L2/17-334
WG2 N4890

On migration issues of the graphetic model

关于字形模型的迁移问题

Script Ad Hoc Group
文字特别小组

18 September 2017
2017 年 9 月 18 日

Before reading this supplementary document please first read the main document L2/17-328 *Script Ad Hoc Group Recommendations on Mongolian Text Model*. This document discusses migration issues and solutions of the graphetic model for the Mongolian encoding, focused on the modern Hudum.

在阅读本补充文档前请首先阅读主文档 L2/17-328《文字特别小组关于蒙古文文本模型的建议》。本文档针对现代胡都木文讨论蒙古文编码字形模型的迁移问题与解决方案。

1 General strategy 总体策略

In the graphetic model, text will be encoded as graphetic characters for graphemes, instead of phonetic characters for underlying phonetic letters in the current model. New characters will be required to be encoded while some of current characters are to be deprecated. This implies a significant migration that will have to be planned for.

在字形模型中，文本会为字位而编码为字形字符，而不是像现行模型这样为底层语音字母而编码为语音字符。新的字符会需要编码，而一些现有的字符会被废弃。这意味着必须对重大的迁移有所准备。

Multiple uninteroperable vendor implementations of the current model currently coexist in the industry. See a list of major vendors in appendix A on page 10.

业界目前共存有现行模型多个不可互通的厂商技术实现。参见第10页附录A的主要厂商列表。

Two general strategies can be considered:

两个总体策略可考虑：

- **Transitional:** First migrate to an improved version
- **过渡：**首先迁移到现行模型的一个改良

of the current model, then migrate to the graphetic model.

- **Direct:** Migrate to the graphetic model directly.

The transitional strategy seems reasonable, as it might be a smoother process and is supposed to provide a unified legacy model for the future. But it is not feasible.

- For the standardization process, both the WG2 meetings and the Unicode Standard releases take place around once a year.
- Given the best estimate, the version 12.0 of the Unicode Standard (to be released in 2019) is the earliest opportunity for any change to become official. It will take a similar timespan to standardize either a new model or patches to the current model.
- WG2 and UTC will not have enough resources to simultaneously review the improvements of the current model and standardize the graphetic model, especially when the latter is expected to arrive as soon as possible.
- Although the timespan might seem to be long, it is not abundant for the industry to prepare for any changes.
- Vendors will not be cooperative on spending resources on migrating to a temporary solution when the real new model is already close.

Therefore, the direct strategy should be taken.

Since the graphetic model is designed to allow coexistence of the current model (see the next section), if needed, a de facto industrial standard can be formed among interested parties, but should not be expected from WG2 and UTC during the migration.

版，然后再迁移到字形模型。

- **直接：**直接迁移到字形模型。

过渡策略看起来有道理，因为它可能会是个更平滑的过程而且应该可以为未来提供统一的遗留模型。但它并不可行。

- 对于标准化程序，WG2 会议举行和 Unicode 标准发布都是大约每年一次。
- 在最理想的估计下，Unicode 标准的 12.0 版（将于 2019 年发布）是正式确认任何改变的最早机会。不论是标准化新的模型还是修补现行模型都会需要类似的时间跨度。
- WG2 和 UTC 不会有足够的资源来同时审核对现行模型的改善并标准化字形模型，尤其当期待字形模型尽早到来时。
- 尽管时间跨度可能看上去很长，但对业界来说，不论是为什么改变做准备这时间都并不充裕。
- 当真正的新模型已经临近时，厂商在花费资源迁移到临时方案这件事上不会配合。

因此，应当选择直接策略。

因为字形模型设计为允许与现行模型共存（见下一节），如有需要，事实上的业界标准可以在各方之间制定，但不应当在迁移过程中期待由 WG2 和 UTC 提供。

2 Coexistence 共存

In principle, the current model and the graphetic model can coexist during the migration.

- Although the possibility of coexistence seems to weaken the motivation for migration, it actually avoids the

原则上，现行模型和字形模型可以在迁移过程中共存。

- 尽管共存的可能性看上去会削弱迁移的动力，这其实避免了来回切换模型导

frustration from switching between models, thus leads to a smooth migration for users.

- In order to ensure stability of the encoding model, existing characters and variation sequences will not be modified.
- Only characters that can have the same shaping behavior in both the current model and the graphetic model will be reused in the graphetic model.
- A single font will be able to support both models. An existing font that supports a certain vendor implementation of the current model can be easily extended to also support the graphetic model. See section 5.

With a simple heuristics algorithm, the exact model used in a piece of text can be identified when needed, because all vowels and some frequently used consonants are encoded with characters distinct between the two models. Even on marginal cases when the heuristics fails to recognize the model, the rendered grapheme sequence is the same.

致的挫败感，因此为用户带来了平滑的迁移。

- 为了保证编码模型的稳定，不会修改现有的字符和变体序列。
- 只有能在现行模型和字形模型中有同样成形行为的字符才会在字形模型中复用。
- 单个字体会可以同时支持两个模型。支持现行方案特定厂商实现的现有字体可以轻易扩展至同时支持字形模型。参见第5节。

有需要时，用简单的启发式算法就可以识别一段文本具体使用的模型，因为所有的元音和一些频繁使用的辅音都是在两个模型间用不同的字符编码。即使在启发式算法无法识别模型的边缘情况下，渲染出来的字位序列也是一样的。

3 Collation 排序

A character-based simple collation should be as a minimal support level for general platforms.

However, considering the long time convention of collating according to underlying phonetic letters, a simple context-dependent algorithm can generate a decent result without natural language processing. The result is not truly based on underlying phonetic letters but is good enough for daily use. Shen Yilei (沈逸磊) is designing such an algorithm.

Note even for the current model a context-dependent algorithm is required for common styles of phonetic collation, because graphemes are often taken into consideration and the graphemic effect of FVSes is context-dependent. See GB/T 30851-2014 *Information technology — Traditional Mongolian sorting* for a collation standard.

See the next section if true phonetic collation is needed.

应当为普通平台指定基于字符的简单排序作为最小支持级别。

然而，考虑到长期以来按照底层语音字母排序的惯例，简单的上下文相关算法可以不用自然语言处理就生成像样的结果。这个结果并不真正基于底层语言字母，但对日常使用足够好。沈逸磊正在设计这样的算法。

注意，即使对于现行模型，常见的语音式排序风格也要求上下文相关的算法，因为经常会考虑字位而 FVS 的字位效用是上下文相关的。参见一个排序标准：GB/T 30851-2014 信息技术 传统蒙古文排序。

如果需要真正的语音排序，参见下节。

4 Underlying phonetic letters 底层语音字母

The current model stores underlying phonetic letters directly in characters, while the graphetic model will only store graphemes. When phonetic letters are needed, natural language processing or metadata tagging is employed to provide information.

- The current model assumes that phonetic letters and graphemes can be both encoded directly in text, with complex automatic and manual shaping rules.
- However in reality, because of the multiple possible character sequences for a single grapheme, users tend to freely assemble characters together to the desired grapheme sequence without caring about if the phonetic information is correct.
- Identification of phonetic letters is also controversial, because of different schools of orthographies and grammars.
- Thus text encoded with the current model is not reliable from the moment it is typed. Not to mention the incompatibility of various vendor implementations.
- In order to deal with unreliable phonetic information, when an accurate sequence of phonetic letters is needed (eg, for collation and text-to-speech applications), text encoded in the current model has to first undergo correction and normalization with natural language processing technologies.
- Note the correction and normalization process is often only based on rendered graphemes, while ignoring the phonetic values of characters. See a solution provided by one of the major vendors, IMUCS: 奥云蒙古文校正系统 (literally "Oyun Mongolian correction system") http://mc.mglip.com:8080.
- Therefore the natural language processing technologies needed for constructing phonetic letters from graphetic text is already available in the industry and should not be a burden.

现行模型将底层语音字母直接存储在字符中，而字形模型将会只存储字位。需要语音字母时，可用自然语言处理或元数据标记来提供信息。

- 现行模型设想语音字母和字位可以用复杂的自动及手动成形规则同时且直接编码在文本中。
- 然而在现实中，因为单个字位会有多种可能的字符序列，用户往往随意用字符拼凑出想要的字位序列而不管语音信息是否正确。
- 不同流派的正字法和语法对语音字母的识别也有争议。
- 因此用现行模型编码的文本从输入的那一刻起就不可靠。更不必提各厂商间互不兼容的实现。
- 为了应对不可靠的语音信息，在需要准确的语音字母序列时（比如排序以及文本至语音转换的应用），现行模型编码的文本必须首先用自然语言处理技术来校正并且正常化。
- 注意，校正与正常化处理经常只基于渲染出来的字位而忽略字符的语音值。参见几大厂商之一内大计算机学院提供的方案：奥云蒙古文校正系统 http://mc.mglip.com:8080。
- 所以，业界已经有从字形文本构建语音字母的自然语言处理技术了，这不会是负担。

5 Fonts 字体

Prototype fonts have been prepared for testing.

Existing fonts can be safely extended to support the graphetic model without breaking previously encoded text.

- The graphetic model only requires a limited small set of contextual rules. This allows a low cost for font development and will lead to a boost of diversity in the font market with new vendors coming in. See the revision of the introductory document for the graphetic model for an introduction.
- Existing glyphs in fonts will be directly reused, thus no design work is involved.
- Because the graphetic model encodes graphemes directly, trans-graphemic contextual rules in existing fonts are not required by the graphetic model. Only the sub-graphemic contextual rules are extended to process glyphs of graphetic characters.
- A well maintained existing font project should be able to support the graphetic model in a couple of hours. Vendors will not face high cost for upgrading fonts.
- Open source references for implementing the graphetic model in fonts will be provided by OpenType experts.

When the migration process starts, in order to phase out the current model, newly produced fonts will be recommended to only provide support for the graphetic model, although technically the legacy model can coexist in a font. Eventually, extended legacy fonts that supports both models should be replaced with graphetic-only new fonts as default fonts in major platforms such as operating systems.

用于测试的原型字体已有准备。

现有字体可以安全地扩展至支持字形模型，不会打破原先编码的文本。

- 字形模型只要求有限的一小组上下文规则。这让低成本的字体开发成为可能而且会因为新厂商的加入而带来字体市场多样性的增长。这方面的介绍参见字形模型介绍文档的修订版。
- 字体内现有的图形会直接复用，因此不涉及设计工作。
- 因为字形模型直接编码字位，它不需要现有字体中跨字位的上下文规则。只扩展亚字位的上下文规则来处理字形字符的图形。
- 良好维护的现有字体项目应当可以在几小时内支持字形模型。厂商在升级字体时不会面临高成本。
- 关于在字体中实现字形模型的开源参考资料会由 OpenType 专家提供。

当迁移过程开始，尽管现行模型在技术上可以共存于字体中，为了逐渐废除现行模型还是会建议新制作的字体只对字形模型提供支持。最终，对于操作系统这样主要平台的默认字体，扩展至支持两个模型的遗留字体应当被只支持字形模型的新字体取代。

6 Text rendering and operations 文本渲染与操作

Text engines will need to update or patch their Unicode Character Database and derived data to recognize newly encoded graphetic characters, especially for the script property and joining type.

- No other changes are required as long as a text engine

为了识别新编码的字形字符，尤其为了字符的文字属性和连写类型信息，文本引擎会需要更新它们的 Unicode 字符数据库和衍生数据，或者打补丁。

- 只要文本引擎已经支持现行模型就不

already supports the current model.

- The three major text engines today, DirectWrite/ Uniscribe (Windows, Office, Edge, Internet Explorer, etc), HarfBuzz (Android, Chrome, Firefox, etc), and Core Text (iOS, macOS, etc), are all maintained by experts that actively collaborate with the Unicode Consortium.

Software libraries that provide support for higher-level text editing operations (such as word boundary detection, word selection and count, hyphenation, and justification) will also need to be updated accordingly.

- Such operations are not yet well supported for the current model in major platforms.
- Although often claimed by experts, average users and the publishing industry do not actually have a significant preference on the special treatments (narrower than normal word space, forbidding line break, extending word boundary, etc) of the whitespace (encoded in the current model as NNBSP U+202F NNBSP with special contextual shaping effect) preceding an enclitic.
- The current model imposes difficulty for hyphenation and justification, because its long-distance shaping effect of vowel harmony is easily broken by a line break or *nirugu* inside a word but expected behavior and solutions are underspecified.

需要其他的更改。

- 如今的三大文本引擎，DirectWrite/ Uniscribe（Windows、Office、Edge、Internet Explorer 等）、HarfBuzz（安卓、Chrome、Firefox 等）、Core Text（iOS、macOS 等）都是由与 Unicode 联盟积极协作的专家维护的。

为高层文本编辑操作（比如词的边界检测、词的选中与计数、在词中选择合理位置断行、文本左右双齐）提供支持的软件库也将需要相应的升级。

- 这些操作还未在主要平台上对现行模型有良好支持。
- 尽管专家经常如此声称，普通用户和出版业并不显著偏好对附加成分前的空白（在现行模型中编码为有特殊上下文成形效用的 NNBSP U+202F NNBSP）进行特殊处理（窄于普通词间空格、禁止断行、拓展词界等）。
- 现行模型对在词中选择合理位置断行以及文本左右双齐的需求造成困难，因为现行模型里元音和谐的的长距成形效用很容易被词内的断行和 *nirugu* 打破，但对预期的行为和解决方案缺乏详细规范。

7 Input 输入

Prototype keyboards have been prepared for testing.

A straightforward keyboard design for the graphetic model will be a simple one-to-one, key-to-character mapping. See figure 1.

- In this case, because the layout deviates from the conventional concept of phonetic letters, keycaps should be well designed to emphasize differences between graphemes.
- Eg, the graphetic character [ɟ] a non-joining should be explicitly distinguished from a joined toothless left

用于测试的原型键盘已有准备。

直截了当的字形模型键盘设计会是简单的一对一、键位对字符映射，如图1。

- 这样的话，因为布局偏离惯例的语音字符概念，键帽应当良好设计以强调字位之间的区别。
- 例如，字形字符 [ɟ] a non-joining（不连写的 *a*）应当明确区别于连写的无牙

tail (a sub-graphemic structure after round consonants, eg, in 𑀩 *ba*; or a grapheme in Hudum Ali Gali text), and 𑀩𑁆 `u tailed` should be distinguished from 𑀩 `U+182A BA`.

- Using `medi` forms by default on keycaps can be a good strategy, because it provides consistency for most characters and allows special characters that do not commonly have `medi` forms to stand out.
- However, such a simply keyboard layout might only become popular among professional typists, similar to the case of the Wubi (五笔) input method for Chinese.

左尾（这是圆头辅音后的亚字位结构，比如在 𑀩 *ba* 中；或者是胡都木阿礼嘎礼文本中的字位），而 𑀩𑁆 `u tailed`（有尾 *u*）应当区别于 𑀩 `U+182A BA`。

- 在键帽上默认用 `medi` 形式（中形）会是个好策略，因为这给大多数字符带来一致性又让通常没有 `medi` 形式的特殊字符得以凸显。
- 然而，这样的简单键盘布局可能只会在专业打字员中流行，类似于中文五笔输入法的情况。

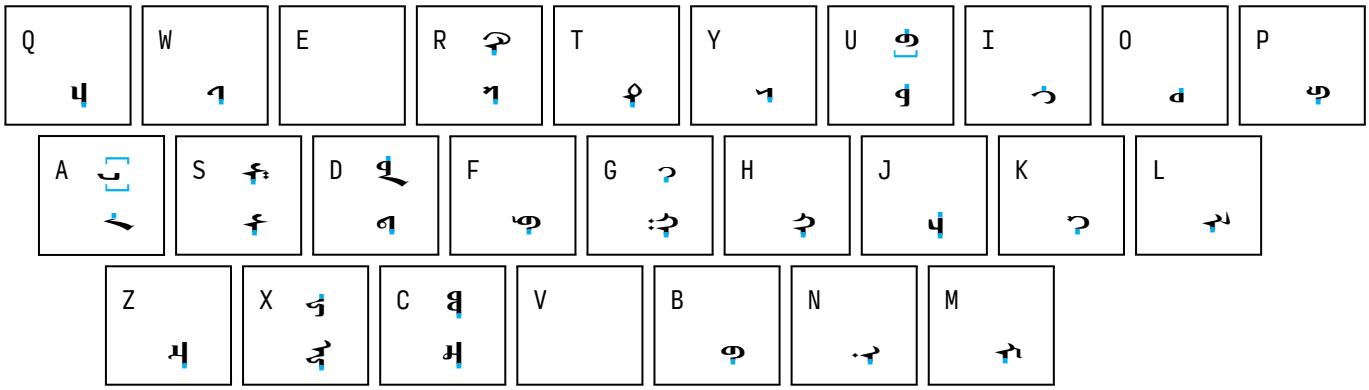

Figure 图 1: Prototype layout A 原型布局 A

To meet the need of average users, phonetic letters can be utilized to various levels in a keyboard layout or an input method design. “Smart” or “whole-word” input methods that provide word suggestions will be especially suitable for average users, and are already popular for the current model because for average users it is too complicated to directly manipulate.

- Smart input methods can keep users from worrying about the exact character sequences.
- Such solutions with word suggestions have already been popularized by the major vendors (Menksoft, Delehi, IMUCS, etc) in the user community, in order to make the current model usable. See appendix A on page 10 for links to existing input solutions.
- Keyboard layouts can also map multiple keys or key combinations to a single graphetic character or se-

为满足普通用户的需要，键盘布局或输入法的设计中可以对语音字母有各种级别的利用。提供侯选词的“智能”或“整词”输入法会尤其适合普通用户。而且因为直接操控现行模型操控太复杂，此类输入法已经很流行了。

- 智能输入法能避免用户为具体的字符序列而困扰。
- 为使现行模型可用，此类有候选词的解决方案已经在几大厂商（蒙科立、德力海、内大计算机学院等）的推动下在用户社群中流行起来。参见第10页附录A中现有输入方案的链接。
- 键盘布局也可以映射多个键位或键位组合至一个字形字符或序列，以提供与

quence, to provide an experience more aligned to the understanding of phonetic letters, while still allows accurate control on output like a simple one-to-one mapping keyboard layout. See figure 2.

语音字母理解更加一致的体验，同时还像简单的一对一映射键盘布局一样允许准确的控制。如图2。

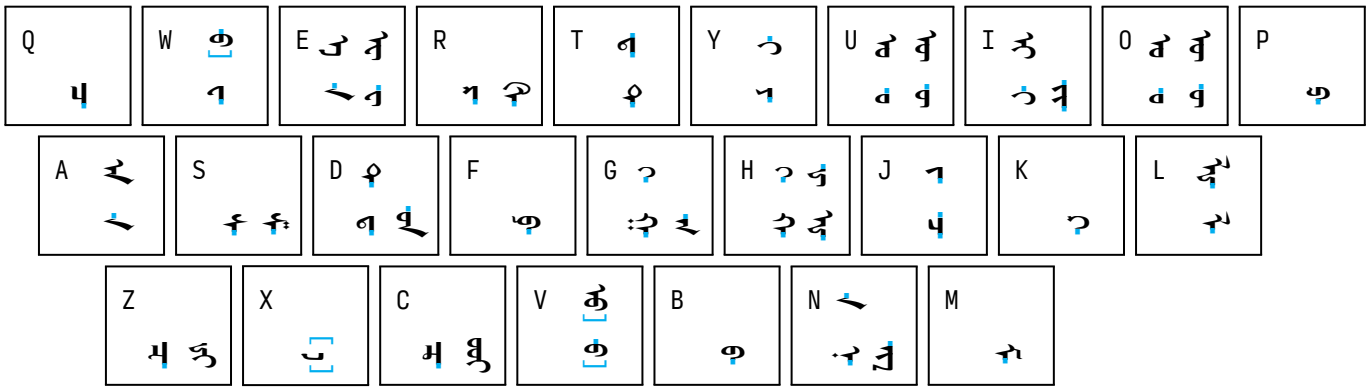

Figure 图 2: Prototype layout D 原型布局 D

- Keyboard technologies that allows contextual operations (from simple dead keys to advanced solutions like Keyman: https://keyman.com) can allow a single key to contextually output different characters, eg, coda forms of consonants.
- Note the more automatic mechanisms a keyboard solution provides, the more it suffers from the complication of contextual rules in the current model. Although with the graphetic model users can easily spot unexpected characters and correct in time, which ensures unambiguous text representation.
- Theoretically a keyboard-like input method can even internally provide the full set of contextual rules of the current model, and works like a virtual machine. So it allows users accustomed to the current model to input in the old way, ie, keys of FVSes and other format characters are manually controlled, but produced characters are graphetic. Such a transcoding is obviously limited to a specific legacy implementation. See also the next section for a discussion on converting data.

- 允许上下文操作的键盘技术（从简单的死键到 Keyman 这样的高级方案：https://keyman.com）可以允许单个键位根据上下文输出不同的字符，比如辅音的音节尾形式。
- 注意，键盘方案提供越多的自动机制就越会遭受现行方案中复杂上下文规则的问题。不过用字形模型时用户能轻易发现意外的字符并及时校正，保证了文本的无歧义表记。
- 理论上，一个像键盘一样的输入法甚至可以在内部提供现行模型的全套上下文规则，然后像虚拟机一样工作。于是习惯于现行模型的用户可以用老方式输入，即手动控制 FVS 等格式字符的键位但产生字形字符。这样的转码显然会局限于特定的遗留实现。另见下一节对数据转换的讨论。

8 Data conversion 数据转换

The graphetic model directly encodes graphemes, while the current model encodes underlying phonetic letters

字符模型直接编码字位，而现行模型编码底层语音字母并依赖上下文规则选择

and rely on contextual rules to select correct graphemes in order to output a grapheme sequence. Therefore, in order to convert data from the current model to the graphetic model, theoretically the algorithm has to implement the full set of contextual rules of the current model.

- Considering the numerous incompatible, underspecified implementations for the current model, the workload can be huge.
- But actually, internally what existing fonts already do is exactly converting phonetic character sequences to grapheme sequences.
- Therefore we can simply map the output glyphs of existing fonts to graphetic characters. In this way it is ensured that all internal rules in fonts are captured. Text engines work as backends for the conversion algorithm to get the glyph sequences.

The information of underlying phonetic letter, if accurate, can be kept during the conversion and stored as metadata.

正确字位以输出字位序列。所以，为了把数据从现行模型转换到字形模型，理论上算法须要实现现行模型的全套上下文规则。

- 考虑到现行模型不兼容且缺乏详细规范的繁多实现，工作量会是巨大的。
- 但其实，现有字体在内部已经在做的事情恰恰就是把语音字符序列转换成字位序列。
- 所以我们可以简单地把现有字体输出的图形映射到字形字符。这样就可以确保捕获字体的所有内部规则。文本引擎用作转换算法的后端来获得图形序列，

如果底层语音字母的信息准确，可以在转换时保留并存储为元数据。

9 Use cases beyond the modern Hudum 现代胡都木文之外的用例

The historical and stylistic forms in L2/16-309 *Proposed additions for Mongolian in 5th edition of UCS* are discussed in the revision of the introductory document for the graphetic model. Basically, the majority of them are either simple duplications of existing graphemes (should be encoded directly with the corresponding graphetic characters) or stylistic variants of existing graphemes (should be handled by fonts). Only a few are candidates for additional graphetic characters.

Solutions for Hudum Ali Gali letters as well as other writing systems that use the Mongolian script (Todo, Manchu, Sibe, etc) are discussed in Zheng Weizhe (郑维喆)'s document *A mixed encoding scheme for the Mongolian block*.

L2/16-309《Proposed additions for Mongolian in 5th edition of UCS》中的历史和风格形式在字形模型介绍文档的修订版中讨论。基本上，多数都是现有字位的简单重复（应当直接编码为相应的字形字符）或者现有字位的风格变体（应当由字体处理）。只有少数一些是新增字形字符的候选。

对胡都木阿礼嘎礼以及其他使用蒙古文的书写系统（托忒文、满文、锡伯文等）的解决方案在郑维喆的文档《A mixed encoding scheme for the Mongolian block》中讨论。

Appendix 附录

A Major vendors 主要厂商

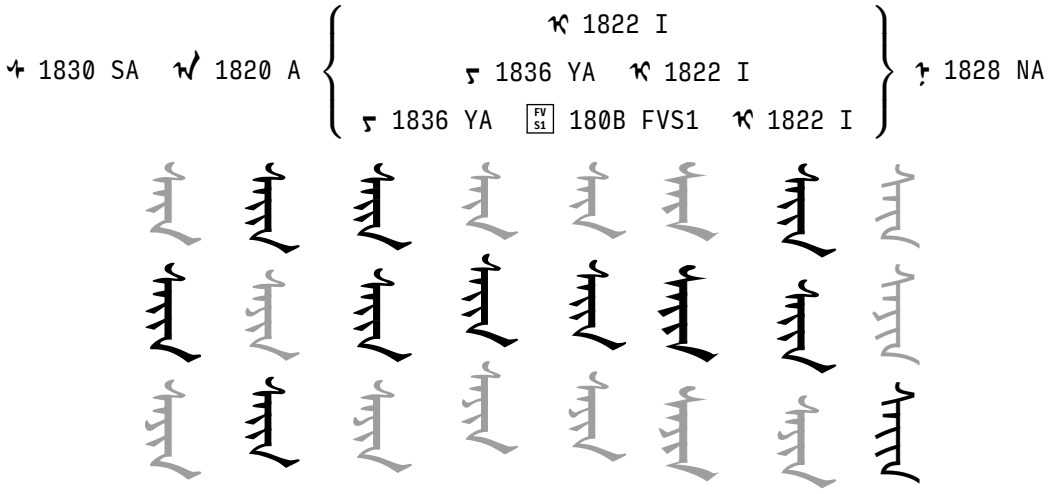

Figure 图 3: Incompatible vendor implementations. 互不兼容的厂商实现。

There are eight major vendors (listed below) offering Unicode–OpenType solutions. Legacy non-Unicode–OpenType solutions still exist.

These vendor solutions are already in wide use but their implementations are all incompatible to each other. See the figure 3 above for how incompatible the vendor implementations are — even for the basic word ᠰᠠᡳᠨ (*sain*: good) there can be at least three ways of encoding and not a single one is supported by all vendors (wrong shapes are colored gray).

1. **Menksoft** (蒙科立). Fonts: Menk Qagan Tig 1.02, etc, available on http://font.menksoft.com. Input: http://ime.menksoft.com/.

2. **Delehi** (德力海), also known as Almas. Fonts: Mongolian White 1.1, etc, available on http://delehi.com/cn and the original Almas site http://mongolfont.com/en. Input: http://www.delehi.com/cn/419.html and http://www.delehi.com/cn/1423.html.

3. **IMUCS** (College of Computer Science, Inner Mongolia University; 内蒙古大学计算机学院), products of

有八个主要厂商（下列）提供 Unicode–OpenType 解决方案。遗留的非 Unicode–OpenType 方案仍然存在。

这些厂商方案已在广泛使用中，但他们的技术实现全都互不兼容。由上方图3可见厂商实现间有多么不兼容——就连基本的单词 ᠰᠠᡳᠨ（*sain*：好）都有至少三种编码方式而且没有一种受所有厂商支持（错误的形状以灰色标出）。

1. **蒙科立**。字体：Menk Qagan Tig 1.02 等，可从 http://font.menksoft.com 获得。输入：http://ime.menksoft.com/。

2. **德力海**，又称 Almas。字体：Mongolian White 1.1 等，可从 http://delehi.com/cn 以及原 Almas 网站 http://mongolfont.com/en 获得。输入：http://www.delehi.com/cn/419.html 及 http://www.delehi.com/cn/1423.html。

3. **内大计算机学院**（内蒙古大学计算机学院），产品以“奥云”品牌发布。字

which are released under the *Oyun* brand. Fonts: Oyun Qagan Tig 2.05, etc, available on http://oyun.mglip.com/mongolfont/index.aspx. Input: http://oyun.mglip.com.

4. **Founder** (方正 *Fāngzhèng*), also the biggest type foundry of Chinese fonts in China: http://foundertype.com; FZMWBTOT_Unicode.TTF.

5. **Huaguang** (华光 *Huáguāng*), a vendor similar to but smaller than Founder: http://hgfonts.com; HGMWXB_NMBS.TTF.

6. **Bolorsoft** (Болорсофт): http://bolorsoft.com; MongolianScript, an open source (SIL Open Font License) font, available on http://font.bolorsoft.com (website currently down).

7. **Microsoft**: Mongolian Baiti, Windows built-in. An outdated introduction page is available (the latest version is 5.52, as of Windows 10 Creators Update): https://microsoft.com/typography/fonts/family.aspx?FID=325.

8. **Google**: Noto Sans Mongolian, a low-contrast design, open source, Android built-in, also available on: https://google.com/get/noto/#sans-mong.

Menksoft, Delehi, and IMUCS solutions are commonly used by average users in China because their fonts and input solutions are released to the public for free. Founder and Huaguang solutions dominate the Mongolian publishing industry in China. Bolorsoft is a major vendor in Mongolia although not well known in China. Microsoft and Google have provided fonts pre-installed in their widely used operating systems, Windows and Android.

An on-going initiative led by the IMEAC (Ethnic Affairs Commission, Inner Mongolia Autonomous Region; 内蒙古自治区民族事务委员会) is trying to improve the current model and unify vendor implementations. This initiative has gained support from Menksoft, Delehi, IMUCS, and Microsoft; Huaguang is likely to join.

体：Oyun Qagan Tig 2.05 等，可从 http://oyun.mglip.com/mongolfont/index.aspx 获得。输入：http://oyun.mglip.com。

4. **方正**，也是中国最大的中文字体厂商：http://foundertype.com；FZMWBTOT_Unicode.TTF。

5. **华光**，类似方正但较小的一个厂商：http://hgfonts.com；HGMWXB_NMBS.TTF。

6. **Bolorsoft**（Болорсофт）：http://bolorsoft.com；MongolianScript，一款开源（协议为 SIL Open Font License）字体，可从 http://font.bolorsoft.com 获得（网站目前下线）。

7. **微软**（Microsoft）：Mongolian Baiti，Windows 自带。有个过期的介绍页面（截至 Windows 10 创意者更新，最新版是 5.52）：https://microsoft.com/typography/fonts/family.aspx?FID=325。

8. **谷歌**（Google）：Noto Sans Mongolian，低对比设计，开源，安卓自带，也可从 https://google.com/get/noto/#sans-mong 获得。

中国的普通用户普遍使用蒙科立、德力海、内大计算机学院的解决方案，因为他们的字体和输入方案向公众免费发布。方正和华光的方案在中国统治了蒙文出版业。Bolorsoft 在蒙古国是主要厂商，尽管在中国不知名。微软和谷歌在他们受到广泛使用的操作系统 Windows 和安卓中提供了预装的字体。

民委项目（由内蒙古自治区民族事务委员会主持的一个进行中的项目）正在努力改善现行模型并统一厂商实现。此项目已获得蒙科立、德力海、内大计算机学院、微软的支持，华光有可能加入。