Complex Text Layout Issues
with examples from Myanmar

Keith Stribley

Graphite Font Integration / Myanmar Localisation

Overview

- Background of Complex Text Layout

  - Requirements for Myanmar Languages

- Integration of Graphite Font support into OpenOffice

  - Demo and examples

- Font Features

  - Benefits and examples

  - Language features

- Line breaking issues

This presentation requires OpenOffice with Myanmar patches, Graphite Font support and Burmese fonts.

Complex Text Layout

- Many South East Asian Languages require Complex Text Layout features such as:
    - Reordering of glyphs
    - Context specific ligatures
    - Context specific positioning
- These need to be implemented in a complex rendering engine such as:
    - OpenType – already in OOo through Uniscribe/ICU
    - SIL Graphite – in graphite01 CWS
    - Apple Advanced Typography (AAT)

Complex Rendering Example for Burmese

- 1 to 1 Unicode to glyph mapping
  - Unreadable without renderer
- Correct rendering
  - Reordering
  - Ligatures
    - 103C 102F
    - 1004 103A 1039 102D
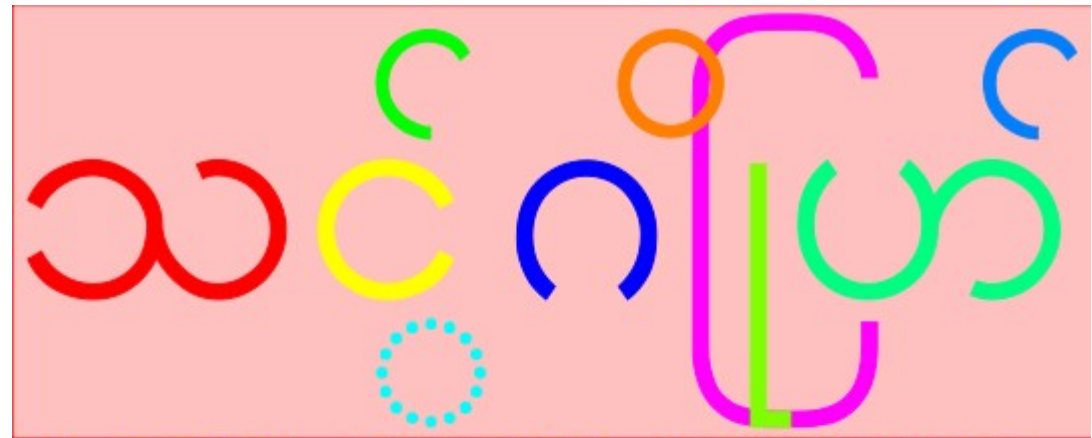
101e 1004 103a 1039 1002 103c 102d 102f 101f 103a
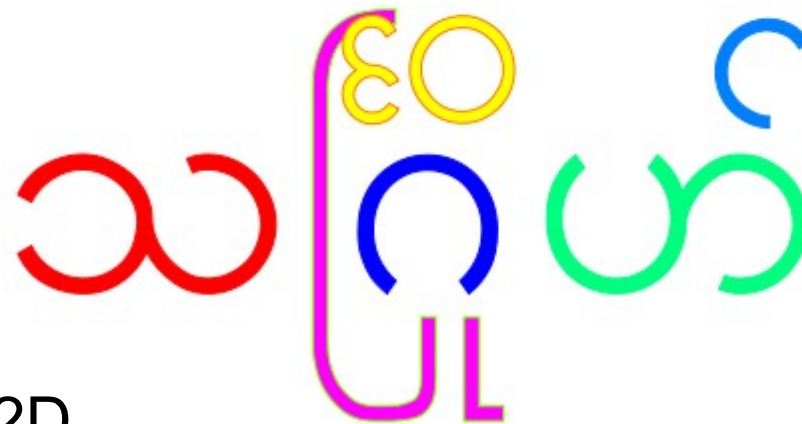
Benefits of Graphite Fonts

- The reordering, ligature and positioning rules are all stored in the font
  - The rendering engine does not need to be modified for each new CTL script
  - Basic Line breaking rules stored in the font
    - no need for source code modifications to application
  - The rules are quick to write in the Graphite Description Language
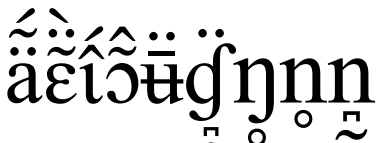- The Graphite library is cross-platform
  - Consistent rendering across platforms

Limitations of OpenType for newly encoded scripts

- The Default OpenType features applied to all languages are limited
  - e.g. no many to many substitutions – making reordering hard to implement
- For optimal performance the OpenType renderer has to be modified for each script
  - this hasn't been done for Myanmar yet (though some effort is under way for ICU and HarfBuzz)
- There are discrepancies between the feature implementations in Uniscribe/ICU/HarfBuzz
  - e.g handling of marks
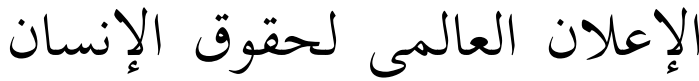
Demonstration of Graphite enabled OpenOffice

- Graphite has been integrated into the font rendering code in the VCL module
  - In graphite01 CWS

- Doulos SIL

  ặɛ̂ɔ̂ɓɗŋɲɲ

  k͡p m͡i l͡w l͡i m͡w ɛ̈ ɔ̈

- Padauk (Myanmar)

  ပိတောက်နဲ့ မြန်မာအက္ခရာ ရေးနိုင်သည်။ မြို့ကြီးမှာ ကွန်ပျူတာသုံးသူများသည်။

- Pig Latin Font

  Igpay Atinlay Emod

- Scheherazade (RTL)

  الإعلان العالمي لحقوق الإنسان

Font Features

- Features allow the user to modify the rendering of a font without changing to a different font

  - Many people prefer a slightly different rendering of a character to the default

  - They apply to a run of text in a similar way to bold/italic attributes

- Features provide many more options to the user than separate fonts

  - Not practical to provide a separate font for every feature combination e.g. Doulos has 37 features: $>2^{37}$ possibilities, though not all will be useful.

Graphite Features in OpenOffice (preliminary)

- Implemented by appending to the font name
  - Required no change to ODF format

Font name/feature setting	Doulos SIL	Doulos SIL:1039=1&1024=2
Rendering	ꞄŊ	ꞄN
1039 Capital R-tail alternate	Upper case	Lower case
1024 Uppercase Eng alternates	Large eng with descender	Capital N with tail

- ":" separates font name from features
- "&" delimits features
- (";" "," already used as alternate font delimiters)

Language Features

- Features are commonly used to customize a font's appearance to the preferences of a specific language community

  - Several features may be involved, so it is easier to just specify the language and the rendering engine will select the relevant features

- Language needs to be passed to VCL font code

  - Font object language field wasn't being set

  - Many minority languages in ISO639-3 don't have a corresponding Microsoft Locale ID or are not currently selectable from the language menu

Language Feature Examples

- Vietnamese has diacritics off centre, unlike other languages:

  - ề ắ ễ ố ậ ấ (Doulos SIL, no features applied)

  - ề ắ ễ ố ậ ấ (Doulos SIL:1029 = 1/Doulos SIL:lang = vie/OOo

    language = Vietnamese)

- Some Burmese prefer ကို to have a tall u ကို

- Burmese prefers ကဲ့ ပု Karen prefers ကဲ့ ပု

Improving font feature support

- Features need to be added to ODF for Graphite, OpenType and AAT

  - IDs are numeric or 4 char tags, values are numeric

  - e.g. `<style:text-properties ...`
    ```
    style:font-features="1039=2 ulon=1"
    style:font-features-complex="hsln=2"
    style:font-features-asian="3=2"/>
    ```

- Need a proper GUI to set features in Character/ Font dialog

  - Graphite stores localized feature names in the font

  - OpenType and AAT have registries maintained by Microsoft and Apple respectively

Line Breaking in Myanmar

- Burmese uses spaces for phrase breaks, not word breaks
  - Users can use ZWSP, but it is hard to train users
  - Automated approaches:
    - Syllable based algorithm often sufficient, but breaks multisyllable words
      - Padauk's Graphite rules have syllable based breaking
      - Syllable based breaking can be added as a patch to ICU
    - Dictionary based algorithm
      - A combination of dictionary lookup and syllable parsing can be used to avoid breaks in mulitsyllable words e.g. by patching ICU
      - Only as accurate as the data in the dictionary

Line breaking example using an ICU based dictionary

- မောင်မောင်အလွန်ကြိုးစားသဖြင့်ကန့်ထရိုက်ထာဖြစ်သွားသည်။ Padauk

- မောင်မောင်အလွန်ကြိုးစားသဖြင့်ကန့်ထရိုက်ထာဖြစ်သွားသည်။

  Myanmar3

- Need 3 Levels:
  - Dictionary (word level)
    - Uses syllables
    - Can't handle rare or loan words
  - Syllable
    - To break long words
    - Not handled by break iterator
    - Font e.g. graphite (writer)
  - Character
    - Last resort

Summary of Outstanding Myanmar Layout issues

- Graphite Font support

  - CWS needs to be integrated

- Font Features need:

  - Support in ODF

  - A proper user interface

  - To be selectable by ISO 639-3 language

- Line Breaking

  - Currently requires patching of ICU for Myanmar

  - May require intermediate level between Word / Character breakiterator similar to hyphenation