полгода выходит новая версия LaTEX2e, которая действительна в течение года, после чего подлежит обновлению. Каждая из версий поддерживает старый LaTEX2.09 на том же примерно 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 LaTEX2e.

Новая версия $\mathcal{A}\mathcal{M}\mathcal{S}$LaTEXа(1.2) уже реализована не в виде форматного файла, а набором стилевых файлов над LaTEX2e и не поддерживает описанные в сопровождавшей старую версию $\mathcal{A}\mathcal{M}\mathcal{S}$LaTEXа документации по NFSS команды, перечисленные выше.

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

Удалось ли в новом LATEX2e 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 очевидно резко затрудняет верстку? Если добавить, что ни один из существующих сегодня на русском языке документов не сможет без предварительного редактирования быть обработан новым форматным файлом, а строки, которые при этом придется вставлять, не приведены ни в одном из существующих руководств по LATEXу, то станет ясно, что решения проблем массы российских пользователей от многоязычного LATEX2e быстро ждать не приходится.

2.3. **Варианты русификации.**

2.4. **Графические расширения.**

2.5. **Стилевые файлы.**

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.

Хочется, чтобы рамки стандарта ни в коем случае не отгораживали 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, то основой всей работы могли бы стать их изучение и систематизация. Может быть, коли уж нет и не может быть устраивающей всех единой рамки стандарта, заменить ее *иерархией* рамок? Как иначе ликвидировать проблему несовместимости разных TEXовских расширений? Как иначе сохранить для пользователей богатство электронных архивов, созданных за последние годы в различных TEXовских форматах? Как иначе избежать быстрого устаревания стандарта? Возможен ли лучший подход к стандартизации? Разумной альтернативы не видно.

Проблема таким образом вылилась в задачу разумной организации системы стандартных описаний стандартных ограничений, гарантирующих переносимость подготовленных документов. Результат должен реализовываться на двух уровнях:

- описания 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 в следующем. Стандарт складывается поэтапно по уровням. Первый уровень включает в себя наиболее популярные исходные форматы: Plain и LaTeX2.09. Каждый следующий уровень обеспечивает 100% корректную обработку всех файлов, включенных в предыдущий уровень. Постепенно уровень стандарта должен стать настолько высоким, чтобы обеспечить корректную обработку всех лежащих на популярных электронных архивах файлов.

Естественно, вы можете спросить, а причем здесь стандарт? Раз уж в принципе каждый из этих файлов и сегодня может быть корректно обработан, для этого лишь надо взять с архива CTAN соответствующие файлы, расположить их в правильных директориях и иногда подправить конфигурационные файлы. Я еще не сказал, что стандарт не должен заставлять пользователя ни искать среди более чем 50 000 файлов CTAN нужные, ни 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 TeX, с плеч пользователя на плечи разработчика интерфейса запускающей TeX 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 TeXовскими файлами, порой чрезвычайно трудно подключить и сконфигурировать все, что нужно для обработки конкретного файла. Как такую трудную интеллектуальную работу может выполнить какая-то оболочка? Наверняка в ее алгоритме окажутся какие-то скрытые ошибки, которые вместо помощи пользователю создадут ему еще большие трудности. Под разными операционными системами используются разные оболочки, что усугубляет недоверие.

Замечу, что на первом уровне (Plain и LaTeX2.09) нет никакой трудности в правильном определении формата: незакомментированная команда `\begin{document}` однозначно указывает на LaTeX. 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 для обработки отдельного файла уточнить конфигурацию.

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 TEXа 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 TEXа, автоматически различающий Plain, AmSTEX, LATEX2.09, AmSLATEX1.1, LATEX2e и AmSLATEX1.2, размещен на anonymous 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 TEX.

Следующим этапом работы должно стать расширение системы до придания ей способности к автоматической обработке 100% TEXовских 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 TEX

Такой язык позволит провести паспортизация существующих документов и вариантов установки TeX 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 TEXовский 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 даст разработчикам будущих программных продуктов заинтересованость в поддержании и развитии этого стандарта в новом понимании - развивающегося стандарта.