Digital writing of hieroglyphic texts

Serge Rosmorduc

Handbook of Digital Egyptology: Texts

UAH MOA 01

Handbook of Digital Egyptology: Texts

Edited by Carlos Gracia Zamacona & Jónatan Ortiz-García

Universidad de Alcalá | EDITORIAL UNIVERSIDAD DE ALCALÁ

Con el patrocinio de la Comunidad de Madrid, Proyecto *The Earlier Ancient Egyptian Mortuary Texts Variability* (www.mortexvar.com), Programa Atracción de Talento 1 (2018-T1/HUM-10215).

Plaza de San Diego, s/n
28801 Alcalá de Henares
www.uah.es

I.S.B.N.: 978-84-18979-09-5
Depósito legal: M-34533-2021

Composición: Solana e Hijos, A. G., S.A.U.
Impresión y encuadernación: Solana e Hijos, A.G., S.A.U.
Impreso en España

Table of contents

3. Digital Writing of Hieroglyphic Texts

Serge Rosmorduc
CEDRIC/Vertigo/Conservatoire National des Arts et Métiers, Paris

Abstract
After a short review of the history of mechanical and computer-based hieroglyphic text typography, we concentrate on the problem of sign encoding. We highlight the difficulty to choose the right grain of detail to encode the hieroglyphic system, i.e. which variations to discard and which to keep. The problem is discussed at length on a practical example, stela Cairo CGC 34504, which features a number of unusual signs.

Keywords
encoding – hieroglyphs – word processor – sign thesaurus – digital philology.

1. Introduction

Since the mid-XIXth century, Egyptologists have had the choice between two techniques for inserting hieroglyphs in their texts. Typography and autography each had their strong points. Handwritten hieroglyphs are very versatile, but most scholars are not as skilled as Nina de Garis Davies or J. J. Clère. With the advent of computers, and especially since the 90's, the use of typeset signs, at least for in-text quotations, has been much easier and cheaper, becoming the standard. Electronic publication of hieroglyphic texts also allows the creation of searchable databases of hieroglyphic texts, even if their development has been slower than expected.

Hieroglyphic fonts were created as early as 1832.[1] Their use was relatively expensive, and by the 1950's, some scholars had suggested to use codes instead of actual printed glyphs,[2] for reasons of costs. The increasing availability of computer resources, first as university mainframes, then as personal computers, introduced a new dimension. Rolf Gundlach and Wolfgang Schenkel introduced the M.A.A.T System,[3] a pioneering work in annotated databases of Egyptian texts. M.A.A.T was strictly a database system, however. It had no typographical functions; it included an encoding of the hieroglyphic texts, which was expanded in later articles by W. Schenkel,[4] but this encoding was used for search purposes only. No attempt was made to extract a visual rendering of the text. GLYPH, by Jan Buurman,[5] was the first system able to actually print hieroglyphs, and, through its encoding, the ancestor of most existing hieroglyphic editors. It's the source for the so-called Manuel de Codage,[6] which is the basic format used even by current hieroglyphic editors. A number of other softwares were developed from 1970 to 1990, often linked with specific publications (see the article by Mark-Jan Nederhof in the present volume, as well as Roberto Gozzoli's publication).[7] The very end of the 80's saw the development of the first user-friendly softwares in the domain, in particular MacScribe, and Glyph for Windows, later Winglyph. The original sign selection from the Manuel de Codage was enriched in the Hieroglyphica to cover the actual fonts available in Glyph, and the signs which add been added for a number of important publications.[8]

Inscribe, by Bob Richmond[9] was published in 1994 and had a very strong integration with Windows through "Object Linking and Embedding", which allowed a seamless integration between the hieroglyphic editor and other softwares.

Most of the systems described above where developed with traditional edition in mind, hieroglyphs being considered as a special kind of text. Although they can write signs in column, those systems don't provide a way to mix line-oriented and column-oriented text, nor to mix pictures and texts. A number of softwares have addressed this need, most notably Visual Glyph and IGlyph.[10] The Revised Encoding System format[11] improves upon the Manuel de Codage; it has influenced the latest Unicode system, and it can be expected that it will inspire other editors.

---

[1] Janssen 1973: 59, n. 1.
[2] Janssen, o.c.: 65.
[3] Gundlach – Schenkel 1970.
[4] Schenkel 1984a; 1984b; 1984c.
[5] Buurman – Schimmelpennick van der Oije 1976.
[6] Buurman et al. 1988.
[7] Gozzoli 2013.
[8] Grimal et al. 2000.
[9] Malek 2005.
[10] Subotic – Lapp 2008.
[11] Nederhof 2002.

The next step for hieroglyphic editors, as for most software, is probably the move to the web. It is now possible to create powerful user interfaces using the Javascript language, even if some crucial elements needed for a word processor is often limited. Web browsers, as software platform, have to face huge security constraints, which prohibit, for instance, full access to copy/paste functionalities. However, for anything related to data sharing and teamwork, it seems the way to go. Many interfaces have been developed to integrate hieroglyphs in Web Pages. Some use the PHP language, for instance the opensource WikiHiero,[12] which is used in Wikipedia, or the system behind the Rosette website.[13] Javascript systems have also been created; they are more versatile and can in theory be used to create mobile applications as well. The most powerful system currently available is the implementation of RES in Javascript.[14] Concerning the Manuel de Codage, a partial implementation is available.[15]

2. The Manuel as a catalogue of glyph

The Manuel de codage was a very important step forward. After all, it's 35 years old, and still the basis of most systems. The third edition of the Manuel in 1988 was an important step: it included the sign list designed by Jochen Hallof for the needs of the Berlin dictionary.[16]

The base of the system was the sign list in Gardiner grammar (without the 1953 extension to the fonts, which were added by Bob Richmond in Inscribe). The multiple sources for the sign list included most of the existing fonts (Gardiner, IFAO), and the Berlin Wörterbuch files. It was geared toward databases and tried to group multiple equivalent variants of a sign under the same code.[17] However, this first structuration was left aside when actual implementations of the Manuel were released, as the demand was mainly for a system to print hieroglyphic texts. Hence this original structure is absent from the sign list of the Hieroglyphica.[18] Text databases, especially in encoded hieroglyphs, are very time-consuming, and, until about 2005, they generally lacked academic recognition (apart from the TLA, which however used transliteration instead of hieroglyphic transcription). The Manuel, also due to problems of funding, lacks a detailed description of the signs. Indirect information can be retrieved by looking at the 1988 version, which displays signs from different sources, and trying to guess the sign origin.

---

[12] Blanchard – Semenik 2004.
[13] Euverte – Roy 2013.
[14] Nederhof – Field 2015.
[15] Rosmorduc (2020) 2015.
[16] Hallof 1988.
[17] Hallof, o.c.: 14 and 17.
[18] Grimal et al. 2000.

3. Approaches to digital hieroglyphic text processing

Even the most careful facsimile of an original text will betray the original in one way or another. It means that any scientific production from a source is an interpretation of this source, and that the scholar must make conscious choices about it. A good iconography, including photographs and facsimiles, is eminently desirable. Even for printed volumes, the production of a companion website might allow the distribution of high-resolution pictures, if possible. The long-term preservation of such websites is extremely important, and many institutions provide dedicated hosting for such purposes.

Outside of facsimile, most scholars will nowadays use a hieroglyphic editor. The result can be incorporated as quotation in a running discussion or used as an easy-to-read version of the original text. Outside of publication, texts can be encoded to produce full text databases. This is a lengthy task, but very useful for the Egyptological community.

3.1. Signs selection

When encoding a text, the first step should be to clarify the purpose of the encoding, and how precise one needs to be. It's not always easy to make a choice. Encoding takes time: a text created for printed publication can also become a convenient searchable reference.

The most serious problem for the encoder is the choice of signs, especially when variants are available. When in doubt, a good rule of thumbs, especially for novice encoders, is to use the simplest possible encoding, especially for phonograms. This is especially true when the original document was written in hieratic.[19] It's also something to consider when the text is encoded from secondary sources, such as the Urkunden or the Ramesside Inscriptions. Some variation there, for instance  for might simply be due to the hieroglyphic handwriting of the Egyptological author. It doesn't mean one should be insensitive to sign variations; only to make the encoding of sign variation a conscious decision.

The hieroglyphic system being open-ended, and variation being quite difficult to define, an encoder will always find cases where the available sign selection doesn't fit the original text. Three solutions will be available. First, the "standard" version of the sign can be selected. The problem here is that there is no good definition of what the "standard" is, but in most cases, commonsense will be to pick the code in the original Gardiner set. This is a strong normalization, which implicitly conveys the meaning "for this sign, only the linguistic value is relevant". Then, the "closest"

---

[19] Polis – Rosmorduc 2013.

available version of the sign may be chosen. This is in a way the most complex approach. If a perfect fit is not available, all typographic signs will differ from the one in the source.

The question is to decide if one of the available signs has the features the encoder finds meaningful in the original text. For instance, if a text contains an enemy tied by the neck, the I might consider using  (A422). But A422 has a number of features, among which [TIED BY THE NECK] seems less relevant than [PEOPLE OF THE SEA]. I will probably prefer a more generic enemy sign. One of the problems of the available encoding system is that it doesn't allow the encoder to be explicit about his choices. It would obviously be cumbersome to force such specifications in all cases, but it would be useful to be able to encode explicitly difficult choices.

In Egyptology, orthographic studies, and more generally studies in the history of the signs, are still relatively in their infancy. In particular, we lack statistical data about sign uses. The availability of large encoded corpora might improve the knowledge of the use of writings, the appearance of new shapes, the reuse of archaistic signs, etc. However, if the encoded data is tainted by approximative encodings, the results may be meaningless. An encoder who chooses the closest approximative match will end up with a text which gives a deceptive sense of epigraphic accuracy.

The last option is to create a new sign, which is possible with some softwares. It should be done with care, because it takes time, on one hand, and it makes the sharing of texts difficult, on the other hand.

4. Extending the sign list

A number of hieroglyphic word processor allow the user to create their own signs. It's mostly the case of VisualGlyph, VectorOffice, and JSesh. The reason is twofold: the set of possible hieroglyphic signs is not limited on one hand, and the existing sign-lists, originally based on the historical lead-fonts, have a very partial coverage of some corpora on the other hand.[20] A hieroglyphic font can be extended in a number of different ways. The first approach would be to use a regular truetype and a font editor such as FontLab or the FontForge open-source software. The main problem is that most operating systems expect fonts to be rather static objects. A font is not something which changes during a user session: it's either installed or not installed. With some systems, font modifications will only be visible after a user disconnects and reconnects.

A second approach will be to consider the signs as graphics, managed by the hieroglyphic text processor itself. For high quality rendering, they should be vector graphics (made of lines and curves). Softwares such as tksesh and VectorOffice

[20] Collombert 2007; Meeks 2013.

include a dedicated editor. JSesh uses a standard format, SVG, which can be created by Illustrator or Inkscape.[21]

A sign may need to be drawn in two quite different cases. In the first case, the encoder estimates that the sign is not present in the available fonts and needs to be added. The second case is when the near-facsimile rendering of the sign is needed. It's the case, for instance, when a few hieratic signs are unreadable, and rendered as-is. It might also be used to render some drawings which are mixed with the original text. The amulet  in P. BM EA 10411, vo 3–4 (= LRLC pl. 4)[22] is such an example. For those signs, which are structurally hapaxes, the ideal system is to include their very drawing in the encoded file. Drawing oriented software will tend to allow this. A JSesh user will currently need to send the SVG drawing of specific his signs as separate files alongside his text, which is a bit cumbersome.

Adding a "regular" new sign is in a way more complex. Graphically, the sign creator is faced with two contradicting requirements. The sign must both be true to the original and mix gracefully with the existing font. Illustrator or Inkscape will allow the sign creator to combine parts of existing signs for that purpose. A full demonstration of the process is given in the JSesh video tutorials. High level of detail can lead to problems, especially as some signs may only be known from very rough examples. For instance, the JSesh sign  (A250) is pretty anachronistic. The actual sign is a quadriga, which appear in Hellenistic stelas such as Louvre C123. Here, the mix of New Kingdom sources and Greek-era motive gives a four-horses Eighteenth dynasty chariot – which should probably be fixed at some point.

It's sometimes possible to mimic a missing sign with a clever combination of existing ones, and of the free positioning system of some editors. For instance, if one needs a seated god  with the  (S67) crown, the JSesh "group building" facility allows to approximate it in a way: , by stacking and scaling the two glyphs in the editor itself. Now, this approach is only reasonable if printing is the only purpose of the encoding (and the printed result is usually far from perfect). As far as encoding goes, the fact that there is only one sign is lost. Search functionalities will consider that the text contains the corresponding sequence of glyphs, not a combination of them; any automated processing (for which there are now relatively good prototypes), such as automated segmentation or automated transliteration would most likely fail. The system of N. Stief[23] had an actual system for meaningful sign combinations, in which primitive elements (body, crown, scepters) could be combined. This latter approach is sound, as the components are

---

[21] Rosmorduc 2009a.

[22] Rosmorduc – Polis 2015: 160.

[23] Stief 1988.

designed for this very purpose. With other systems, pounder whether or not a new sign is actually needed and, if the answer is positive, draw it.[24]

If a new sign is created, it should be given a code. The problem, if the text is shared, is that two different signs should have two different codes. JSesh provides a system where each user can ask for a numeric user ID, which will appear as a kind of signature in the code of his signs. A prefix "US" (for "User Sign"), followed by the user ID, is used in front of each personal code. There are also a number of rules for suffix codes, which can mark the sign as significant variant (VAR), as mere different graphical rendering (VER), or as completely new signs (EXT).

For instance, the sign 𓀀 has the code US1A33VARA, which means it was created by the user with id 1, that it's a variant of A33. The final "A" leaves the door open of further variants. The code is very long, but the sign can be typed in JSesh without using the full coe.

4.1. The future of sign lists

The current source for sign codes, the Hieroglyphica, lacks documentation. Two existing projects, the "Paleographie hiéroglyphique" at the IFAO and the Thot Sign List Project («Thot Sign List» 2020), should improve the situation significantly.

An improved architecture for the sign encoding would allow the encoder to express his intent when encoding a particular sign. This entails a multi-layered system for codes. There seems to be a consensus toward three levels of details – their designations and exact delimitations are not yet completely settled.

A generic code which would designate the sign for its broad linguistic value, without regard for its specific details. In this respect occurrences, 𓎛, 𓎜 or 𓏏 would all be encoded in the same way.

More specific code would designate the significant variants; the criteria for such code would be based on explicit features, expressed in the sign description. STANDING and KNEELED would differentiate 𓏏 and 𓎛.

Actual graphical realization would correspond to a third level of code. This would allow one to give a code to a complete hapax, to an unreadable scribble, or even to the specific shape a sign has in a particular font.

In this approach, the "standard" shape for a sign, as 𓀁, could be rendered because of three different encodings : it would appear as an "unmarked" rendering for the "seated man" sign, as a rendering for the "seated man with closed hands" variant, and as a shape in JSesh original font.

The system appears complex and cumbersome, but it could be made mostly transparent for users – in most cases, the first level would suffice for instance.

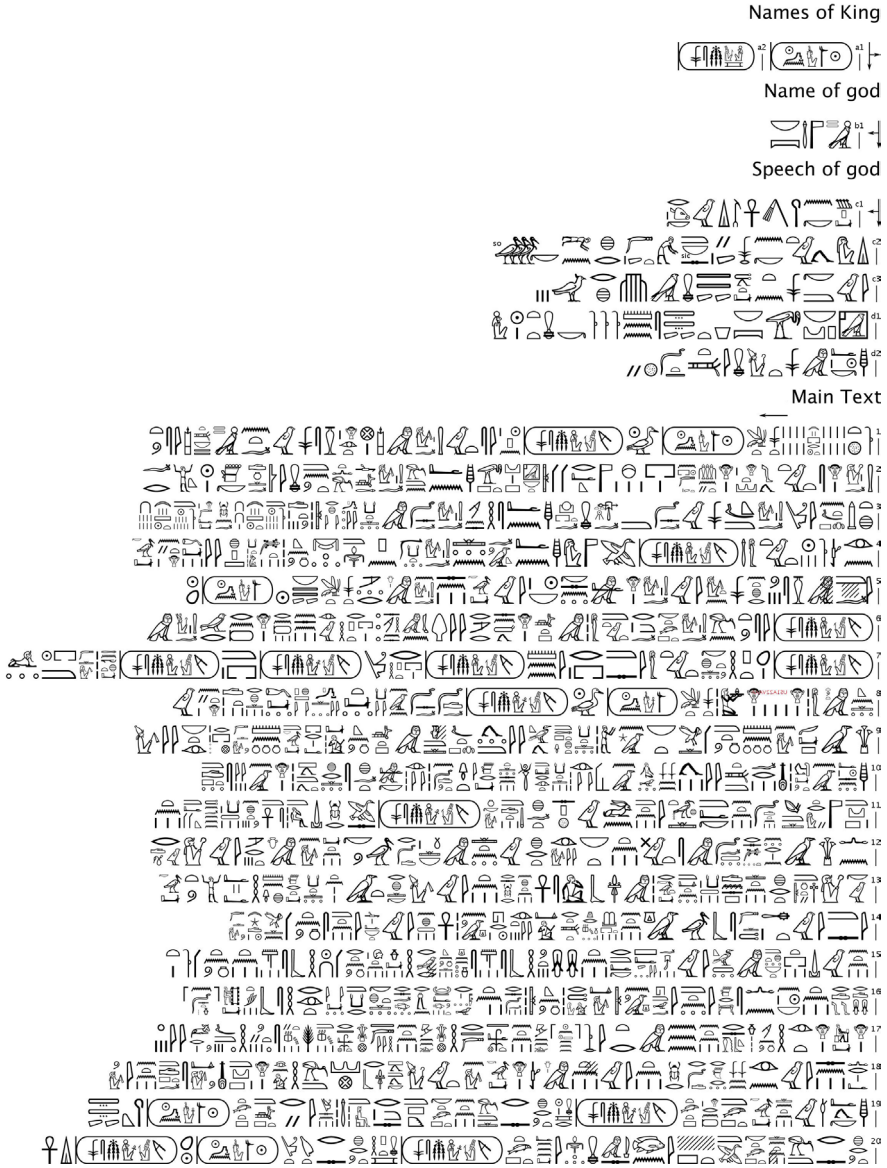
---

[24] Rosmorduc 2009b.

5. Practical examples with JSesh

We will illustrate the problems linked to digital text edition with two examples: the publication of an ostracon, for facsimile like edition, and a thorough study on the edition of a monumental inscription, Stela Cairo CGC 24504 (Fig. 1).

Fig. 1. Stela Cairo CGC 34504



5.1. Respecting the text layout: diplomatic edition

Traditionally, ostraca are published both in facsimile and diplomatic hieroglyphic transcription. By keeping the same layout in the hieroglyphic transcription and in the facsimile, it is easier for the reader to check the transcription. For this purpose, a drawing-oriented software like IGlyph[25] or its forerunner VisualGlyph would be more convenient.

There is, however, a way to use JSesh for this kind of work. The principle is to type the text in JSesh as usual, and then to export it to a vector format which can be processed in a software like Inkscape or Adobe Illustrator. SVG (Scalable Vector Graphics) is probably the best format for this process. The resulting graphical file can then be edited and enhanced as a simple drawing. A detailed tutorial for this process is available on the JSesh website.[26] The current downside of this system is that the edited file is a plain vector drawing. It's perfectly fit for printing, as SVG is a vector graphic format, but it loses any information about the text it contains. The resulting document can't be searched for text.

Some experiments have been done in order to expand JSesh with advanced graphical capabilities, which would allow one to have a document mixing hieroglyphs and drawing, which would be both searchable and editable, but there are currently more pressing issues.

5.2. Dealing with Variation: Stela Cairo CGC 34504 (Figure 1)

In this section, we will examine a document, the stela of Year 8 from Manshîyet eṣ Ṣadr (= KRI II, 360–362; Stela Cairo CGC 34504), and highlight the decisions a scholar faces when encoding it in the framework of the Manuel de Codage.

Cairo 34504 was published by Ahmed Bey Kamal[27] and republished with a full study and a photography by Abdel Hadi Hamada.[28] It contains a discourse from Ramses II to his quarrymen and sculptors. It features a number of unusual signs, especially representations of statues,[29] which was one of the reasons why it attracted our attention. We decided to use it as an experimental ground for the present paper, to see what it takes to give the closest approximation to the original text. The result of this work is included in the text library of JSesh 7.4.

Typing a text and reading it are usually intermixed tasks; when a sign is a bit unclear, reading the text instead of sticking to its graphical aspect will often solve

---

[25] Subotic – Lapp 2008.
[26] Rosmorduc 2007.
[27] Kamal 1908.
[28] Hamada 1938.
[29] Polis – Rosmorduc 2013: 160.

the problem of choosing the right sign. Obviously, the risk with this attitude is to overly standardise the text. When faced with a problem in the choice of signs, the encoder must then analyse it, especially when he doesn't find an exact match in the font repertoire.

In the rest of this section, we have categorized the encoding issues according to their theoretical importance.

5.2.1. *Reading difficulties*

The first kind of issues one meets is very practical, but not very difficult to solve. It concerns damaged or difficult to read signs. In some cases, the problem is very light. In column (c2), for instance, the shape of  (V22) is a bit peculiar; but it looks like more an engraving problem than a significant variant, and we ignore the variation altogether.

When creating a database, large corpora like the KRI or the Urkunden are very useful. However, references to the original document allow a much better rendering of a text, even when the reading seems obvious. Most edition of Cairo 34 504 render the end of line 16,  , as "*ḳd-nḏt*", following A. Hamada; *nḏt* being a kind of large vase, known from the Pyramid Texts.[30] However, when translating the text, we considered the reading "*nḏs.t*", as in the compound word *ḳd-nḏs.t*, "potter" (ALEX 79.3177). A check on photographic pictures of the stela seems to allow the reading  (O34) instead of  (N16). The ⸢ ⸣ marks, used in classical philology for "dubious reading" are quite useful here.

In some cases, the original text editors are a bit too cautious. For instance, in line 20, the  (T34) in group nms is a bit strange and looks remotely like a  (F25). The KRI display the group as  along with a footnote about its correct reading. It's not a problem for a paper edition, but in a database-oriented approach, such care would introduce a ghost interpretation of F25.

5.2.2. *Graphical problems*

We have insisted above about the importance of meaningful features. However, when a text is displayed, some problems linked more with geometry or writing support may occur. In a number of cases, in our stela, the "stone" determinative  (O39) looks more like a square than like a rectangle, due to graphic constraints. Some encoders, especially if their concern is the visual rendering of their text, might be tempted to use the Q3 mat instead:  . It would be very problematic. The use

[30] Hamada 1938: 229, n. 4.

of Q3 instead of O39 would of course hinder any automated search of statistical processing; more, if by chance the text is displayed with a detailed version of the Q3 sign, showing the matting, the visual rendering itself will be wrong.

As in many inscriptions, the circle-shaped signs ⊙ (N5), ◌ (O50), ⊗ (049) and ⊜ (Aa1) are underspecified. The ○ (D12) sign should certainly not be used instead, as it has value of its own. The supplement to the Gardiner fonts (Gardiner 1953, p. 6) includes a "Z16" sign which has been added, first to Inscribe, then to Unicode as "EGYPTIAN HIEROGLYPH Z013, Ux133F8. The sign was added to JSesh in order to fully support the available Unicode encoding as US1Z13EXTU, ◯, and can be more easily typed using the "Z13" code, followed by a hit on the space bar. The 1953 supplement itself doesn't contain any guideline about the use of the new signs, contrarily to the JEA articles A. H. Gardiner devoted to earlier extensions of his fonts.[31] The presence of the sign in the "Z" family could hint that its intended use is to replace round signs when they don't provide any detail. But other interpretations might be plausible, for instance it might be a logogram for "circle" in mathematical papyri.

In our text, the lack of detail is related to the physical medium and the size of the signs, and not to a conscious decision to introduce some kind of ambiguity, and the most reasonable approach seems to supply the linguistically expected sign.

5.2.3. *Problems with significant variants*

In a number of cases, we had difficulties choosing the right variant in the Manuel.

In column c3, we find an unrecorded variation of G23  . The available rekhyt signs are G23 ,  (G24), (G24A) and (G24B). The sign in Cairo 34 504 has "adoring arms" but no wings. The encoder has three choices here: using G23 as an "unmarked" sign; or G24B as "closest" approximation, if one considers that the presence of "arms" is more important than the absence of wings. The third possibility would be to create a specific variant of the sign. The choice will depend on the purpose of the encoding. To ease searches, in the case of a linguistic database, the standard G23 would be reasonable. The use of G24B signals to the reader that the text contains a "fancy" variant of G23 and hints to its creative use of the writing system. However, if any statistics are computed on this basis, they will be wrong, as G24B and the actual sign differ significantly. Creating a specific sign would be the best approach to deal with this variation.

The same kind of problem occurs with sign (G41A), line 9. It's a determinative for tnw, not very readable, but clearly linked to the idea of "bird killed by a throwing stick". It seems that the wings of the bird are somehow broken or restrained, much

[31] Gardiner 1929; 1931.

like those of a bird held by an offering bearer. If we consider G41A as being the "wounded bird" sign, the graphical variant of the text, while interesting, seems too light to drive us to draw a specific sign. As in other cases, it's a decision the encoder must take, and which largely depends on the research problem the scholar has decided to address. As an aside, G41A is a good example of "partial variant" of a sign, as it covers only some of the values of G41.[32]

5.2.4. *Interesting graphical choices covered by the Manuel*

The text contains a large number of determinatives and suffix pronouns designing the king: signs (A42) (A42A) (A42B) (A43B) (A43C) (A304E). From a strict linguistic point of view, (A42) would suffice. However, the phenomenon, which is quite usual on monumental inscription, deserves closer scrutiny, and is worth encoding. The Manuel de codage, here, gives us a sound choice of signs.

In line 19, a sign similar to (A50), appears as a logogram for a royal statue. The similarity is deceptive, as it's far from the normal values of A50. The code A63A is a much better fit, even if at reduced size, one can hardly tell the difference. A50 is a "man of rank seated on chair", which is quite different from a "king seated on a cubic throne".

5.2.5. *Candidates for sign creation*

In most cases, one can get a reasonable approximate rendition of a text with the current set of glyphs. However, line 7–8 of the stela contains the sentence *mḥ=j pr Rᶜ m šsp.w ḳn.w m tw.wt ḥr* *ḥr*: "I have filled the temple of Re with sphinxes, with statues performing the -offering and the -offering". Hamada transliterates respectively as *šms-ᶜntyw* (WB IV, 484) and *jɜ.t-jḫt* (WB I, 573).[33] The Manuel de codage features a number of signs for this value, (E159) and its variants, which occur in late temples. The logographic quality of the sign, and the typological difference between the two kinds of statues displayed justify the creation of a new sign. As for the JSesh suggested rules,[34] I have used my user Id (1), taken a base code I reasonable at the time (A22, for is a statue), and expanded it with "VAR" plus a letter "A" to indicate it was a significant variant of A22, ending up with sign (US1A22VARA). As the original sign is very small, I have used full scale representations taken from the tomb of Rekhmire. This approach to font creation, which is already used for some signs in Gardiner fonts, can lead to the creation of

---

[32] Gardiner 1957: 472; Meeks 2004: 88–89.

[33] Hamada 1938.

[34] Rosmorduc 2008.

chimera, most notably the  (L3) sign, whose original is not even a drawing, but a jewel.[35]

In retrospect, it shows the problems of sign classifications. Linguistically speaking, the sign is much more a variant of  (E159) than of A22 (or, from diachronic point of view, E159 is a variant of this sign). However, the standard sign classification logic doesn't fit with this approach, as our occurrence clearly fits in the "Man and his occupations" category, whereas the late period sign is classified in the "Mammals" category.

6. Unicode encoding of hieroglyphs

A long-time project,[36] hieroglyphs were finally added to Unicode in Unicode 5.2 (2009) and extended with formatting characters in Unicode 12.0 (2019). The current sign catalogue, built around the original Gardiner Font and its extensions, allows the quotation of hieratic texts, for instance. For monumental texts, regular fonts lack flexibility in all cases. The most interesting features is that Unicode can be used in regular files (XML, Word, Excel, etc…), and processed through any computer software. It doesn't require specific engineering as Manuel de Codage files. It is expected that a future version of JSesh will include Unicode Export facilities. An innovative application, SINUHE,[37] based on Google's Japanese input system, can be used to type hieroglyphs in any software, including Word and Excel.

7. Bibliography

More detailed information about the encoding systems will be found in Mark-Jan Nederhof's article, in this volume. For sign lists, the introduction to (Meeks 2004) is a very good starting point; alongside (Meeks 2013) and (Polis et Rosmorduc 2013). (Gozzoli 2013) provides a convenient history of hieroglyphic word processors. The works of N. Stief (Stief 1985; 1988) and W. Schenkel (Schenkel 1974; 1984) are also of note. The original informatique et égyptologie, especially the first volumes, are historically quite important, but currently difficult to find. The web site of Bob Richmond, author of Inscribe, contains numerous interesting articles and experiments (Richmond 2016). The JSesh website (Rosmorduc 2007) contains a lot of information and resources about the software.

---

[35] Meeks 2010: 292.
[36] Ewen 1996.
[37] Kilani et al. 2016.

Allen, J.P. (2002). *The Heqanakht Papyri*. Publications of the Metropolitan Museum of Art Egyptian Expedition 27, New York.

Buurman, J. 1985. "Printing of Egyptian Hieroglyphs by Means of a Computer". In *Informatique et Égyptologie* 1: 13–32.

Buurman, J. – Grimal, N. – Hainsworth, M. – Hallof, J. – van der Plas, D. (1988). *Inventaire des signes hiéroglyphiques en vues de leur saisie informatique. Manuel de codage des textes hiéroglyphiques en vue de leur saisie sur ordinateur. Informatique et Égyptologie 2*. Mémoires de l'Académie des inscriptions et belles-lettres, n. s. 8, Paris.

Collombert, P. (2007). "Combien y avait-il de hiéroglyphes?", *Égypte, Afrique & Orient* 46: 15–28.

Collombert, P. (2010). *Le Tombeau de Mérérouka: Paléographie. Vol. 4. Paléographie Hiéroglyphique*. Cairo.

Euverte, V. – Roy, C. (2013). "Hieroglyphic Text Corpus: Towards Standardization". In Polis, S. – Winand – J. Gillen, T. (eds.), *Texts, Languages & Information Technology in Egyptology. Selected papers from the meeting of the Computer Working Group of the International Association of Egyptologists (Informatique & Égyptologie), Liège, 6–8 July 2010*, Liège: 111–120.

Ewen, J.B. (1996). "Proposal for the Inclusion of the Hieroglyphic Script in the Unicode Standard and ISO 10646". *Informatique et Égyptologie* 10: 53–56.

Gardiner, A.H. (1957). *Egyptian Grammar Being an Introduction to the Study of Hieroglyphs*. Oxford.

Gardiner, A.H. (1929). "Additions to the New Hieroglyphic Fount (1928)". JEA 15.1/2: 95.

Gardiner, A.H. (1931). "Additions to the Hieroglyphic Fount (1931)". JEA 17.3/4: 245–47.

Gardiner, A.H. (1953). *Supplement to the Catalogue of the Egyptian Hieroglyphic Printing Type Showing Acquisitions to December 1953*. Oxford.

Gozzoli, R.B. (2013). "Hieroglyphic Text Processors, Manuel de Codage, Unicode, and Lexicography". In Polis, S. – Winand – J. Gillen, T. (eds.), *Texts, Languages & Information Technology in Egyptology. Selected papers from the meeting of the Computer Working Group of the International Association of Egyptologists (Informatique & Égyptologie), Liège, 6–8 July 2010*, Liège: 89–101.

Grimal, N. – Hallof, J. – van der Plas, D. (2000). *Hieroglyphica: Sign List - Liste Des Signes - Zeichenliste*. Publications Interuniversitaires de Recherches Égyptologiques Informatisées 1. Utrecht.

Gundlach, R. – Schenkel, W. (1970). *Lexikalisch-grammatische Liste zu Spruch 335a der Altägyptischen Sargtexte. LL/CT. 335 A, als Probe für die Erstellung einer Sargtextkonkordanz mit Hilfe einer elektronischen Datenverarbeitungsanlage*. Darmstadt.

Hallof, J. (1988). "Probleme Bei Der Codierung von Ägyptischen Schriftzeichen". In *Informatique et Égyptologie* 5: 11–18.

Hamada, A. (1938). "A Stela from Manshîyet Eṣ-Ṣadr". *ASAE* 38: 217–30.
Haring, B.J.J. (2006). *The tomb of Sennedjem (TT1) in Deir el-Medina: palaeography*. Paléographie hiéroglyphique 2, Cairo.
Inkscape. n.d. <https://inkscape.org/fr> [accessed: 5/25/2020].
Janssen, J.M.A. (1973). "Les Listes de Signes Hiéroglyphiques". In *Textes et langages de l'Égypte pharaonique: cent cinquante années de recherches 1822 - 1972. Hommage à Jean-François Champollion* 1. Bibliothèque d'étude 64, Cairo, 57–66.
Kamal, A. (1908). "Stèle de l'an VIII de Ramsès II". *Recueil de travaux relatifs à la philologie et à l'archéologie égyptiennes et assyriennes* 30.3/4: 213–218.
Kilani, M. – Miyagawa, S. – Chapman, D. (2016). *SINUHE*. <https://github.com/somiyagawa/SINUHE> [accessed: 5/22/2020].
Malek, J. (2005). "A Significant Development in Hieroglyphic Word-Processing: The Release of the New InScribe 2004 for Microsoft® Windows", *Discussions in Egyptology* 61: 61–66.
Der Manuelian, P. (1994). *Living in the past: studies in archaism of the Egyptian twenty-sixth dynasty*. London.
Meeks, D. (2004). *Les architraves du temple d'Esna: paléographie*. Paléographie hiéroglyphique 1, Cairo.
Meeks, D. (2006). *Mythes et légendes du Delta d'après le papyrus Brooklyn 47.218.84*. Mémoires publiés par les membres de l'Institut français d'archéologie orientale 125, Cairo.
Meeks, D. (2007). "La paléographie hiéroglyphique : une discipline nouvelle". *Égypte, Afrique & Orient* 46: 3–14.
Meeks, D. (2010). "De quelques 'insectes' égyptiens: entre lexique et paléographie". In Hawass, Z. A. – der Manuelian, P. –Hussein, R.B. (eds), *Perspectives on ancient Egypt: studies in honor of Edward Brovarski*, Cairo: 273–304.
Meeks, D. (2013). "Dictionnaire hiéroglyphique, inventaire des hiéroglyphes et Unicode". *Document numérique* 16.3: 31–44.
Montfort, R. (2000). *Les fontes hiéroglyphiques*. <http://raymond.monfort.free.fr/1.html> [accessed: 5/22/2020].
Nederhof, M.-J. (2002). "A Revised Encoding Scheme for Hieroglyphic". In *Proceedings of the 14th Table Ronde Informatique et Égyptologie. Pisa*. <https://mjn.host.cs.st-andrews.ac.uk/publications/2002e.pdf> [accessed: 5/22/2020].
Nederhof, M.-J. (2013). "The Manuel de Codage encoding of hieroglyphs impedes development of corpora", In Polis, S. – Winand – J. Gillen, T. (eds.), *Texts, Languages & Information Technology in Egyptology. Selected papers from the meeting of the Computer Working Group of the International Association of Egyptologists (Informatique & Égyptologie), Liège, 6–8 July 2010*, Liège: 103–110.
Nederhof, M.-J. – Field, A.M. (2015). *RES in JavaScript*. <https://mjn.host.cs.st-andrews.ac.uk/egyptian/res/js> [accessed: 5/22/2020].

Polis, S. – Rosmorduc, S. (2013). “Réviser le codage de l’égyptien ancien: vers un répertoire partagé des signes hiéroglyphiques”. *Document numérique* 16.3: 45–67.

Richmond, B. (2004). *InScribe 2004*. Saqqara Technology.

Richmond, B. (2016). “Hieroglyphs Everywhere Project”. *Hieroglyphs Everywhere Project*. <http://hieroglyphseverywhere.blogspot.com> [accessed: 5/22/2020].

Richmond, B. (2017) 2019. *HieroglyphsEverywhere/UMdC*. <https://github.com/HieroglyphsEverywhere/UMdC> [accessed: 5/22/2020].

Rosmorduc, S. (2003). “The Tksesh database and information exchange in the Egyptological community”. In Hafemann, I. (ed.), *Wege zu einem digitalen Corpus ägyptischer Texte: Akten der Tagung „Datenbanken im Verbund“ (Berlin, 30. September - 2. Oktober 1999)*. Berlin: 143–154.

Rosmorduc, S. (2007a). “Creating Complex Drawings with JSesh and Inkscape | JSesh”. *JSesh*. <https://jsesh.qenherkhopeshef.org/fr/tutorials/facsimile> [accessed: 5/22/2020].

Rosmorduc, S. (2007b. *JSesh*. <https://jsesh.qenherkhopeshef.org> [accessed: 5/22/2020].

Rosmorduc, S. (2008. “Jsesh Documentation, Extending the Sign List”. *JSesh Documentation*. <http://jseshdoc.qenherkhopeshef.org/doku.php/doc:en:importingglyphs> [accessed: 5/22/2020].

Rosmorduc, S. (2009a. ‘Glyph Creation 1: Creating a New Glyph with Inkscape’. *JSesh*. <https://jsesh.qenherkhopeshef.org/fr/tutorials/glyphs1> [accessed: 5/22/2020].

Rosmorduc, S. (2009b. ‘Glyph Creation 2: Combining Existing Signs into a New One’. *JSesh*. <https://jsesh.qenherkhopeshef.org/fr/tutorials/glyphs2> [accessed: 5/22/2020].

Rosmorduc, S. (2020) *Rosmord/JavaScriptSesh. JavaScript*. <https://github.com/rosmord/JavaScriptSesh> [accessed: 5/22/2020].

Rosmorduc, S. – Polis, S. (2015). “The hieroglyphic sign functions: suggestions for a revised taxonomy”. In Amstutz, H. – Dorn, A. – Müller, M. – Ronsdorf, M. – Uljas, S. (eds), *Fuzzy boundaries: Festschrift für Antonio Loprieno* 1, Hamburg: 149–174.

Schenkel, W. (1972). “Neue linguistische Methoden und arbeitstechnische Verfahren in der Erschließung der ägyptischen Grammatik”, In *Textes et langages de l’Égypte pharaonique: cent cinquante années de recherches 1822 - 1972. Hommage à Jean-François Champollion* 1, Cairo: 167–176.

Schenkel, W. (1974). “Gesichtspunkte für die Neugestaltung der Hieroglyphenliste”. *GM* 14: 31–45.

Schenkel, W. (1984a). “Weiteres zur Transkription des Hieroglyphisch-Ägyptischen”. *GM* 77: 81–83.

Schenkel, W. (1984b). “Weiteres zur Transkription des Hieroglyphisch-Ägyptischen II”. *GM* 79: 43–51.

Schenkel, W. (1984c). "Weiteres zur Transkription des Hieroglyphisch-Ägyptischen III". *GM* 82: 51–60.

Stief, N. (1985). "Hieroglyphen, Koptisch, Umschrift, u.a.: Ein Textausgabesystem". *GM* 86: 37–44.

Stief, N. (1988). "Weitere Möglichkeiten bei der Hieroglyphenausgabe via Computer". In *Informatique et Égyptologie* 5: 46–51.

Subotic, I. – Lapp, G. (2008). *IGlyph*. <http://subotic.github.io/iGlyph> [accessed: 5/22/2020].

Thot Sign List, <http://thotsignlist.org> [accessed: 5/22/2020].

Winand, J. – Polis, S. – Rosmorduc, S. (2015). "Ramses: an annotated corpus of Late Egyptian". In Kousoulis, P. – Lazaridis, N. (eds), *Proceedings of the Tenth International Congress of Egyptologists: University of the Aegean, Rhodes. 22–29 May 2008*, Vol. 2, Leuven: 1513–1521.

Winand, J. – Polis, S. – Rosmorduc, S. (2015) *Ramses Online*. <http://ramses.ulg.ac.be/site/aboutRamses> [accessed: 5/20/2020].