The Multilingual Lion: \TeX\ learns to speak Unicode

Jonathan Kew
SIL International

April 7, 2005
Background

- **T\TeX**: free typesetting system with a 25-year history
  - stable, reliable, flexible, widely implemented
  - experienced user community
  - rich collection of supporting tools
- Originally designed for English typesetting
  - support for accents and other European characters
  - language support extended via custom fonts, macros, and preprocessors
Traditional \TeX input conventions

- Input text is ASCII (or 8-bit codepage)

<table>
<thead>
<tr>
<th>Source text</th>
<th>Typeset output</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>'{\textipa{a}}</td>
<td>á</td>
<td>typical accent command</td>
</tr>
<tr>
<td>\c{\textipa{c}}</td>
<td>ç</td>
<td></td>
</tr>
<tr>
<td>\textipa{aa}</td>
<td>å</td>
<td></td>
</tr>
<tr>
<td>\textipa{---}</td>
<td>-</td>
<td>ligature in typical \TeX fonts</td>
</tr>
<tr>
<td>\textipa{$\alpha$}</td>
<td>\alpha</td>
<td>math mode symbol</td>
</tr>
<tr>
<td>\textipa{{\textipa{dn acchaa}}}</td>
<td>अच्छा</td>
<td>using custom preprocessor</td>
</tr>
</tbody>
</table>
Multilingual typesetting with \TeX

- Text input
  - Escape sequences for non-ASCII characters
  - Multiple 8-bit codepages
  - Preprocessors for complex scripts

- Font support
  - Fonts limited to 256 glyphs
  - Custom-encoded fonts with specific glyph sets

- All tied together via complex \TeX\ macros
  - Difficult to understand and extend
  - Difficult to integrate with other packages
Towards a cleaner solution

- Unicode: all required characters directly represented
  - no need for “escape sequences” to access characters not included in the current codepage
  - no need to switch between codepages according to the language/script being typeset
  - characters rendered via standard access codes

- Character/glyph model and modern font rendering technologies
  - complex script handling moved out of the domain of the text data stream
The Multilingual Lion: \TeX learns to speak Unicode

Typesetting Unicode text with \TeX

- Accented characters

\begin{verbatim}
\halign{#\hfil\quad& #\hfil\cr
dan& dan\cr
dubok& dubok\cr
džabe& đak\cr
džin& džabe\cr
Džin& džin\cr
dak& Džin\cr
Evropa& Evropa\cr}
\end{verbatim}
The Multilingual Lion: \TeX learns to speak Unicode

Typesetting Unicode text with \LaTeX

- CJK ideographs

```latex
\font\han="STSong" at 16pt
\font\rom="Gentium" at 8pt
\def\hc#1#2{\vtop{\hbox{\han #1}
  \hbox{\kern10pt\rom #2}}}
\vtop{\hc{書く}{ka-ku}
  \hc{最も}{motto-mo}
  \hc{最後}{sai-go}
  \hc{働く}{hatara-ku}
  \hc{海}{umi}}
```

書く
ka-ku

最も
motto-mo

最後
sai-go

働く
hatara-ku

海
umi
The Multilingual Lion: \TeX{} learns to speak Unicode

Typesetting Unicode text with \LaTeX{}

• Complex scripts

\begin{verbatim}
\c 1
\p
\e نیمز ادخ م تاعورش 1
\v
\e ترتيب نیمز تقو نا 2
\v
\e پیام نام دادنوا وچاثم جو ادخ نامتم چچ کئان
\e یک یکپ اریق حور چی
\e یشور \\
\v
\e ونذ محج ادخ نهذت 3
\e یکپ یکشور وس “. یائت
\end{verbatim}
The Multilingual Lion: \TeX{} learns to speak Unicode

Key changes from \TeX{} to X\TeX{}

- Unicode as the text encoding
  - directly use Unicode input text, Unicode-encoded fonts
- Fonts and rendering technologies
  - use any fonts available in the host computer
  - use existing smart-font rendering systems
- Additional features for multilingual typesetting
  - optional font features
  - line breaking for Asian scripts
- Backward compatibility issues
  - support for legacy \TeX{} fonts and documents
From 8 to 16 bits...

- Character type in \TeX{} code was 8-bit value
  - one option: process text as UTF-8
- Character codes used to index a number of tables
  - character category, case pairs, etc.
- Decision to use 16-bit character codes
  - all 256-element tables enlarged to 65,536 elements to match the extended character set
  - extended \TeX{} commands that refer to character codes
From 8 to 16 bits... and beyond?

- Unicode does not fit in 16 bits either!
- `\TeX` handles non-BMP characters as UTF-16 surrogate pairs
  - properties of individual characters cannot be set
  - unlikely to matter for typesetting usage: all surrogate codes can be treated as simple printable characters
  - keeps size of internal tables moderate, without extensive restructuring
- Using UTF-16 happens to match the font rendering APIs that `\TeX` uses
Implementing the character/glyph model

- Required for support of complex scripts in Unicode
- Significant change from traditional TEX model
  - TEX regards “a specific character code in a specific font” as the fundamental unit of text to be typeset
  - assumes such a character has known, fixed dimensions
  - provision for ligatures by character substitutions
  - a paragraph consists of sequence of “character” nodes, to be precisely placed, and intervening “glue” nodes
- A Unicode character may not map to a single, known glyph
  - many scripts require contextual selection of glyphs
  - must measure characters in context, not in isolation
Implementing the character/glyph model

• Initial implementation using ATSUI on Mac OS X
  • typesetting process collects runs of characters (words)
  • calls ATSUI text layout APIs to measure width
  • a \TeX\ paragraph consists of sequence of “word” nodes separated by “glue”

• Typesetting engine positions words, not glyphs
  • this is the job of the font rendering engine
Implementing the character/glyph model

Nodes in a \TeX\ paragraph

- char: T
- char: h
- char: e
- glue: word space
- char: q
- char: u
- char: i
- char: c
- char: k
- glue: word space
- char: f
- char: o
- char: x
- glue: word space

Corresponding nodes in \Xe\TeX\

- word: The
- glue: word space
- word: quick
- glue: word space
- word: fox
- glue: word space
Implementing the character/glyph model

- OpenType Layout support using ICU library
  - alternative font layout engine
  - provides support for OpenType features in Latin fonts
  - supports a number of complex (Indic/Asian) scripts
- \TeX uses either ATSUI or ICU according to layout tables found in fonts
  - overall typesetting process is independent of font technology in use
  - distinction required only at lowest level of measuring a run of text in a given font
  - documents may freely mix AAT and OT fonts
Implementing the character/glyph model

- ATSUI APIs used in typesetting
  - ATSUCreateStyle, ATSUSetAttributes
  - ATSUCreateTextLayout, ATSUSetTextPointerLocation, ATSUSetRunStyle
  - ATSUGetUnjustifiedBounds, ATSUDrawText

- ICU APIs used in typesetting
  - ubidi_open, ubidi_close, ubidi_setPara, ubidi_getDirection, ubidi_countRuns, ubidi_getVisualRun
  - LayoutEngine::layoutChars, getGlyphs, getGlyphPositions
Hyphenation support

- Paragraphs formed of lists of “word boxes”
  - treated as indivisible units in the token list
  - allows TeX to remain unaware of low-level details
- If acceptable line breaks not found, hyphenation required
  - extract text characters from word nodes
  - find hyphen positions using TeX’s algorithm
  - repackage words as word fragments and discretionary break nodes
The Multilingual Lion: \LaTeX{} learns to speak Unicode

Hyphenation support

• Modifying the node list to allow hyphenation

Two glue different glue foxes
Two glue dif hyphen? fer hyphen? ent glue foxes

• Problem: unused hyphen points break rendering

Two glue dif fer -
ent glue foxes

Two different foxes

• Need to re-merge word nodes after choosing breaks

Two glue differ-
ent foxes

Two different foxes

27th Internationalization and Unicode Conference
Berlin, Germany, April 2005
Advanced font features

- OpenType language systems
  \texttt{\font\Doulos="Doulos SIL/ICU"}
  \texttt{\font\DoulosViet="Doulos SIL/ICU:language=VIT"}

Unicode cúng cấp một con số duy nhất cho mỗi ký tự

Unicode cúng cấp một con số duy nhất cho mỗi ký tự

\texttt{\font\Brioso="Brioso Pro"}
\texttt{\font\BriosoTrk="Brioso Pro:language=TRK"}

... gelen фирмалары ... tarafindan ...

... gelen фирмалары ... tarafindan ...
Advanced font features

- Custom AAT features

\font\Doulos="Doulos SIL/AAT"
\font\DoulosAlt="Doulos SIL/AAT:
Alternate forms=Literacy alternates,
Small \v-hook straight style;
Uppercase Eng alternates=Capital N with tail"

Xosphate na Mose ḷo
Njutitotoŋkeke la anyi,
eye wôna wohlë vu ṣe
votrutuwo ŋu bene dòla
si atsrñ ŋgɔgbeviwo la
nagawɔ nuvevi Israel
viwo ya o.

Xosphate na Mose ḷo
Njutitotoŋkeke la anyi,
eye wôna wohlë vu ṣe
votrutuwo ŋu bene dòla
si atsrñ ŋgɔgbeviwo la
nagawɔ nuvevi Israel
viwo ya o.
East Asian languages

- Line breaking without word spaces
  - \TeX{} normally breaks lines at “glue” arising from spaces
  - Chinese, Japanese, Thai, etc. do not use word spaces

  โดยพื้นฐานแล้ว, คอมพิวเตอร์จะเกี่ยวข้องกับเรื่องของตัวเลข. คอมพิวเตอร์จัดเก็บ
  โดยการกำหนดหมายเลขให้สำหรับแต่ละตัว. ถนนหน้าที่ Unicode จะถูกสร้างขึ้น, ได้มีระบบ encoding อยู่หลายร้อยระบบสำหรับการกำหนดหมายเลขเหล่านี้.

- Use ICU line-break: \texttt{\LaTeX{}linebreaklocale "th"}

  โดยพื้นฐานแล้ว, คอมพิวเตอร์จะเกี่ยวข้องกับเรื่องของตัวเลข. คอมพิวเตอร์จัดเก็บ
  ตัวอักษรและอักขระอื่นๆ โดยการกำหนดหมายเลขให้สำหรับแต่ละตัว. ถนนหน้าที่ Unicode จะถูกสร้างขึ้น, ได้มีระบบ encoding อยู่หลายร้อยระบบสำหรับการกำหนดหมายเลขเหล่านี้.
Backward compatibility

- Legacy \TeX{} fonts, especially for math mode
  - supported via \TeX{} font metrics and Type 1 font files
  - allow many existing \TeX{} documents to work
  - not Unicode-compliant!

\[
\left( \int_{-\infty}^{\infty} e^{-x^2} \, dx \right)^2 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-(x^2+y^2)} \, dx \, dy \\
= \int_0^{2\pi} \int_0^\infty e^{-r^2} r \, dr \, d\theta \\
= \int_0^{2\pi} \left( -\frac{e^{-r^2}}{2} \bigg|_{r=\infty}^{r=0} \right) \, d\theta \\
= \pi.
\]
Backward compatibility

- Non-Unicode input text
  - by default, input read as Unicode (UTF-8 or UTF-16)
  - legacy codepages supported via ICU converters
  - set codepage of current input file:
    \XeTeXinputencoding "charset-name"
  - set initial codepage for newly-opened input files:
    \XeTeXdefaultencoding "charset-name"
Backward compatibility

- Support for legacy keying practices
  - typical input:
    ```\TeX'```---a typesetting system
  - generates: ```\TeX''```---a typesetting system

- Font mapping for compatibility

  ; TECKit mapping for TeX input conventions
  U+002D U+002D  <>  U+2013 ; --  ->  en dash
  U+002D U+002D U+002D  <>  U+2014 ; ---  ->  em dash
  U+0027  <>  U+2019 ; '  ->  right single quote
  U+0027 U+0027  <>  U+201D ; ''  ->  right double quote
  U+0022  >  U+201D ; ''  ->  right double quote

  - generates: ```\TeX''```—a typesetting system
More fun with font mappings

\def\SampleText{Unicode -
   это уникальный
   код для любого символа,
   независимо от платформы,
   независимо от программы,
   независимо от языка.}
\font\gen="Gentium"
\gen\SampleText
\bigskip
\font\gentrans="Gentium:
   mapping=cyr-lat-iso9"
\gentrans\SampleText

Unicode - это уникальный код для любого символа, независимо от платформы, независимо от программы, независимо от языка.

Unicode - èto unikal'nyj kod dlâ lûbogo simvola, nezavisismo ot platformy, nezavisismo ot programmy, nezavisismo ot âzyka.
**Xe\TeX** and other \TeX\ extensions

- **\TeX\G\X**
  - a direct ancestor of \Xe\TeX, but now obsolete

- **e-\TeX**
  - basis of current \Xe\TeX\ implementation
  - provides a number of features, especially bidi support

- **Omega, Aleph**
  - ambitious project to extend \TeX\ to all scripts
  - complex configuration, no direct smart-font support

- **pdf\TeX**
  - widely-used extension providing rich PDF support
  - no native Unicode or smart-font support
The Multilingual Lion: \TeX{} learns to speak Unicode

For more information

- \TeX{} web site and mailing list
  - http://scripts.sil.org/xetex
  - http://tug.org/mailman/listinfo/xetex

- Contact information
  - mailto:jonathan_kew@sil.org

- Questions... and answers?