最小化修订方案的实施情况介绍
——为最短时间内实现蒙古文编码的相对稳定和统一

An introduction to the execution of our Solution with Minimal Modifications
——for achieving a stable and unified Mongolian encoding as soon as possible

梁金宝（Liang Jinbao）
2019.4.3
目录 | MENU

01 概述
Outline

02 分析与分类存在问题
Problem Classification

03 我们做了哪些工作
What have we done?

04 对未来工作的建议
Advice for future work
1. 客观事实
现在的蒙古文编码确实存在一些问题，并且在部分应用场景下从制定最初开始，持续影响着蒙古文在信息技术领域的正确使用。这是客观事实。为了解决这些问题，陆续有不同个人和不同机构在尝试各种改善方案。如纯字形方案、虚拟图形字母方案、改善的语言字母方案、语言音节方案等等。

1. Objective Fact
The current Mongolian code does have some problems, and in some application scenarios, from the beginning of the development, it continues to affect the correct use of Mongolian in the field of information technology. This is an objective fact. In order to solve these problems, different individuals and different institutions are trying to improve various solutions. Such as a pure glyph scheme, a virtual graphic letter scheme, an improved language letter scheme, a language syllable scheme, and the like.
2. My Advice

In my opinion, the most urgent thing at present is that we have to improve the existing schemes with minimal modifications in the shortest possible time, so that Mongolian will have a stable and unified coding scheme in the shortest time. Let users enjoy stable use of the language, then we could further the idealized scheme with more research, demonstration, and test. At that point we wouldn't be so hurried to set up a standard. We have many units and individuals participated in the re-engineering of standards and implementation standards. It is difficult to resolve the disagreements between them and build one standard in three years. However, under the current circumstances of Mongolian coding situation, even a year will also make us felt too long. We have to build a stable and singel standard in the shortestetet amount of time.
3. Problem classification

After analyzing the problems that have been discovered so far, it is easy to know that the problems existing in coding now are roughly classified into the following three types of problems:

1. Some software systems are not fully supporting the complex text rendering engines.
2. The confusible issue and the internet security issue caused by it.
3. Incompatible coding between different manufacturers’ software products.
1. Some systems are not fully supported for complex text rendering engines

Some systems are not fully supported for complex text rendering engines. For example, some legacy software such as Windows XP, Photoshop, embedded system control LED display, old browser and so on. We should follow the trend of the technology development. There are always other ways to solve the problem of historical stock software. Just like people are already doing it now. We should not be disturbed by these factors when we improve our standards. We are not trying to solve these kind of problems.
2. 同形异码问题及由此导致的网络安全问题
The confusable issue and the internet security issue caused by it

同形异码问题是当前编码方案本身固有的问题，我们无法通过对当前方案进行修订来解决。我们能做的只有在【忠于语言文字字母结构的编码方案带来的便利性】和【它附加带来的同形异码复杂性】之间的平衡考虑。

The homomorphic problem is an inherent problem in the current coding scheme, and we cannot solve it by modifying the current scheme. All we can do now is keep the balance between ‘the convenience brought by the coding scheme loyal to the linguistic letter structure’ and the ‘complexity of the isomorphic complexity it brings’.
Incompatible coding between different manufacturers’ software products

As long as the average user says “the coding standard is not uniform” and “the current coding scheme is flawed”, it is basically plagued by this type of problem. The Unicode standard content does not contain the normative content to solve this problem. It is the letter deformation rule of the language text system, which is referred to as the conversion rule. As long as we have a unified set of conversion rules to follow together, then these problems will be solved. The main content of our minimization of revisions is to standardize and unify it.
**The Solution with Minimal Modifications**

**What revisions have we made?**

According to the report of Mongolian Working Group Meeting 2 (April 7, 2018, USA), we have carried out more targeted ten items on the third type of problem (Incompatible coding between different manufacturers' software products). It will be introduced on the next page. There is an agreement that the phonetic model should not be abandoned.

**The key point of the report for MWG2**

1. There is an agreement that the phonetic model should not be abandoned.
2. All attendees want to have a stable Mongolian encoding.
3. For NNBSP, further clarifications are needed and perhaps there are potential changes.
4. We need one set of OpenType rules for rendering Mongolian text based on the current model. Having competing font standards is detrimental to the user community.
1. Solve problems caused by NNBSP with MSC model

**Status:** When using NNBSP in Mongolian, there will always be cases where the extra component is displayed incorrectly in all or part of the version on iOS, macOS, Android, Office, Browser. This phenomenon has not been solved for a long time. There is only one way to solve this problem quickly and completely: abandon the use of NNBSP.

**Solution:** Combine MVS&NNBSP into MSC

There are two separators used in Mongolian: MVS and NNBSP. The function is 1. continuous word, 2. split word vowel A/E, 3. split additional component. We renamed MVS as MSC and used MSC wherever there is MVS and NNBSP requirements, which is a complete replacement for these two control symbols.

**Notes:** We still need to use NNBSP?
Solve problems caused by NNBSP with MSC model

 MSC, SP

 MSC, MSC, SP

 MSC

 SP

 MSC, Mongolian Suffix Connector, U+180E

 SP, SPACE, U+002A
2. Using NIRUGU to Solve Problems Caused by ZWJ

Solve problems caused by ZWJ with NIRUGU

现状: 在蒙古文中强制生成位置上下文环境时，也就是显示孤立字形（上中下）时，需要使用ZWJ(U+200D)。但是在很多平台上使用的时候，跟NNBSP (U+202F) 一样总会出现不稳定的错误现象。并且从编码制定最基础开始一直没有彻底解决。

Status: ZWJ (U+200D) is required when forcing the location context in Mongolian, that is, when displaying isolated glyphs (initial, middle, final). However, when used on many platforms, there is always an unstable error like NNBSP (U+202F). And it has not been completely solved since the beginning of the code-design.

解决方法 | Solution:
- 用NIRUGU代替ZWJ
- Replace ZWJ with NIRUGU

Win10, Word2010
Clearly define the UCD-Attributes of NIRUGU and ZWJ

现状：蒙古文字母是依据上下文环境来自动变形的。所以我们在词中间插入并不具备语法属性的NIRUGU和ZWJ的时候，我们需要明确定义它的透明属性。即，是上下方向上传递语法属性，还是阻断语法属性。到目前为止并没有明确定义，所以不同厂家的做法是不同的。这也编码不兼容的原因之一。

Status: Mongolian letters are automatically deformed according to the context. So when we insert NIRUGU and ZWJ with no grammatical attributes in the middle of the word, we need to explicitly define its transparent attributes. That is, whether the syntax attribute is passed in the up and down direction or the syntax attribute is blocked. So far there is no clear definition, so the practices of different manufacturers are different. This is also one of the reasons for coding incompatibility.

解決方法 | Solution:

- 建议NIRUGU上下传递语法属性。ZWJ阻断语法属性
- I recommend that NIRUGU pass syntax attributes up and down. ZWJ blocks syntax attributes.
4. 统一连写位置模型和语言学词里位置模型之间的定义分歧

Unify the differences for cursive-joining model and linguistic-joining model

The cursive-joining model and the linguistic-joining model do not actually conflict. The glyph identified as a final form on the cursive-joining model can be identified as a middle form on the linguistic model. This is only different in the way of identification. In our Solution with Minimal Modifications, after the adjustment of the fvs allocation scheme, there is no disagreement between the two models. This is one of the important differences of opinion on the MWG1 & MGW2.
The cursive-joining model is biased towards technical implementation, and the linguistic-joining model is biased toward linguistic theory. In order to make the coding scheme more rigorous and systematic, we have slightly adjusted the positional property of first letter for mongolian suffix. However, we should be clear that no matter how technically adjust the positional attributes, he will not affect how to interpret and identify from the linguistics. Even on a technical level, you classify a glyph as a final form, but in linguistics we can call it a middle form. This is not at all conflicting.
Unicode Code Chart is a very simple form that can carry less information. In order to simplify the correspondence between FVS and letter variants, we should adopt a variant to fix the corresponding FVS. Instead of a variant, use this FVS in this context and use that FVS in another context. This is the solution to the problem of Inconsistent-Use-of-FVSs.
In the current coding model, invisible FVS is a very troublesome thing. But we can greatly improve the user experience through technical means. Although it is a small optimization, it is very obvious to improve the user experience, so it is listed separately here. The solution is that when the user uses FVS in a wrong way, or the user uses an extra FVS, the font shows him a hinting error graph.
In fact, the lack of a unified conversion rule is the major cost of the current “unification of coding”. The development of a stable and unified transformation rule based on the current coding scheme is the key point of our hard work this year.
No matter what type of information product, it is necessary to do large-scale of systematic testing. We also collected a set of word tests that cover all the deformation rules. And the deformation test compound detection is performed on each manufacturer’s word.
There are two types of punctuation marks used in Mongolian typesetting. One is the Mongolian text, such as ‘‘’, like: ‘，《》. The other is borrowed from other language coding areas, such as ‘‘,’’. However, there is currently no detailed specification for the use of punctuation. Such as visual design requirements, usage requirements, etc. We know that these may not be within the scope of the coding standard, but it is important for Mongolian use. So we tentatively made the punctuation recommendation specification document.

1. Vertical alignment feature
2. Proportional feature of blank area inside glyph
3. How to use the text (pre-space, after-space)
开发&测试&改进&实施
Development&Testing&Improvement&shishi

自从上次第二次蒙古文编码工作会议之后，我们并没有停留在方案研究等基础性工作上。在一年的时间内，我们联合各个字体厂商和相关研究机构通力合作，进行了最小化修订方案的可行性论证、产品开发、产品测试、产品实施等各个环节的工作。目前制作了相互完全兼容的一百多套字体，并在三十多家企事业单位的五十多个工程项目中推荐使用。其中包括制作七千多种蒙古文电子图书的工程。我们的字体将会很快像社会无偿提供。

Since the last MWG2, we have not stayed in basic work such as program research. In a year's time, we worked together with various font manufacturers and related research institutions to carry out the work of the-Solution-with-Minimal-Modifications of demonstration, development, testing, implementation and other aspects. At present, more than 100 sets of fonts that are fully compatible with each other have been produced and recommended in more than 50 engineering projects of more than 30 enterprises and institutions. This includes the production of more than 7,000 Mongolian e-books. Our fonts will soon be available free of charge to society.
<table>
<thead>
<tr>
<th>Font Name</th>
<th>Sample Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>传统蒙古文白正体</td>
<td><img src="image1" alt="Sample Image" /></td>
</tr>
<tr>
<td>传统蒙古文黑正体</td>
<td><img src="image2" alt="Sample Image" /></td>
</tr>
<tr>
<td>传统蒙古文新正体</td>
<td><img src="image3" alt="Sample Image" /></td>
</tr>
<tr>
<td>传统蒙古文黑体</td>
<td><img src="image4" alt="Sample Image" /></td>
</tr>
<tr>
<td>传统蒙古文等体</td>
<td><img src="image5" alt="Sample Image" /></td>
</tr>
<tr>
<td>传统蒙古文手楷体</td>
<td><img src="image6" alt="Sample Image" /></td>
</tr>
<tr>
<td>传统蒙古文行书体</td>
<td><img src="image7" alt="Sample Image" /></td>
</tr>
<tr>
<td>传统蒙古文横写体</td>
<td><img src="image8" alt="Sample Image" /></td>
</tr>
<tr>
<td>传统蒙古文手写体</td>
<td><img src="image9" alt="Sample Image" /></td>
</tr>
<tr>
<td>传统蒙古文行楷体</td>
<td><img src="image10" alt="Sample Image" /></td>
</tr>
<tr>
<td>传统蒙古文楷体</td>
<td><img src="image11" alt="Sample Image" /></td>
</tr>
<tr>
<td>传统蒙古文ZS</td>
<td><img src="image12" alt="Sample Image" /></td>
</tr>
</tbody>
</table>

德力海公司字体 (27)
The Fonts for Delehi Company (27)
### The Fonts for Menksoft Company (28)

<table>
<thead>
<tr>
<th>Font Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional Mongolian White Font</td>
<td>Includes traditional Mongolian, Tungus, Khitan, and Mongolian script fonts.</td>
</tr>
<tr>
<td>Traditional Mongolian Regular Font</td>
<td>Traditional Mongolian regular font.</td>
</tr>
<tr>
<td>Traditional Mongolian Black Font</td>
<td>Traditional Mongolian black font.</td>
</tr>
<tr>
<td>Traditional Mongolian Bold Font</td>
<td>Traditional Mongolian bold font.</td>
</tr>
<tr>
<td>Traditional Mongolian Title Font</td>
<td>Traditional Mongolian title font.</td>
</tr>
<tr>
<td>Traditional Mongolian Script Font</td>
<td>Traditional Mongolian script font.</td>
</tr>
<tr>
<td>Traditional Mongolian Regular Script Font</td>
<td>Traditional Mongolian regular script font.</td>
</tr>
<tr>
<td>Traditional Mongolian Bold Script Font</td>
<td>Traditional Mongolian bold script font.</td>
</tr>
<tr>
<td>Traditional Mongolian Title Script Font</td>
<td>Traditional Mongolian title script font.</td>
</tr>
<tr>
<td>Traditional Mongolian Regular Block Font</td>
<td>Traditional Mongolian regular block font.</td>
</tr>
<tr>
<td>Traditional Mongolian Bold Block Font</td>
<td>Traditional Mongolian bold block font.</td>
</tr>
<tr>
<td>Traditional Mongolian Title Block Font</td>
<td>Traditional Mongolian title block font.</td>
</tr>
<tr>
<td>Traditional Mongolian Regular Cloud Font</td>
<td>Traditional Mongolian regular cloud font.</td>
</tr>
<tr>
<td>Traditional Mongolian Bold Cloud Font</td>
<td>Traditional Mongolian bold cloud font.</td>
</tr>
<tr>
<td>Traditional Mongolian Title Cloud Font</td>
<td>Traditional Mongolian title cloud font.</td>
</tr>
<tr>
<td>Traditional Mongolian Regular Wave Font</td>
<td>Traditional Mongolian regular wave font.</td>
</tr>
<tr>
<td>Traditional Mongolian Bold Wave Font</td>
<td>Traditional Mongolian bold wave font.</td>
</tr>
<tr>
<td>Traditional Mongolian Title Wave Font</td>
<td>Traditional Mongolian title wave font.</td>
</tr>
<tr>
<td>Traditional Mongolian Regular Edge Font</td>
<td>Traditional Mongolian regular edge font.</td>
</tr>
<tr>
<td>Traditional Mongolian Bold Edge Font</td>
<td>Traditional Mongolian bold edge font.</td>
</tr>
<tr>
<td>Traditional Mongolian Title Edge Font</td>
<td>Traditional Mongolian title edge font.</td>
</tr>
<tr>
<td>Traditional Mongolian Regular Frame Font</td>
<td>Traditional Mongolian regular frame font.</td>
</tr>
<tr>
<td>Traditional Mongolian Bold Frame Font</td>
<td>Traditional Mongolian bold frame font.</td>
</tr>
<tr>
<td>Traditional Mongolian Title Frame Font</td>
<td>Traditional Mongolian title frame font.</td>
</tr>
</tbody>
</table>

*Note: The above font styles are examples and may not correspond to actual font styles available for Menksoft Company.*
The Fonts for Galtu Company (50)
The Fonts for Galtu Company (50)
Fonts are the way to implement coding schemes, and it is a time-consuming and laborious task to create a set of fonts that are aesthetically pleasing and conform to the OpenType standard. Art and technology can’t be lacking. If the standard is uniform, the font is also generic. There is no need for everyone, every company, every institution to make their own fonts. The hundreds of Mongolian scripts we produce are all freely released to the world, and can be used free of charge as long as there is a need for use.
已实现的 & 还未实现的
Implemented & Not Yet Implemented

我们已做到的 | What We Have Done
1. 不同厂商产品之间的编码兼容
   Code compatibility between different vendors’ products
2. 各大操作系统平台上不乱码（字体升级到本方案后）
   No garbled on major operating system platforms (after font upgrade)

※我们认为在基于现在编码模型，这是我们以最小的代价能够做到的最好的效果。
We think that based on the current coding model, this is the best we can do at the least cost.

我们没能做到的 | What we can’t implement:
1. 让XP和Photoshop等软件直接支持unicode蒙古文
   Let XP and Photoshop directly support unicode Mongolian
2. 让嵌入系统控制的LED显示屏直接显示unicode蒙古文
   Let the embedded system control LED display directly display unicode Mongolian.
3. 在现在的编码模型上消除同形异码现象
   Eliminate the confusible issue on current coding models

What about on iOS | Android | Windows | Linux ?
下一步计划的意见
The Advice For Next Step

事情比较复杂，参与的人多，分歧也比较大。我建议分阶段来实施。这样我们容易取得阶段性成果。我建议分三步进行。这样我们步步为营，大家一起努力一步一步地往前推进此项工作。

Things are more complicated, there are more people involved, and the differences are larger than in the past. I recommend that implement our assignment in several stages. It will more easier for us to achieve a phased result. I recommend that put our task in three steps. In this way, we can work step by step, and everyone can works together to advance our task.
The First Step: Focus on the code charts

In the first step, we should focus on reaching a consensus on the Unicode Code Chart. That is, when we answer the following three questions from others based on this form, our answers should be consistent:

1. What are the Mongolian letters? What is its code point?
2. What variants do each Mongolian letter have in the position of cursive-joining model?
3. When a variant cannot be properly displayed by context-driven, which FVS should we use to force the glyph to be requested? (fvs static allocation & fvs dynamic allocation)
In the second step, we agreed on the transferring rules (shaping rules) level and formed a unified Unicode Technical Note. This issue was also mentioned in the last MWG2 Report, but we are busy with the development of technical notes, testing a lot of work, failed to complete this task. But now that we have completed the preliminary work, we have the conditions to start production. I hope that many parties will work together to complete this work.

The point of the report for MWG2（April 7, 2018, USA）
- It was decided that creating a Unicode Technical Note that would be a visual guide to the behavior of the Mongolian script would be very beneficial. Liang Jinbao and Kamal Mansour agreed to cooperate to create such a document on Mongolian script behavior.
完成了前两步工作，我们就有了充足的时间去研究更多的、更理想化的编码方案。我们在追求理想方案的道路上，如果只看远方不看眼前，那么我们有可能总会原地踏步……

After completing the first two steps, we have plenty of time to study more and more idealized coding schemes. On the road to pursuing the ideal solution, if we only focus on the distance and do not pay attention to the front, then we may always staying in the same place..
非常感谢 | THANKS