L2/03-387

Title: **Chinese Character Description Languages (CDL)**
Source: Richard Cook <rscook@unicode.org>
Status: Expert Contribution
Date: 2003-10-26-22:22
Action: For consideration by UTC and IRG

The current document is an extract of nine pages from my 2003 PhD Dissertation. These pages are intended to introduce some aspects of Wenlin Institute's CDL, a system which holds much promise for management of CJK Unified Ideographic character data. This document augments part three of L2/03-286 (Cook & Bishop). Additional documents in this series will follow, including a draft of the full XML CDL specification.

References

Cook, Richard S. 說文解字-電子版 *Shuo Wen Jie Zi - Dianzi Ban: Digital Recension of the Eastern Han Chinese Grammaticon*. UC Berkeley, Dept. of Linguistics, 2003.
L2/03-286 <http://www.unicode.org/L2/L2003/03286-cook2.txt>.
L2/02-221 <http://www.unicode.org/L2/L2002/02221-cdp-idc.pdf>.

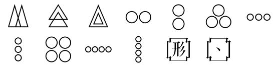
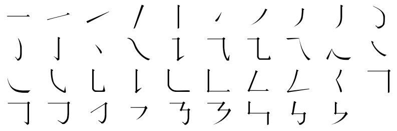
Abbreviations

SW = *Shuo Wen Jie Zi* (121 AD text, see above).
HDZ = *Hanyu Da Zidian* (see the kHanYu header in Unihan.txt).
SBGY = *Song Ben Guang Yun* (see the kSBGY header in Unihan.txt).

2.5.5 Chinese Character Description Languages (CDL)

In order to quantify the relations among character forms, to enable ancient texts to “talk to each other”, I have employed three Chinese Character Description Languages (CDL)[41] in association with SW and Unicode-based variant mapping tables. One of these CDL’s, the CDP system developed at Academia Sinica in Taiwan is a Big5-based component system. The second is the IDS system of the Unicode Standard. The third, and by far the most advanced system that I am aware of, is a stroke and component-based system being developed by Wenlin Software (Bishop, 2003), to which I am (and have been) contributing, with special regard to extending its applications for computer standards work. Some basic information about these three CDL’s is tabulated in Table 2-27, along with sample elements used in each.

Table 2-27. Some Elements of Three Chinese Character Description Languages

Name	*Creator*	*Elements*
CDP	*Chinese Document Processing Lab., Institute of Information Technology, Academia Sinica, Taiwan*	
IDS	*Unicode 4.0, Unicode Consortium*	⿰ ⿱ ⿲ ⿳ ⿴ ⿵ ⿶ ⿷ ⿸ ⿹ ⿺ ⿻
CDL	*Wenlin 3.x, Wenlin Institute*	

The CDP and IDS systems both have extreme limitations, both in regard to their intended purposes and also in regard to their basic elements. For this reason, they are not discussed here, and the reader is referred to the Glossary entries for CDP and IDS for further details. Due to legacy data issues, the CDP system was however employed by me in preparation of the HDZ and SBGY data appearing in the present study, and for this reason this system is discussed in some detail in Section 3.2.3.2.

[41] The name “CDL” for WL’s stroke-based system was coined in a discussion I had with Kenneth Whistler in 2002.

Wenlin's stroke-based system has the potential to become a full-fledged CDL such as might be adequate for handling all encoding issues, and though we cannot look at details of its software implementation here, some of the distinctions which it makes serve as the basis for the full CDL described here.

2.5.5.1 An Extensible Set of Basic Script Elements for *Han*

An extensible set of basic script components for *Han* is envisaged as a means for quantifying the relationships among characters and also among glyph variants, for indexing and encoding purposes, and for the purpose of building variant tables to be used for investigating the inter-relations among texts and inscriptions.

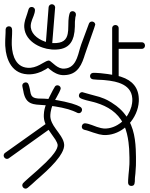
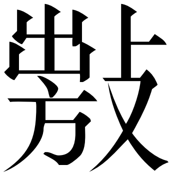
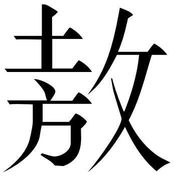
The set of basic stroke types listed above in Table 2-10 (repeated in Table 2-27), augmented with other more rare basic stroke types constitutes the basic set of distinctive features. The members of this set, used in accordance with a standard Cartesian coordinate grid (rather than the CDP or IDS type of spatial relation operators), and in association with a few transformations necessary for rare characters (*cf*. Table 2-11), provide a means for unambiguous mathematical description of all Chinese characters. By means of such descriptions, it is possible to automate the identification of component structures, and to quantify the differences among character forms.

These basic script elements and their associated transformations (treating positioning, scaling, flipping and other stroke modifications all as "transformations") altogether constitute the set elements, and this set is "extensible" insofar as the only limits on set membership are practical ones. That is to say that if someone cares to make a distinction which has not already been made, then the CDL is able to accommodate addition of that new distinction. The addition of a rare stroke type would be one example of the CDL's extensibility. The addition of a "flip horizontal" transformation to the CDL (*cf*. Table 2-11) would be another example of its extensibility, in that the CDL does not at present have any such transformation.

2.5.5.2 CDL Descriptions: Examples

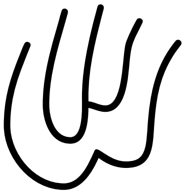
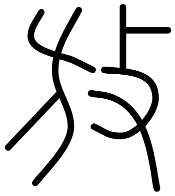
For particular examples of CDL descriptions, let’s revisit in Table 2-28 below the forms first exhibited in Table 2-19:

Table 2-28. Members of the 敖 *áo* Graphical Variant Class (reprise)

			敖
{eci} == {gjb}	[U+22f8d]	[U+e109]	[U+6556]

The traditional componential analysis of the seal form gives us only two components for this character, as follows:

Table 2-29. Seal components of 敖 *áo*

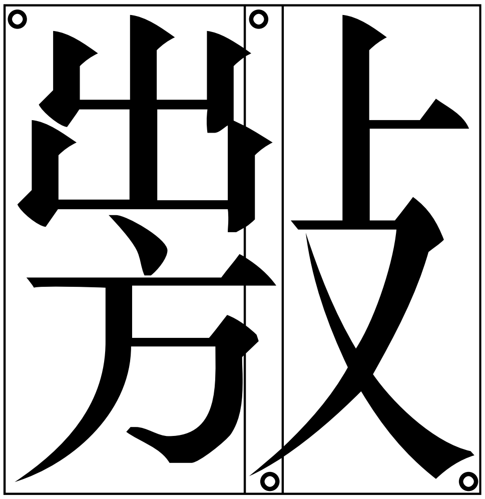
	

It is apparent here that the form of the component {ech} /pʲɑŋʔ/ (313.04), /pʲɑŋs/ (426.48) in the compound graph has undergone 省 *sheng* ‘contextual distortion’ in combination with the {gja} /tɕʰiueis/ (356.05), /tɕʰiuet/ (474.46) component. This distortion exemplifies the distinction between *etymographical componential analyses* on the one hand, and *simple structural componential analyses* on the other. A *kai* representation of the above analysis in Table 2-29 would use simply 出 and 放 components.

And yet, attempting to give a structural representation of the form using a CDL of some type requires that the two parts of *one* of the etymographic parts of the character be transformed (scaled, distorted) *separately*. That is, in order to place 出 over 方 , we must first separate 方 out of the compound etymographic component 放 (which, by the way apparently has no encoded ultra-*fanti* form with the full 攴 {dge} /pʰuk/ (452.18), /pʰok/ (466.09) component). This kind of transformation (independent scaling of separate parts of a single component), while not programmatically impossible, is however rather inconvenient. Ideally, one might like to preserve the etymographic component analyses as much as possible in one's CDL descriptions, and yet it is rather more convenient to simply treat the two things distinctly, as separate though related tiers. SW's 省 *sheng* etymographic components often omit component elements entirely (rather than simply distort them). And as we have seen (Section 2.5.3), there is often no consensus on component analysis itself.

One solution is to simply ignore *theoretical* (etymographic) explanations of the character for the purposes of the CDL (etymographic information can be stored elsewhere), and worry only about the character's *actual* appearance in the specific context of its occurrence. This presents us with yet another problem, since there is in fact no such character as ⿱出方 (at least there may not have been until now).[42] Figure 2-1 below[43] illustrates the simple CDL description of two components in left-to-right combination. Note that the two components ⿱出方 and 攴 each have bounding boxes, and that each bounding box has "control dots" at its upper left and lower right corners (to control component scaling within the grid space).

Figure 2-1. 𢾍 CDL for Ultra-*fanti* 𢾍 [U+22f8d], with PUA component



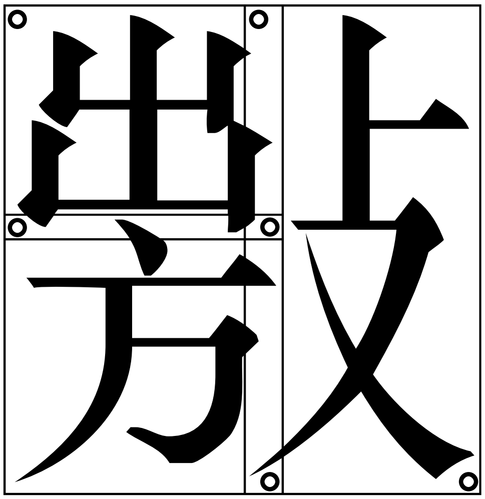
This structure ⿱出方 with 出 over 方 is not an independent element in the script, but might be identified as simply a highly bound graphical component (according to the usual left-to-

[42] But cp. the left side of {gyy} /ʔʲɑnʔ/ (280.16), /ʔʲanʔ/ (295.19).

[43] CDL Figures in this section are based on WL's implementation, and were produced using *Wenlin 3.x* and *Fontographer* 4.1.4 software; see *Bishop*, and *Altsys*, both in the Bibliography, and *fn. 45* below.

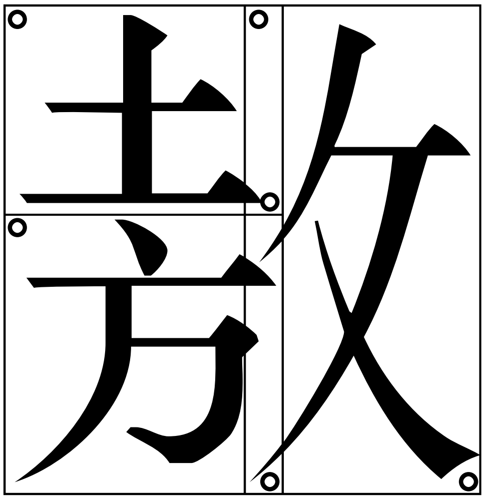
right structure of characters with the 攴/攵 radical). We can assign this form to PUA, as I have done for the present discussion: [U+e0c0]. And yet this type of PUA usage is rather pointless for highly bound structures which might be decomposed into non-PUA (in this case BMP) components, as in Figure 2-2.[44]

Figure 2-2. 𢾍 CDL for Ultra-*fanti* 𢾍 [U+22f8d], with BMP components



Similarly, for the second square-script form in Table 2-28, rather than defining a nonce PUA component for the left-hand side, we simply resort to elements of a somewhat lower-level description of this bound form of the BMP graph 敖 [U+6556]. See Figure 2-3.

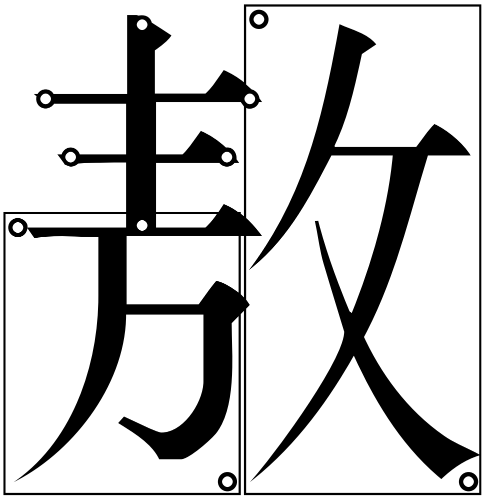
Figure 2-3. CDL for PUA graph [U+e109], with BMP components



[44] To transform multiple components, a grouping mechanism might be used to associate components.

The CDL description of [U+e109] (PUA graphical variant of 敖 [U+6556]) given in Figure 2-3 above) specifies the components 土 [U+571f], 方 [U+65b9], and 攵 [U+6535], each with the (x, y) coordinates of its bounding rectangle (each with two “control dots” at the corners) within the grid space (bounding box) of the composite character as a whole. This description is not self-contained. Rather, in order to display the composite character, the language interpreter uses the CDL descriptions of each of the three components. In general, components can be any characters that are themselves defined as sequences of basic strokes and/or simpler components.[45]

Figure 2-4. 敖 CDL for BMP graph 敖 [U+6556], with 3 stroke components



Similarly, the CDL description for the Unicode 3.0/4.0 reference glyph of BMP graph 敖 [U+6556] specifies the first three strokes as basic stroke types 一, 一, and 丨, each with the coordinates of its starting and ending points (note the positions of the control dots at stroke extremities), and possibly using stroke modifiers (note stroke 3), and then specifies the two components 万 [U+4e07] and 攵 [U+6535] with their bounding rectangles.

All of the above CDL descriptions can of course be reduced all the way down to the stroke level, since all components are comprised of only basic stroke types to which trans-

[45] This is an expansion of Cook 2003b; portions of the discussion of Figure 2-3 and Figure 2-4 derive from discussion with Thomas Bishop (see the acknowledgements in Cook 2003b).

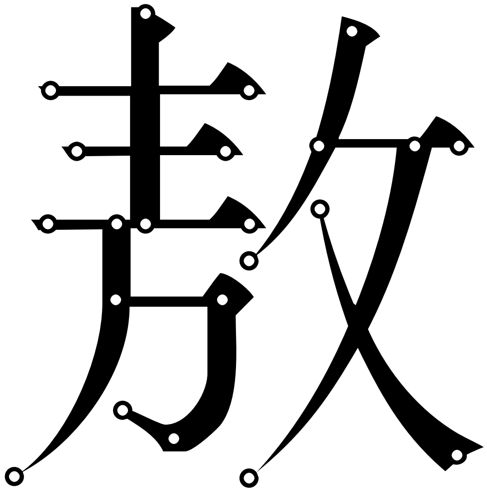
formations of various kinds may be applied. Thus, the 10 strokes of the graph for 敖 [U+6556] in Figure 2-4 can be reduced to the sequence given in Figure 2-5 below.

Figure 2-5. Sequence of strokes for Figure 2-4

一	一	丨	一	㇆	丿	㇒	一	㇒	㇏
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*

Figure 2-6 below presents the graph at its lowest level of decomposition, where all strokes are represented as independent entities (note the control dots at all stroke extremities, and some stroke types have more than two controls).

Figure 2-6. 敖 CDL for BMP graph 敖 [U+6556], with 10 stroke components

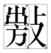
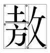


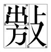
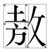
2.5.5.3 CDL Constraints

Bear in mind that the CDL for a given script entity is not simply a succession of loosely defined basic stroke types, but that each stroke type is rigorously defined as a specific sequence of straight and curved segments, that each stroke type has a specific range of behaviors (allowable transformations), and that in a particular usage instance each stroke type has associated features which define its context (coordinates and transformations).

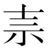
Specific instantiations of a given type therefore provide sufficient information both for identification of the type and also for quantification of the degree of variation among members of a particular type class. So, for example, stroke type A is readily distinguishable from stroke type B, due to the properties of each. And stroke type A with transformation type X is readily distinguishable from stroke type A with transformation type Y, due to the properties of the transformation.

Likewise, when a given type is employed in a particular composite structure, the presence or absence of the type, or the presence or absence of a particular transformation of the type, may be sufficient grounds for distinguishing the two composite structures.

Comparison of Figure 2-2 with Figure 2-3 may reveal, for example, that the differences between the CDL for  and  lie in the upper left component, and in the form of the right component, and that otherwise all relative proportions (component transformations) are identical, as is the third component 方 . This information alone might be sufficient to indicate a possible relation among these two forms. If however additional information is added to the mix, such as information on allowable or known variant component shapes, then the possibility of connection becomes even stronger.

We know in this case that 支 / 攵 component variation alone is never sufficient grounds for exclusion of a variant relation among two forms. If a computer program evaluating the possible difference among  and  is given this additional information, then the suggested relation between the two forms would be quite strong.

2.5.5.4 CDL-Driven Inferences, VarClass Determinations, and Unifications

We might infer from this that 出 / 土 component variation is also evident among other encoded forms, *i.e.* that 土 is a simplification of 出 in other compounds as well. Searching our component data for characters with this 出 component, and looking at their variants in our variant mapping tables, we find that this is in fact the case. For example, in writings of characters with the 祟 祟 {adi} /siueis/ (351.01) component, we sometimes find this written as  instead, as in writings of 款 {izc} /kʰuɑnʔ// (285.48). Note the second and third members of the triple variants listed here: 歀 款 款 .

From our initial inference regarding 出 / 土 variation, we now know that this can be extended to 出 / 土 / 木 component variation (note that variant mappings here and elsewhere are dependent upon the HDZ entries, in this case those for 歀 𣢾 款). We learn from this also that not only does 𡋬 sometimes vary with 柰 , but 柰 varies with 奈 in some compounds, so that the chain of related component forms has now become 祟 𡋬 柰 奈 . As independent characters, the preceding four forms may all have distinct usages, but in compounds the usage of one of these four in one text may be interchangeable with the usage of another of these in another text.[46]

It is clear then that variant mapping has implications not simply at the character-to-character level of mapping, but also at the *character-component*-to-*character-component* level of mapping. Variant unification (that is, the determination that there is non-distinctive variation among varclass members) can be undertaken at the component level, with either higher or lower level component descriptions.

Also, although the stroke order for Chinese characters is usually quite well defined, there are exceptional cases in which there are competing stroke orders. The CDL descriptions themselves are sensitive to stroke order, and yet stroke order might also be ignored for certain purposes, *e.g.* in variant mapping.

Up to this point we have seen elements of a CDL and how these elements may be employed to categorize the relations among script entities. We have also seen how such categorizations might be useful for certain purposes, including indexing forms, identifying, cataloguing and analyzing character variants. As we conclude this Chapter and move into the next, we shall consider specific extended examples of CDL usage for the purposes of historical linguistics.

[46] To anyone who has worked on comparative semantic data (*e.g.* Tibeto-Burman gloss data in the STEDT databases), this kind of progression must seem familiar.