

## On the Kodo incense patterns

Eduardo Marín Silva

01/07/2022

**Introduction.** This is a response to the proposal of encoding Kodo incense patterns ([L2/22-117](#)); wherein I recommend to encode the most relevant characters atomically, rather than using the complicated system of character sequences.

**The proposed encoding.** The rationale behind the model is one of codepoint space. Given the entirety of the possible patterns (up to ten packets), it would take more than 3 Unicode planes to encode them all. However, this assumes that all possible patterns are attested and therefore needed to represent.

Indeed, according to the author, all attested patterns with more than five packets appear in a single source (NN4: 源氏香の具 / 中野幸一旧蔵); it is clear that very few users could make use of those symbols, as compared with the others, that are attested to this day (the author does not even propose those as Named Character Sequences due to their obscurity). In addition, out of the thousands of possible patterns only less than fifty are attested. So even if every attested pattern was encoded separately, it would only take 126 codepoints or a block of eight rows. In contrast, the “Yijing Hexagram Symbols” occupies four rows and the “Tai Xuan Jing Symbols” occupies six rows.

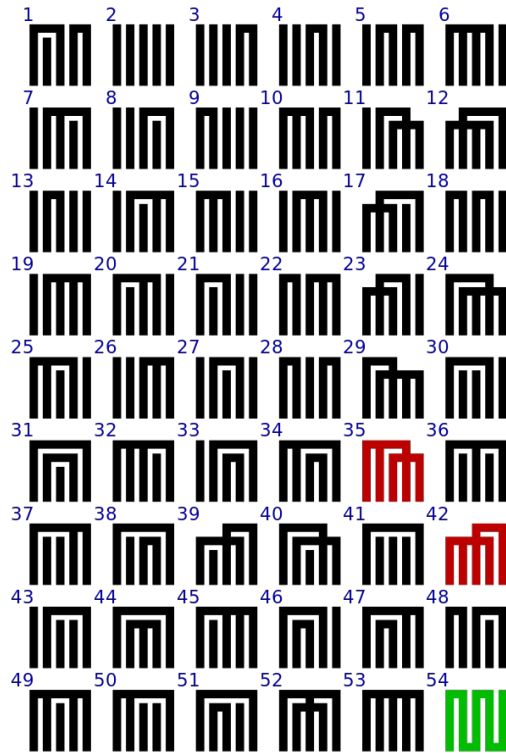
The proposed model has other complications like not being very intuitive to understand. In effect, the complexity would deter software designers from supporting the system and making use of UPA instead, like they already do.

It would also require a large overhead to map the sequences to the glyphs. A rendering system that “deduces” the glyph from a given sequence is bound to be incomplete, because it can’t distinguish certain different glyphs as different sequences (e.g. the sequence 3,5,1,4,2 has two very different looking glyphs), requiring the ad-hoc variation selector, which up to this point, has assumed each sequence has at most one variant.

Even the shortest sequences have four characters, which if for some reason were to be interrupted (like a line break or a missing termination character), would look either like tofu, a bunch of numbered control characters or no glyph at all. In other words, there would **NOT** be a good fallback rendering that at least signals the intended symbol (without consulting an external source, defeating the purpose of encoding). At that point, even attaching pictures seems like a better alternative.

**My proposal.** I propose to encode the symbols as atomic characters, which would be the 5 Sanshukō patterns, followed by the 16 Keisukō patterns and finally the 52-57 Genjikō patterns.

Five Genjikō and one Keisukō patterns are considered variants of other patterns, so they could either be encoded as SVS’s or atomically. I would recommend encoding the variants atomically, because it’s not obvious to the average user that one glyph counts as a variant of another (not unless they are colored). “The Tale of Genji” utilizes at least two pairs of variants, so there is evidence that different meanings can emerge from different glyphs in otherwise equivalent symbols (see figure below).



Symbols for the Tale of Genji chapters with the red pair being variants of each other and the green character being considered a variant of #54

Source:

[https://commons.wikimedia.org/wiki/File:Genji\\_chapter\\_symbols\\_groupings\\_of\\_5\\_elements.svg](https://commons.wikimedia.org/wiki/File:Genji_chapter_symbols_groupings_of_5_elements.svg)

The naming scheme for the characters could either follow the Japanese semantic meaning or the sequence specified by the author of L2/22-117. The first option runs into trouble for the Keisukō patterns, given they have two equally valid meanings. While the Bell numbering sequence does nothing to inform the user about what the character in question *is*, it at least is robust to multiple meanings and changes thereof. Another possible naming scheme would just give a sequential serial number to the symbols, using a well agreed upon order for the symbols.

Here I give examples of the three naming schemes using the pattern 𠄎 as an example:

1. Semantic
  - a) KODO LINEAR PATTERN SNOW-IN-THE-LONELY-TIME (\*)
  - b) KODO LINEAR PATTERN KO MINENOYUKI (\*)
2. Bell Number
 

KODO LINEAR PATTERN 3-2-1
3. Serial
 

KODO LINEAR PATTERN SANSHUKO 3 (‡)

In any case, the variant form can be distinguished by appending the term VARIANT FORM, ALTERNATE FORM or FORM TWO.

A block of five rows should suffice for this purpose. If the need arises to encode the other 48 larger patterns, an “Extended” block of four rows can be added in the future, with a whole row ready in case more patterns are discovered.

\* These readings were obtained using Goggle Translate on the given string (孤峯の雪) and therefore is likely to be mistaken; this is just intended as a mock-up, with the correct names determined by the experts

‡ The naming of this would group the patterns by prefixing one of the three names (SANSHUKO, KEIZUKO or GENJIKO) to the serial number, so there would also be a KEIZUKO 3 and a GENJIKO 3

**In summary.** I have argued against the complex encoding model proposed in L2/22-117 in favor of encoding the patterns atomically in two phases. The reason of doing in two phases instead of one, is to allow for a better preparation to encode the larger patterns: Look for more sources as well as the planned use of them. Not encoding the larger patterns is not a big loss in my opinion, but including them isn't going to occupy much space either.

###

\* These readings were obtained using Goggle Translate on the given string (孤峯の雪) and therefore is likely to be mistaken; this is just intended as a mock-up, with the correct names determined by the experts

‡ The naming of this would group the patterns by prefixing one of the three names (SANSHUKO, KEIZUKO or GENJIKO) to the serial number, so there would also be a KEIZUKO 3 and a GENJIKO 3