

# **“Smart” Font and Shaping Systems for Complex Asian Scripts**

Christopher Fynn

*Talk given at [PanL10n](#) workshop, National University of Computer and  
Emerging Sciences, Lahore Pakistan, 20 January 2004*

## **What is a "complex script"?**

Asian scripts derived from Arabic and Brhami are called “complex scripts” since:

- the logical and visual order of characters may be different (requiring reordering)
- Arabic text is read from right to left, but numbers are read from left to right (requires bi-directional rendering)
- the form and position in which underlying nominal characters should be rendered is variant and dependant on the context of the characters (requiring context dependant glyph substitution and positioning)
- sequences of characters often form complex ligatures (requiring many to one glyph substitution and composition)
- some characters may have multiple glyph parts which are split e.g. the Tamil short-O vowel (requiring one to many decomposition)

## **Leagacy “font based” encoding**

Legacy computer systems had a one-to-one mapping between data characters and the glyphs or letterforms in the font being used. These systems used “font-based” or “glyph based” character encodings - where a different character represents each form of a single letter. In Arabic fonts, isolate, initial, medial and final forms of letters were mapped to a different characters. In Devanagari fonts, full and half forms of consonants had separate code points and in Tibetan different forms of a single vowel (eg *zhabs-skyu*) were mapped to a separate data characters. Characters with two parts [e.g. Tamil short-O vowel or Tibetan Indic RII vowel] required a separate character for each part.

At the same time in these systems complex ligature glyphs - representing conjuncts of several consonants (or several nominal characters), were mapped to single characters.

By thier nature these systems place restrictions on designers of fonts made to operate operate on them, since the glyphs or elements in such a font have to match the analysis of the scripts "shaping" behaviour determined at the time the application was written or the encoding designed. Often such an analysis is only appropriate for a single form or style of a script. With complex writing system this leaves little room to design a wide variety of fonts for different styles of the same script - and variant fonts have to have very similar metrics.

## **“Smart” fonts**

Modern font and shaping systems designed to render UCS characters for complex scripts are built on a modular design where there is a clear distinction made between nominal characters and the glyphs used to render them. The application environment provides a

library routines with an API that applications can use to render complex script text. These routines in turn apply tables of rules,- contained in the font itself - to reorder characters, map from characters to glyphs, make contextual substitutions and position marks and other glyphs in relation to each other. Since these tables are part of the font the font developer has maximum control over the shaping behaviour of the glyphs in his font.

### **Different smart font formats:**

There are currently three slightly differing technologies and three corresponding font formats using this modular approach:

- 1) Microsoft & Adobe's OpenType
- 2) Apple's AAT and
- 3) SIL's Graphite.

The individual font formats used by all these systems are based on the TrueType font format. Essentially each system places an extended set of tables in a TrueType font file containing rules and lookups the rendering system can apply in order to display the underlying text correctly using that font. Since the TrueType font format has extensible tables, and a font developer can add any number of them to a font, it is theoretically possible (but a great deal of work) to create a single font with the different tables necessary to for that font to work with all three systems.

### **OpenType fonts & Uniscribe**

OpenType was in an earlier incarnation called TrueType Open. This was introduced by Microsoft in about 1994 as an enhancement to the TrueType font format specification with specifications for extra tables intended to support complex scripts. However Microsoft didn't release any software that actually took advantage of these extra tables. While it may have been possible for third-party developers to build their own rendering system and applications do do this, none did and consequently there was no real take up of this technology at the time.

In 1999 TrueType Open was re-incarnated as OpenType when Microsoft & Adobe announced the release of the OpenType font format. The table specifications for complex scripts remained essentially the same but additional lookups were specified to handle "fancy" Latin typography (of interest to Adobe's customers). The main technological difference between TrueType Open and OpenType is that OpenType fonts can contain glyph outlines in either PostScript or TrueType format.

With the release of Windows 2000 Microsoft finally introduced software in the form of *Uniscribe*, a Windows system-level component dll, which is a "shaping-engine", designed to use OpenType font features for complex script layout. This is an integral part of the Microsoft Windows 2000 and XP operating systems. At about the same time the applications Internet Explorer 5 and Office 2000 designed to use the features provided by Uniscribe were released.

However complex script support in Uniscribe is implemented on a script-by-script basis with a particular sub-set of OpenType features being applied in a particular order to each script. The initial version of Uniscribe supported only the Arabic, Devanagari & Tamil scripts. An updated version of Uniscribe shipped with Windows XP and Office XP but, while this version implemented support for additional complex scripts, it still lacked full support for many scripts.

Beta versions of the Uniscribe dll (USP10.DLL) with support for some additional scripts are obtainable through Microsoft's beta testing program and with their OpenType font development tool VOLT . It is possible to update Windows 2000 and Windows XP with this version of the Uniscribe DLL. However the version of Microsoft Windows expected to ship complete with built-in support for these additional scripts, currently known by the code name "Longhorn", will still not be released for some time.

Support for complex scripts and OpenType is also being implemented in Linux and other Open Source Operating systems through code from open source projects such as IBM's "International Components for Unicode" or ICU Library, Pango and FreeType Layout (FT Layout). These essentially provide the same services Uniscribe provides in MS Windows. Already there is more than one distribution of Linux available with the whole desktop translated into Hindi with the Devanagiri script and an Urdu distribution of Linux has been developed by

## **ATSUI & AAT fonts**

In 1999, Apple introduced Apple Type Services for Unicode Imaging or ATSUI which is now the basis for all Unicode text rendering in the MAC OSX system. The corresponding font format, Apple Advanced Typography or AAT, is the successor to their Quickdraw GX font technology.

With ATSUI all the information necessary for rendering a script is built into the font by its developer in the form of a table based "state engine". While this means that font developers do not have to wait for Apple to implement support for particular scripts – the tables in AAT fonts are necessarily far more complex. Since OSX is quite new there are currently few applications available that actually take advantage of ATSUI/AAT layout

AAT/ATSUI is also supported in IBM's open source ICU (International Components for Unicode) library that is being used to implement support for AAT/ATSUI under Linux and other open source operating systems.

## **SIL's Graphite**

Graphite was developed by The Summer Institute of Linguistics' Non-Roman Script Initiative who had a requirement for a system with which they could implement standards-based support for lesser-used scripts uncatered for by major software vendors. The name Graphite applies to both the font technology and the rendering software

Font developers can write script descriptions and lookup tables similar to those in AAT fonts using a high level language and compile these into a font using tools also developed by SIL.

At this time the only Graphite-aware applications available are the text editor WorldPad and the Mozilla web-browser. SIL has recently released the Graphite source code under the Free Software Foundation's GNU General Public Licence and there is considerable interest in Graphite in the Linux community. The recently announced SILGraphite open source project aims to stimulate 3rd party development (applications and fonts).

## **Which system to use?**

The most practical of these systems is OpenType since it is currently well supported on both Windows and Linux, and there are fairly user friendly tools available to create OpenType fonts. The lookups in OpenType fonts are also far less complex and easy to create. However, due to the number of staff Microsoft have available to work on new

scripts, lengthy testing procedures, and the time between releases of new versions of MS Windows - support for the particular script you use might not be available for quite some time - at least on that operating system. Although there may be a long waiting time the advantage is when support appears for the script you need it should have been thoroughly tested.

If you need support for your script today and have the necessary skills, you can update the code (for example) in Pango or Qt to support your script and make that available on Linux under GNOME or KDE. The risk you take here is that you will have to do the testing and QA yourself and when Microsoft finally implements and ships support for your script in Uniscribe it may be implemented differently from what you have done. Naturally most people want their fonts to work on both systems - so this might entail modifying the code you've added, re-working the font tables and distributing a new version of your fonts at a later date.

Once there are more applications available which make use of AAT/ATUSI under OSX if your requirements were limited to say a particular organisation like a newspaper publisher it might make sense to use this system. But since Apple currently only occupies a tiny and specialised market share in South and South East Asia, in my opinion it is currently not a practical solution to meet the needs of the larger general userbase in this area.

AAT/ATUSI has some support under Linux so, if you plan a Linux only solution, and you require a little more flexibility than that provided by OpenType it might be worth exploring further.

SIL's Graphite seems designed particularly to offer support for minority scripts which are not yet encoded in the ISO 10646 / Unicode Standards and for custom applications which may have requirements beyond those provided by OpenType.

## **Sources & further information:**

- [Examples of Complex Rendering](#)
- [Rendering Technologies Overview](#)
- [OpenType Specification](#),
- [AAT & ATSU](#)
- [Graphite](#)
- [TrueType Font File Specification](#)

- Chris Fynn  
[cfynn(at)gmx.net]