This extended abstract demonstrates how extending \TeX to natively handle the Unicode character set greatly simplifies the task of multilingual and multi-script typesetting. Because all characters of all the world’s scripts are included in a single standard, it is not necessary to convert external encodings to a special internal representation, or to manage multiple input encodings for different languages, and any combination of scripts and languages can be freely mixed in a single document — even in a single line of text. The \texttt{Xe\TeX} extension of \TeX makes it simple to use Unicode throughout, from input text to hyphenation tables and font access.

In addition to adopting Unicode as the standard character encoding, \texttt{Xe\TeX} has built-in support for modern font technologies (TrueType, OpenType, AAT), including glyph layout behavior defined in font tables. This means that complex scripts such as Indic and Arabic can be typeset with no special font setup and configuration. For example, using an off-the-shelf Arabic font, whether from a major vendor or a free font developer, involves no complex conversion processes or the creation of an “alphabet soup” of .tfm, .vf, .ocp, .map, .enc, .fd, etc. files; just drop the .otf or .ttf file into the computer’s Fonts directory, and select the typeface in a \TeX document.

Including Arabic in a \LaTeX document can then be as simple as declaring the font to be used:

```latex
\usepackage{fontspec}
\newfontinstance{arfont}{Scheherazade}
% for in-line Arabic we need R-L control
\newenvironment{ar}{\arfont}{%}
```

To include the Arabic script in a document, we can just enter \begin{ar} \ldots \end{ar} in the source text, with Unicode Arabic text within the ar environment (not shown here because \texttt{cmte} does not include Arabic characters).

For extended passages of Arabic, one additional factor needs to be taken into account: the overall paragraph direction should be made right-to-left, so that the paragraph indent and alignment of the last line behave as expected:

```latex
% simple environment for R-L paragraphs
\newenvironment{ArabicPar}{\everypar={\setbox0\lastbox \beginR \box0 \arfont}}{}%\newenvironment{ArabicPar}{\everypar={\setbox0\lastbox \beginR \box0 \arfont}}{}% This environment allows Arabic paragraphs to be properly laid out, as in figure 1.

Because \texttt{Xe\TeX} uses Unicode text and fonts, rather than a complex collection of macros to provide the script support, it is trivial to include other scripts such as Japanese, Devanagari, or many others in the same document. All we need is an appropriate Unicode font that covers the required character repertoire:

```latex
% Japanese, with proper line-breaking
\newfontinstance{japfont}{Hiragino Kaku Gothic Pro}
% Hindi
\newenvironment{Hindi}{\devfont}{%}
```

With these declarations, we can set Japanese and Hindi just as easily as Arabic. Figure 2 shows two examples using fonts included as standard with
Figure 2: Japanese and Hindi text set by Xe\TeX

\[
\begin{align*}
\alpha &= f(z) \\
\beta &= f(z^2) \\
\gamma &= f(z^3)
\end{align*}
\]

\[
\begin{align*}
x &= \alpha^2 - \beta \\
y &= 2\gamma
\end{align*}
\]

\[
p_1(n) = \lim_{m \to \infty} \sum_{\nu=0}^{\infty} \left(1 - \cos^2 m(\nu^n \pi/n)\right)
\]

Figure 3: Math displays in Computer Modern (with custom encodings and multiple fonts) and Cambria Math (a single Unicode font, with no .tfm, etc.), typeset from the same source text.

Mac OS X; similar results are obtained with OpenType fonts available on Windows, GNU/Linux, and other systems.

A more thorough implementation of script and language switching should of course also change hyphenation patterns, quote-mark styles, and other typographic niceties according to the language in use. These minimal examples show how easily multilingual fonts can be used; producing high-quality typography in varying scripts may require additional refinements.

Ongoing work on Xe\TeX{} includes some experimental features to support the use of OpenType math fonts, which can contain a huge collection of math alphabets (italic, bold, blackboard, fraktur, script, etc.) and symbols, all encoded according to the Unicode standard. Forthcoming Microsoft products will include the Cambria Math font, and other projects such as the STIX fonts can be expected to support the same OpenType standard for math metrics. Xe\TeX{} aims to be able to use such fonts directly, without needing to create custom-encoded subfonts, .tfm files, etc., and the current status of these features will be demonstrated. A couple of examples from The \TeX{}book are shown in figure 3, in both the original Computer Modern and Unicode-compliant Cambria Math fonts.