Authentic Arabic: a case study

Right-to-Left Font Structure, Font Design and Typography

Thomas Milo, DecoType, Amsterdam, the Netherlands

As the most elaborate of all the RTL scripts, Arabic represents an unusual challenge. This treatise discusses the interaction between text encoding and font technology against the background of an understanding of the structure of Arabic script.

The oldest known Arabic scripts consist of a single layer of ambivalent or multivalent letters. Like the vowel signs, the additional script layer to disambiguate letters emerged gradually and with regional variations. Understanding how Arabic script emerged and evolved provides clues both to encoding and to rendering issues of Arabic and related scripts. Here is an underestimated and very powerful aspect of Arabic script: it makes it fundamentally *archigraphemic* in structure.

An archigrapheme occurs when the distinction between two or more letters is neutralized. The archigrapheme is a graphic unit that consists of the shared features of neutralized letters, minus the features that differentiate them. In the archigraphemic analysis of Arabic script, vowels and dots are different layers of additional, variable features.

These issues are relevant in the context of Unicode-related discussions, because:

1. The Unicode Standard today assumes limited, grapheme-based (i.e., explicit) use of Arabic script;

2. Grapheme and ligature-based legacy technologies have lead to misconceptions and inconsistencies both in the code structure and the visual rendering of languages written in Arabic script;

3. Archigraphemic encoding of scripts like Arabic is the key to sophisticated operations on computerized Arabic text corpora and addresses apparent regional and diachronic variation;

4. The archigraphemic approach is fundamental to proper Arabic font technology;

5. Archigraphemic font technology creates optimal conditions for contemporary Arabic font design;

6. Operating systems need to specify the open architecture required to facilitate the optimal technology for rendering a given script, to give the user access to existing and future expert font rendering and layout mechanisms.

Phoneme vs. Grapheme

Script terminology is partly inspired by and derived from the linguistic doctrine of *phonology*. Linguistics defines a *phoneme* not as sound, but as *a bundle of distinctive features* in the context of a given language. By analogy the grapheme should not be considered a visible sign, but *a bundle of distinctive features* in the context of a given script.

The linguistic relevance of a feature is established by isolating it from semantically different minimal word pairs. This can be represented as follows:

feature → / phoneme ↓	labial	dental	nasal	word
/m/	+	-	+	*"map"*
/n/	-	+	+	*"nap"*
The phoneme /m/ *differs from the phoneme* /n/ *in the features of dentality and labiality in a contrastive opposition*				

In the structure of Arabic writing system there is a close analogy:

feature → / grapheme ↓	single dot below	double dot above	tooth	letter
ب	+	-	+	*beh*
ت	-	+	+	*teh*
The Arabic letter beh ب *differs from the Arabic letter the Arabic letter* teh ت *in the features of* single dot below *and* double dot above *in a contrastive opposition.*				

The example below shows 5 different letters ("*beh*-class" graphemes). The skeletons are identical; only the attachments are different.

ب ت ث پ ٹ

Allophone vs. Allograph

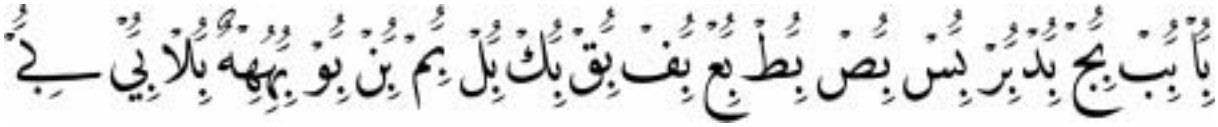
The physical realization of the sound of a phoneme falls outside the scope of linguistics proper, it is to the discipline of *phonetics* to analyze and describe it. The sound of a phoneme has many subtle context-determined variations that do not affect the linguistic meaning and therefore escape the native speaker: the *allophones*. For the phoneme /n/ they can be illustrated as follows:

Phoneme	Contextual positions of Allophones
/n/	**[n- | -n- | -n]**
A phoneme can occur in initial (n-), *medial* (-n-) *and final* (-n) *position. In each position there are variations of the actual sound caused by modulation to the surrounding sounds: the allophones.*	

There are also subtle variations in shape that escape the naïve reader, so here again is an analogy:

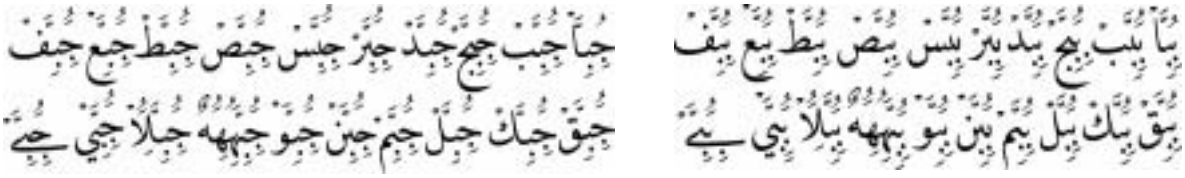
Grapheme	Contextual positions of Allographs
Arabic letter BEH	[بـ | بـ | ب]
The contextual positions initial, middle and final are the allographic categories. The actual allograph is the result of interaction with the allographs of any adjacent graphemes.	

Most simplified fonts have only *one* glyph of each position to cover allographs. Legacy typography incorporates a small, random selection of additional allographs in "nostalgic" *ligatures*.



A selection of beh*-class allographs in initial position. The theme letter (*beh *in this example) is surrounded by a double set of vowels and followed by a parade of final forms. (generated by DecoType Arabic Calligraphic Engine)*

Interestingly, there is no traditional pattern for listing the middle forms[1]. In the example below the first list is expanded to show a small selection of *beh*-class allographs in middle position:



Two examples of beh*-class allographs in middle position in a quasi traditional presentation showing all final forms. The block on the right hand side has initial* beh*, on the left initial* jeem*. (generated by DecoType ACE)*

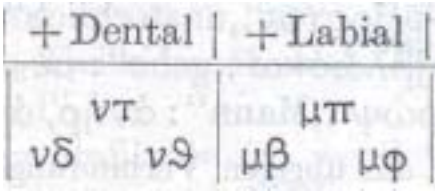
The visual realization of graphemes falls outside the scope of Unicode. It should be left to the field of expert technologies to handle the allographic level of the Arabic script in order to create the right conditions for professional type design. Operating systems should be designed to facilitate such developments.

---

[1] For modern tables see *Writing Arabic, a practical introduction to Ruq'ah script*, T.F. Mitchell, Oxford 1953. These tables cover only groups of two and three base letters, i.e., a fraction of the total required.

Archiphoneme vs. Archigrapheme

In the sound system of Classical Greek – and that of most languages – there are no minimal word pairs with the opposition /n/:/m/ when these phonemes are followed by /b/ or /d/. In fact, while /mb/ and /nd/ exist, /md/ and /nb/ are ruled out[2].



combinations of /n/ *or* /m/ *with following dental or labial consonant (classical Greek)*

The functional difference between these phonemes disappears and results in a new phenomenon: the *archiphoneme* (symbolized with a capital letter of one of the neutralized phonemes). This can be represented as follows:

feature → / archiphoneme ↓	Labial	dental	nasal	example
/N/	(+)	(-)	+	em*bryonic*
/N/	(-)	(+)	+	en*demic*
The archiphoneme is a phonological concept that consists of the shared features of neutralized phonemes, minus the features that differentiate them.				

In the large corpus of historical Arabic texts the distinctive features were used sparingly, leading to the following analogy:

feature → / archigrapheme ↓	single dot below	double dot above	tooth	example
ب	(+)	(-)	+	*beh? teh?*
ب	(-)	(+)	+	*teh? beh?*
The archigrapheme is a graphic unit that consists of the shared features of neutralized letters, minus the features that differentiate them. In this analysis of Arabic script, vowels and dots are different layers of additional, variable features.				

Many important historical texts are only known in "defective", i.e., archigraphemic script. Even where an old manuscript in *scriptio plena* exists, it often derives from an archigraphemic original, which implies that the layers of secondary script are later interpretations. These additional layers – both vowels *and* dots – are the ones most vulnerable to scribal errors. In fact these documents are, strictly speaking, only truly original on the archigraphemic level. Academic analysis of computerized versions of such corpora is frustrated by the present graphemic structure of Arabic in Unicode. Alternative archigraphemic encoding with reversible compatibility would be ideal.

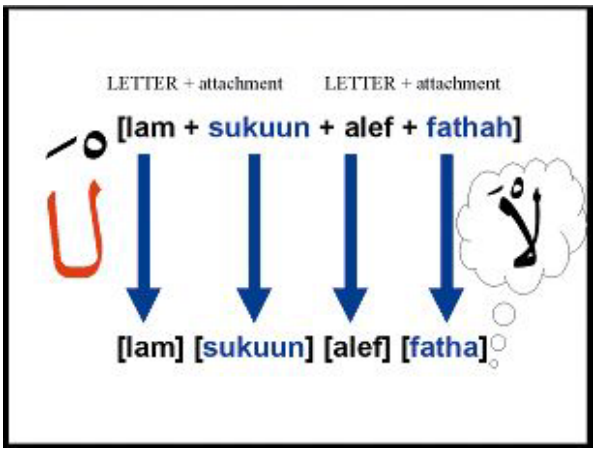
[2] W.Brandenstein, *Griechische Sprachwissenchaft I, Einleitung, Lautsystem, Etymologie*, Berlin 1954, §50.

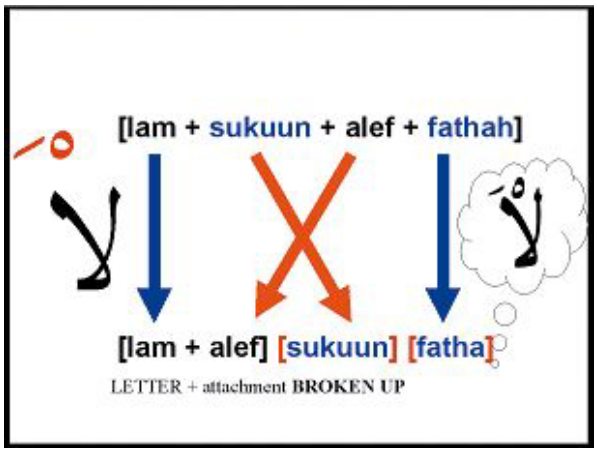
How font structure inadvertently affects data structure

The adaptation of Arabic script to the typewriter was the ultimate step in the process of simplifying its morphology. It is a classic case where quality is sacrificed to minimize design effort. To freeze this fluid writing system on just forty-four keys, it was stripped of all ligatures – but one. *Alef* follows *lam* in such a way that this particular sequence of two letters cannot be dissected.

As a result, *one* key represents a curious ligature. With a bit of imagination the *lam-alef* key is more than a permanent carry-over from typography. It can be seen as a metaphor of the resilience of the Islamic writing culture against mechanical maltreatment, since in Arabic it means: “No”!

With the evolution of the keyboard into a data-entry tool, old typing habits create an interesting problem. In order to be linguistically consistent with the consonant-plus-vowel structure, attachments should follow the governing letter directly.





However, the widely used table-driven Arabic script today does not generate ligatures of letter groups when they are separated by attachments. Therefore in the case of *lam-alef*, to achieve the correct visual result the writer is forced to rearrange the data in a manner that is graphemically incorrect – or just forget about the diacritics. This is just one example of how defective font technology has created chaos in the world of computerized Arabic.

The other way around: data structure inadvertently affects font structure

In post-1920's Arabic typography a related problem with attachments can be observed. The typesetter has no means to insert attachments to letters that are part of a ligature, so he replaces them with individual "typewriter" glyphs in order to fit in the attachments, often "camouflaged" by an extra carrier line. Mechanically he has no other option but to sacrifice the typographically correct ligatures, and he assumes this is æsthetically acceptable.

English metal font (1950s) shows crude insertion of vowels, sacrificing a ligature (shown separately as well)

Arabic Computer typefaces accommodate for vowels by sacrificing ligatures (dimmed in the background)

In computing the erroneous classification of ligatures as *optional* leads to shaping algorithms that allow falling back on *typewriting* when inserting vowels, with comical effects as described above. In adequately designed Arabic font technology the attachments would not influence the main script.

The images above were generated with the DecoType ACE technology. It allows the attaching of distinctive dots and vowel markers without affecting the skeleton text.

At this point it must be stressed, that this type of defect hampers *all* sophisticated fonts that were conceived to function with ligatures and full vocalization. Technology has wreaked havoc under Arabic type instead of facilitating it. Even innovative design, simplification and restructuring Arabic script as a conscious cultural choice, is caught in a straightjacket of technical shortcomings. Operating systems need to provide the open architecture for expert systems to deal with such issues.

Legacy Arabic typesetting technology is mostly grapheme-based

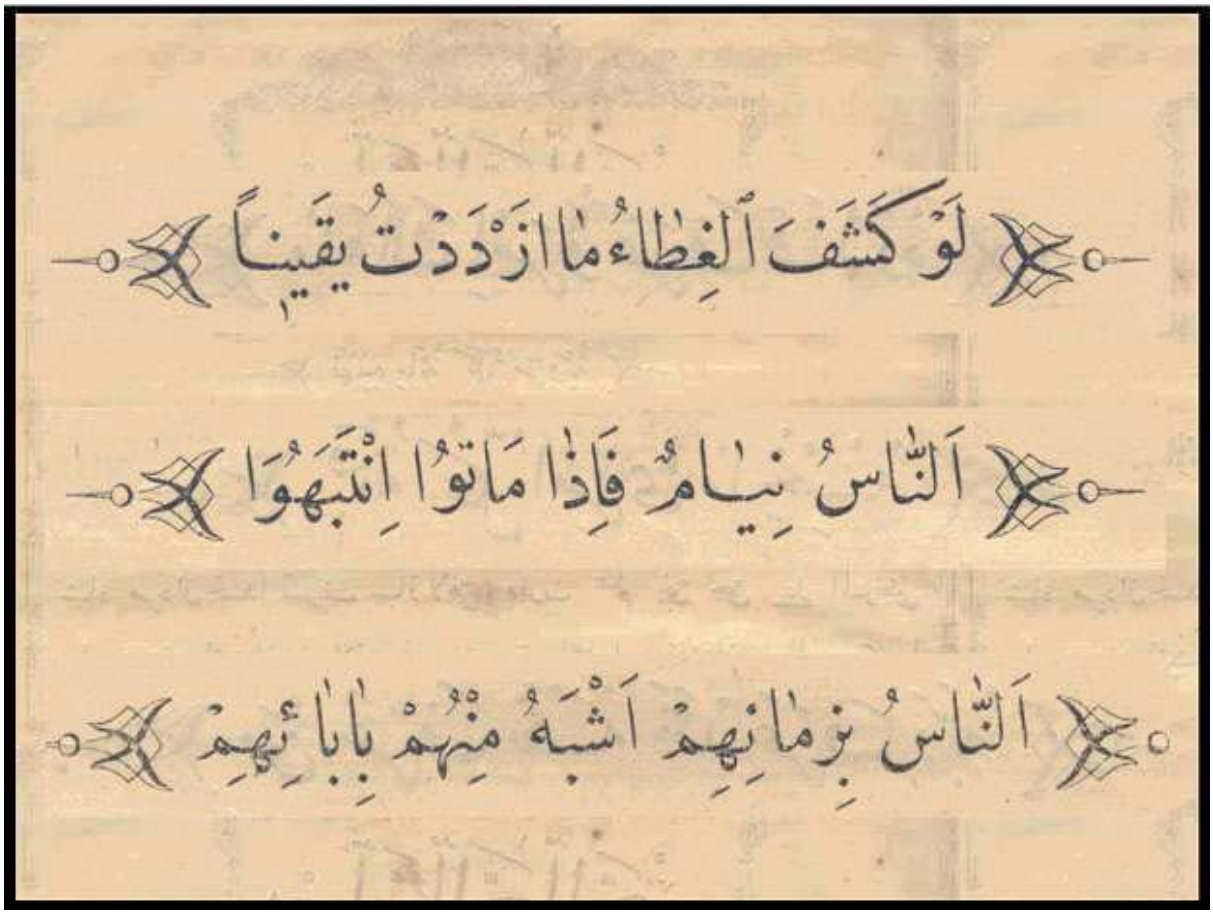
Why vowels should affect the structure of the graphic skeleton can be understood by analysing the mock-up below. It shows *grapheme*-based Arabic typographic technology: it treats dot-attachments as an integral part of the letter[3]. Interestingly, remnants of an earlier *archigrapheme*-based technology can be seen in this design as well: the ligature on the right allows for dots to be attached separately.

The two main castings contain built-in attachments (one has a single and the other a triple upper dot). The ligature on the right is designed to allow an attachment to be packed under it, placing attachments over or under its extending pointed shape, e.g., yeh-khah-*(when two dots are added as in the example. The letter block on the left represents two attachments united in a ligature. This particular metal construction positions these attachments significantly away from the last letter. However the natural place for them is above the last letter:* lam *with* shadda *and* dhamma*. Because of the graphemic structure of the font, it cannot deal correctly with Arabic script.*

[3] The metal elements were once part of an Arabic font produced by the Dutch Tetterode company, from which I managed to salvage a few types. The example therefore has to be random, on the basis of the available forms. I mirrored it for the sake of comparison.

The Mother of Arabic Typography: based on archigraphemes

Ottoman Naskh (spelled *Nesih* in modern Turkish) definitely guided all Middle Eastern efforts in typography. In the 1860's an Armenian typographer by the name of Ohannis Mühendisoğlu[4], an Ottoman-Turkish citizen, finally succeeded in reproducing this script in a way that met the demanding standards of the Islamic Calligraphic tradition[5]. His sublime approach to typographic solutions appears to have been fully *archigrapheme*-based.

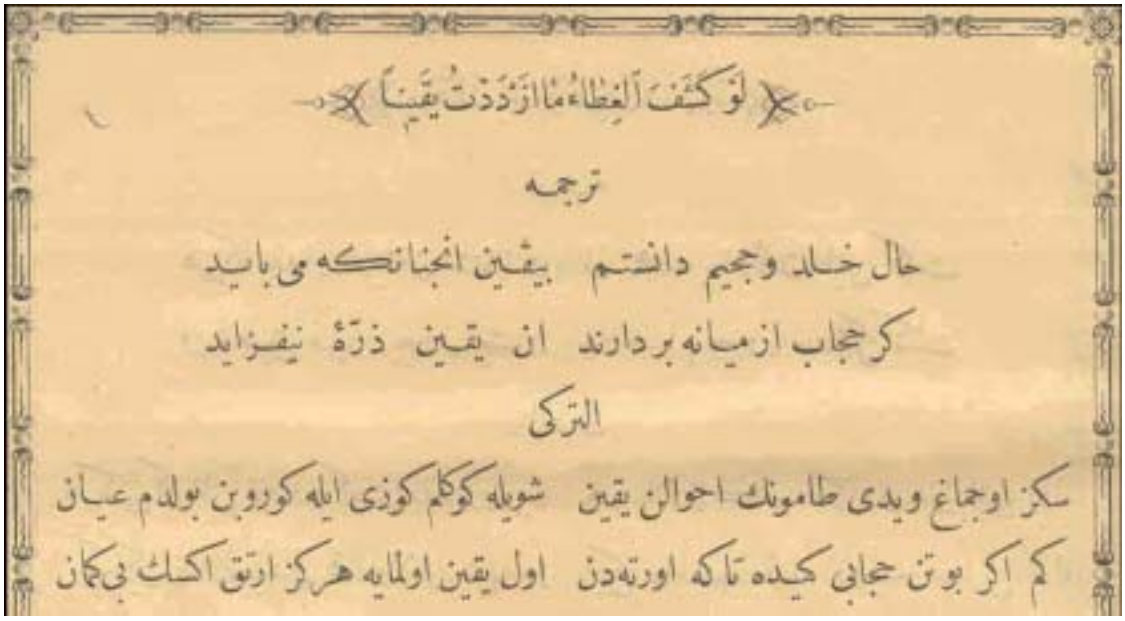


Arabic phrases, typeset in metal, showing integral coverage of the Arabic script morphology and correct placement of the attachments.

[4] Mühendisoğlu is the Turkish version of his name with the literal meaning of *son of the land surveyor* (or *civil engineer*). His name is also encountered in Ottoman-Persian (Mühendiszade) and Armenian forms (Mühendisyan).

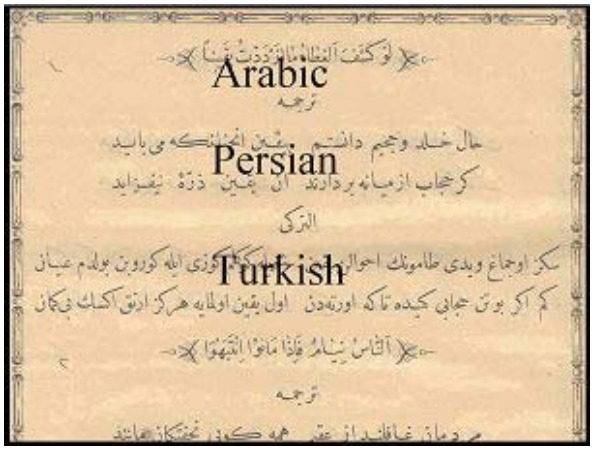
[5] An *arzuhal* or petition to the Sultan of the Ottoman Empire dated 1283 a.h. (1865 a.d.) came in my possession in 1983. In it Mühendisoğlu announces that for the first time a valid Naskh typeface was designed. He describes how he used the handwriting of the late *şeyh-ül-hattatin* (leading calligrapher) Mustafa İzzet Efendi to accomplish this historical achievement. Uğur Derman, the leading specialist of Ottoman calligraphy, reports corroborating evidence to the Turkish Librarians' Association (Türk Kütüphaneciler Derneği Basım ve Yayıncılığımızın 250. yılı Bilimsel Toplantısı. 10-11 Aralık 1979, Ankara; Bildiriler, Ankara, 1980, vii, 174 pp: M. Uğur Derman, *Yazı sanatinin eski matbaacılığımıza akisleri*. pp 97-118). In this essay he also mentions the advanced Ta`liq typefaces designed by him as early in the 1840's. In spring 2001, I discovered two of only three books ever printed in Mühendisoğlu's Ta`liq.

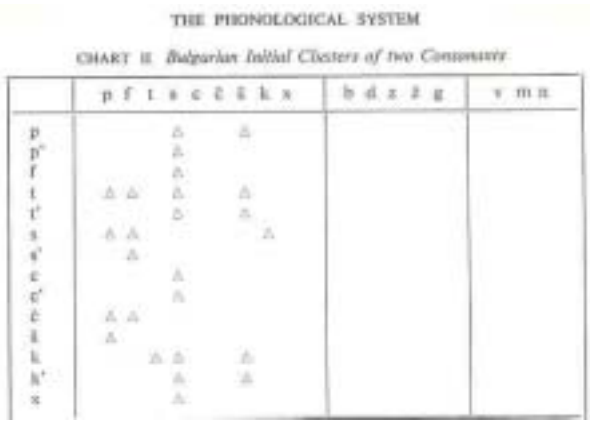
The previous illustration shows close-ups from brilliant typesetting by Mühendisoğlu in the Yeni Hurufat[6] in the three main languages of the Ottoman Islamic world: Arabic, Persian and Turkish.



Each Arabic phrase is followed by a Persian translation and an (Ottoman-) Turkish explanation.

Any given language has its own distinct pattern of sound combinations, so when an Arabic font is used for two extra, totally different languages, it exposes more of the structure behind glyph combining and ligature use than any monolingual typography ever could.



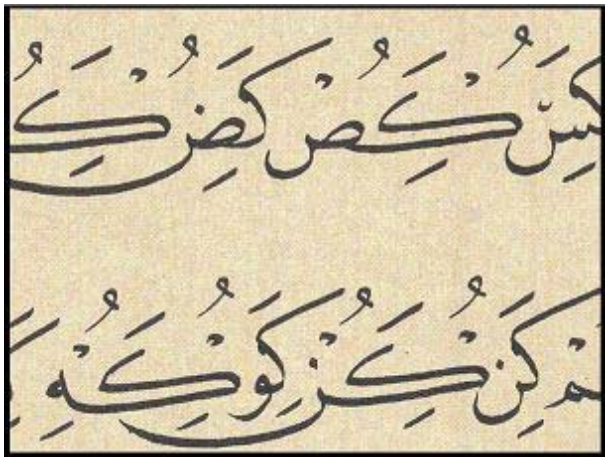


Languages do not just differ in respect of phoneme inventory; they also differ in phoneme usage. The table on the right shows that "permissible" combinations of a certain language[7] produce a unique "fingerprint". Another example can be seen in English, which happens to differ from Greek in the distribution pattern of the cluster /ps/: in the borrowing ellipsis, */ps/ can be matched with English phonemes, but not in the word* psyche. *If English were spelled morpho-phonologically – like Arabic – we would never see the letter group* ps- *in intial position: sykee.*

[6] According to the colophon it was printed in Istanbul 1869-70 a.d. In spring 2001, I made the sensational chance discovery of this rare book printed in the exact same Nesih typeface as the petition of 1865 described in footnote 5

[7] Bulgarian consonant clusters at the beginning of a word, taken from H.I. Aronson, *Bulgarian Inflexional Morphology*, the Hague 1968.

Rather then attempting to create his own version of Arabic script, Mühendisoğlu (1810-1891) modelled his typography on the handwriting of *Kazi Asker* (Supreme Judge) Mustafa İzzet Efendi (1801-1876), ranking among the viziers or ministers of the Ottoman State. İzzet Efendi, i.e. Lord İzzet, was a man of great authority. He was a composer of *Ottoman* classical music and the leading calligrapher of his times. Among the many calligraphic and musical compositions of his hand are the large tableaux inside the historical Aya Sofya Mosque in the very heart of Istanbul, capital of the Ottoman Empire. This lofty man certainly was not the type to be involved in *type design* and it can be ruled out that the craftsman and the calligrapher ever met.



"Meşk mürakka'i" or writing exercise by İzzet Efendi elaborating the shapes of the letter kaf in the Nesih (Naskh) style, the artistic equivalent of the étude in the Western musical tradition).

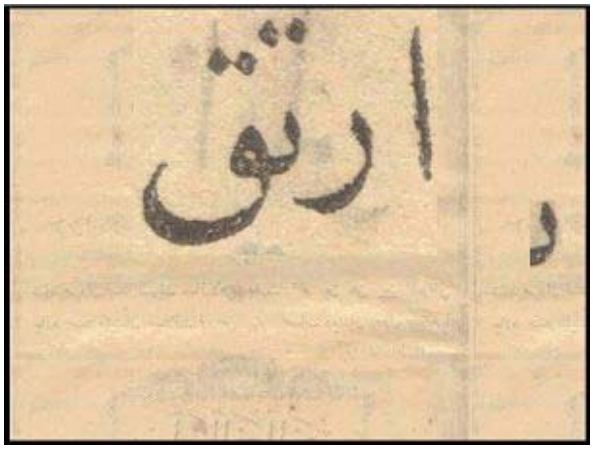
The interior of the Aya Sofya Mosque in Istanbul, showing the circular tableaux with the names of 'Othman and Abu Bakr, two of the Caliphs (successors) of the Prophet. All are calligraphies by İzzet Efendi

This adaptation by Mühendisoğlu of İzzet Efendi's calligraphy is the starting point of all later Arabic Naskh typefaces. The font was graphically extremely sophisticated as it was designed to follow all the *allographic* rules of Naskh in the tradition of the copyists, the professional book producers before the advent of typography. The essential feature is that it deals with both dot and vowel attachments as separate horizontal layers above and below the main script. In other words, the design was *archigraphemic*. However, the seeds of decay are already present in this 40-page booklet. The initial pages immaculately implement every rule with the correct glyph. As the page numbers go up, so go the number of calligraphic typos: the zenith of Arabic typography stands at the beginning of erosion rather than evolution of Naskh script[8]. This is an extremely good design, but it should have had a computer program to support it!

[8] Naskh, Thulth, (Naskh-i-) Taliq and Ruq'ah scripts are governed by well-organised and logical morphological rules, the knowledge of which is rare among typographers and type designers today. Even in the Nesih typesetting of his first petition in 1865 Mühendisoğlu makes two composition errors.

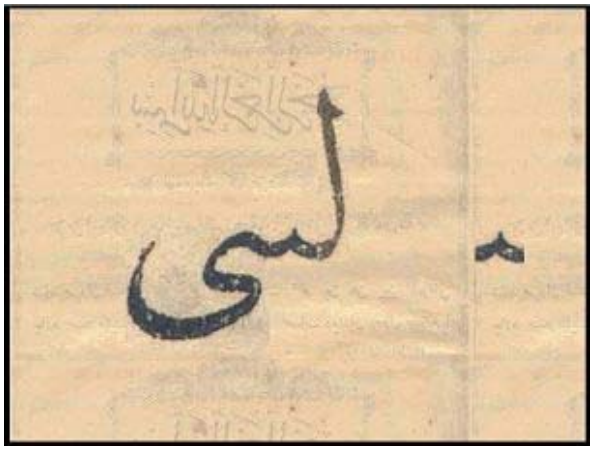
Allographic decay – even a sublime design is pointless without proper skills

Contemporary fonts all show considerable simplification that lead to a drastic change in appearance of the Arabic script. Below in a series of steps, using a minimal selection of glyphs of the Yeni Hurufat, the road to the currently widespread font type is reconstructed.

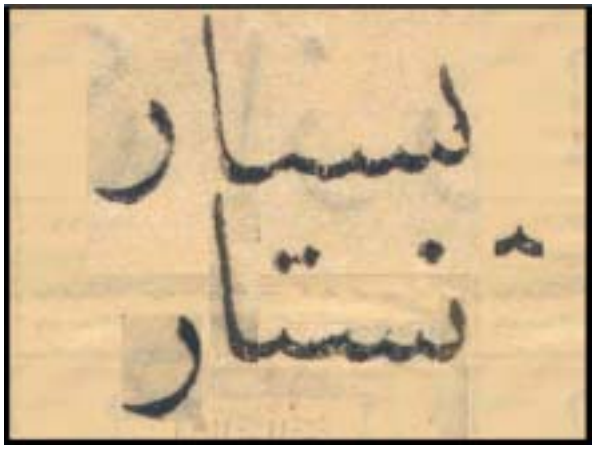


The beh-*class had a special initial curve assimilated to the archigrapheme* qaf/feh *and* waw *is today is generalised into the initial form in* all *contexts.*

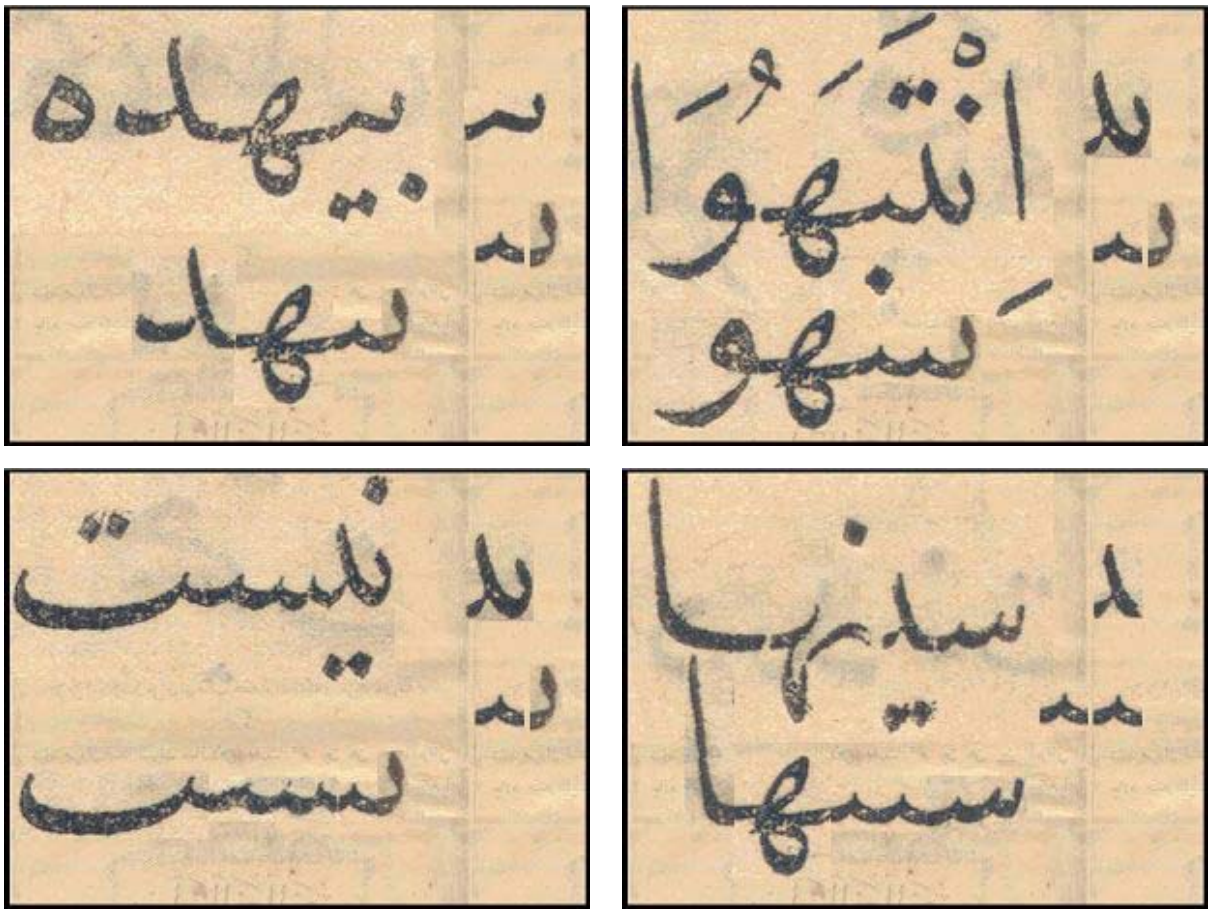
Even the best-designed font becomes pedestrian when generic forms are used: theh-reh *as designed (top) with under it when composed with out-of-context elements.*



A specialised beh-*class allograph (occurs only when preceding certain inverted* beh-*class allographs). In simplified typography it used as a generic middle form.*

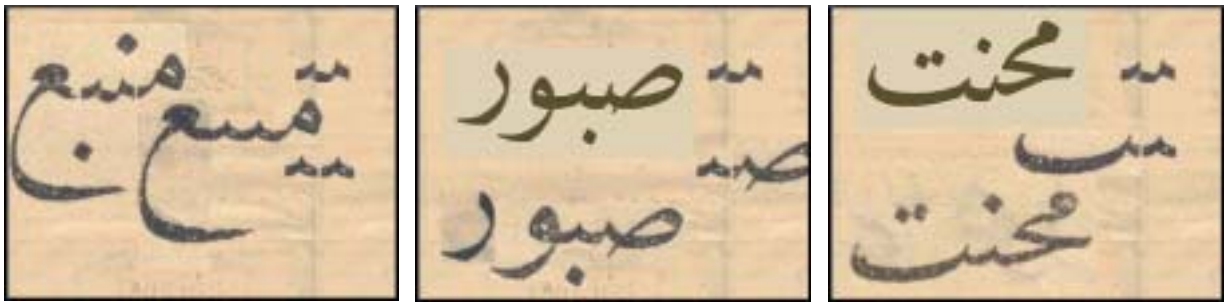


A curved beh-*class allograph in a word composed by Mühendisoğlu as designed (top) and the same word composed with out-of-context- borrowings from the same design. The result: modern typography!*



The combination of new generic initial and middle forms cannot be used without breaking the rules of the traditional allographic system.

An interesting example of lost know-how can be seen when two *beh*-class allographs occur between dissimilar letters. Mühendisoğlu cut a special double-pointed curve ligature. However, this typesetting method must have been very tiring to handle: in this booklet Mühendisoğlu applies the correct form only twice. Another observation can be made: even this sublime Naskh design cannot implement this rule in all contextual situations, because in a metal font, boundaries are out of necessity on the graphemic level. The underlying calligraphic mechanism operates on the level of pen strokes regardless the graphemic status of the larger unit they build.



The illustrations above show A special ligature for -beh-beh- *in middle position. Modern fonts glyph sets do not include this feature. The next two images show how the double curve ligature crosses grapheme boundaries: one part belongs to the (archi-) grapheme* beh *and the other part belongs to the following or preceding (archi-) grapheme.*

Using the Archigraphemic concept in font technology

Operating systems provide new font technologies in the wake of the emerging Unicode Standard. In the case of Arabic, they can be put to good use for building grapheme-based fonts for optimal Unicode coverage. However, the Unicode Standard has no provision for archigraphemic Arabic yet, but this fundamental structure of Arabic script can be exploited at least in the design phase: one can build an automated Arabic-specific font tool.

FontLab, the most up-to-date font-designing software, has been adapted in a joint effort by its designers and DecoType, to produce simple OpenType fonts for Adobe InDesign and WindowsXP. This tool also provides an efficient interface to build legacy style ligatures of up to 4 graphemes.

Arabic Base Glyphs

The archigraphemic structure implies that Arabic graphemes share many structural elements. This phenomenon was exploited here to the maximum. In the examples below ("BEH-class" graphemes), the skeletons – archigraphemes – are *identical;* only the attachments – distinctive features – are *different*. This observation forms the basis of the automation process.

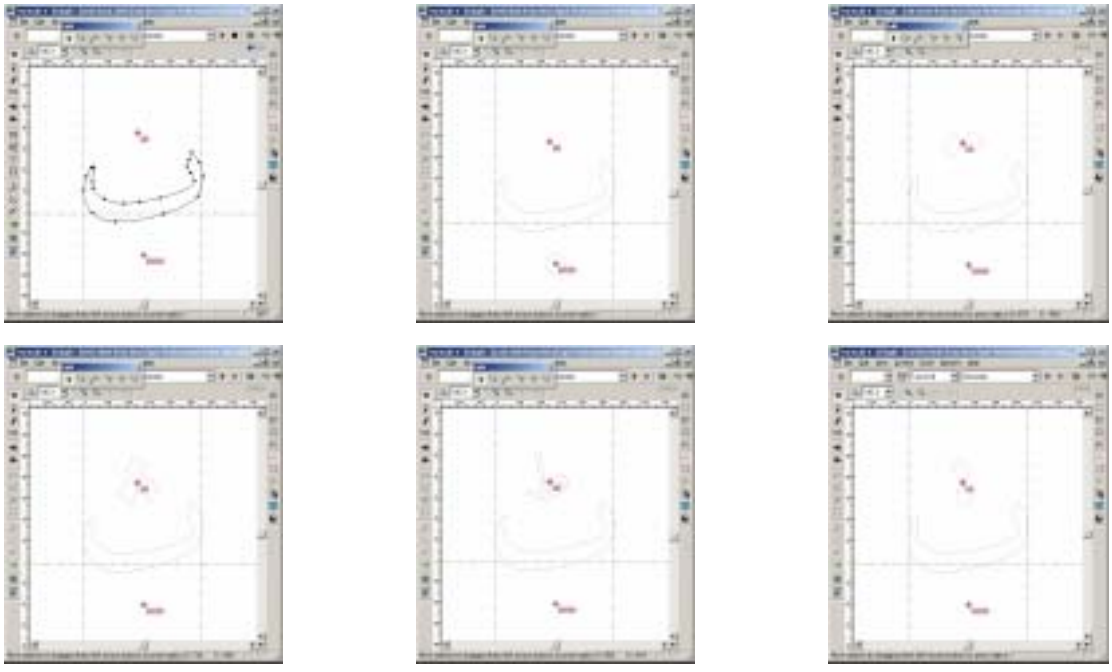
ب ت ث پ ٹ

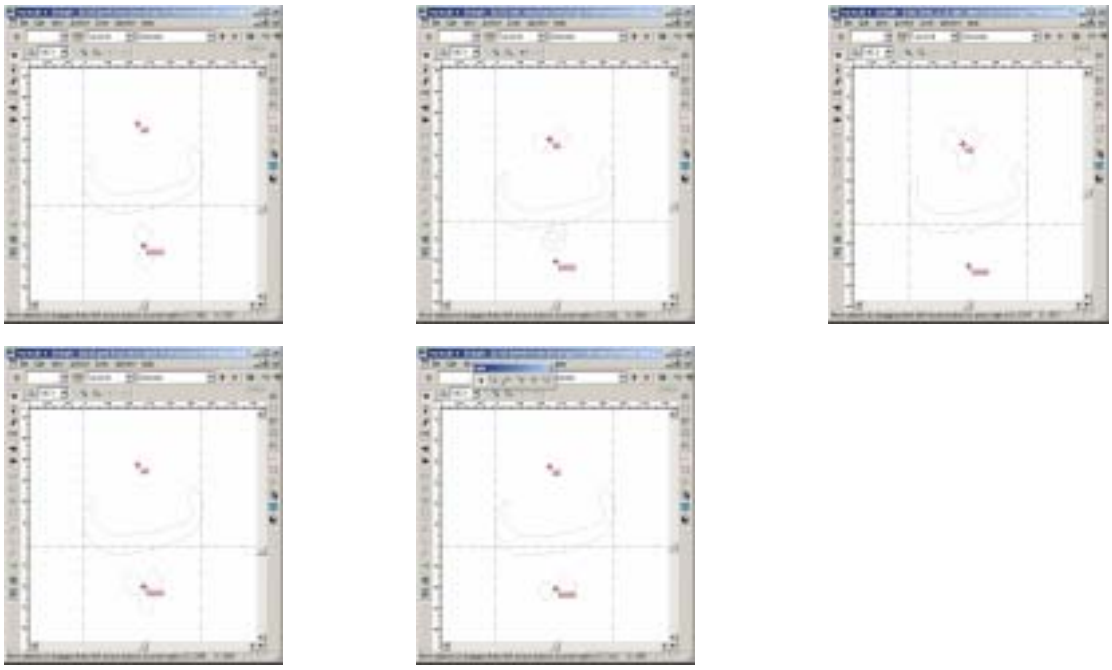
Some letters that share the same sub-letter element

ن ف غ ظ ض ذ ز خ

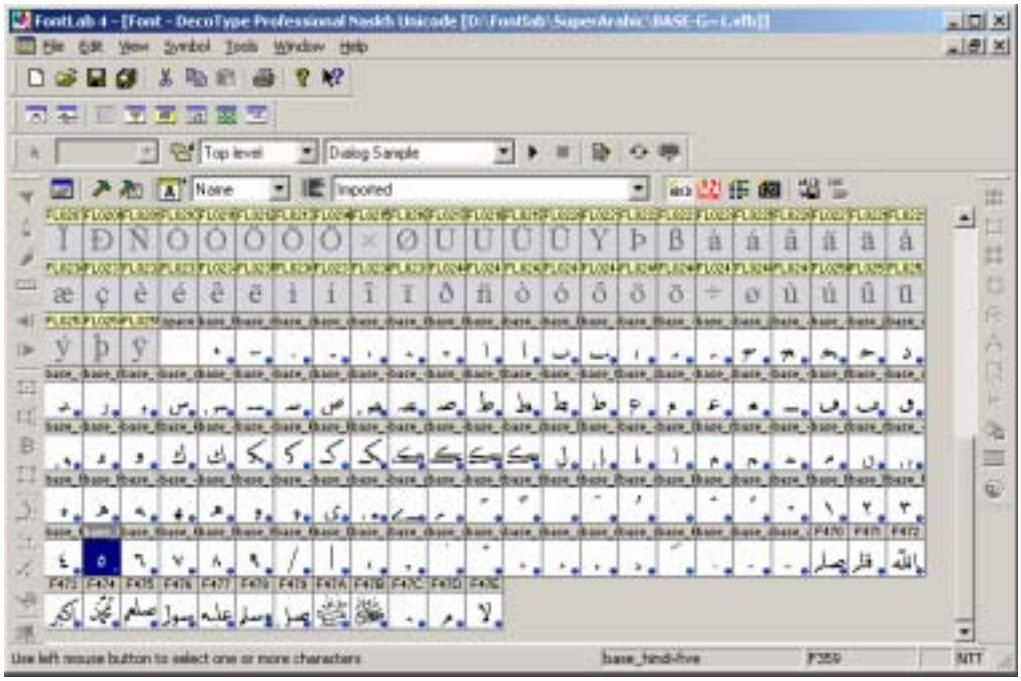
Some letters that share the same attachment

Exploiting the repetitive nature of Arabic writing, only the outline paths of a limited set of sub-letter elements need to be drawn. The printable glyphs are composed by references to these base glyphs.

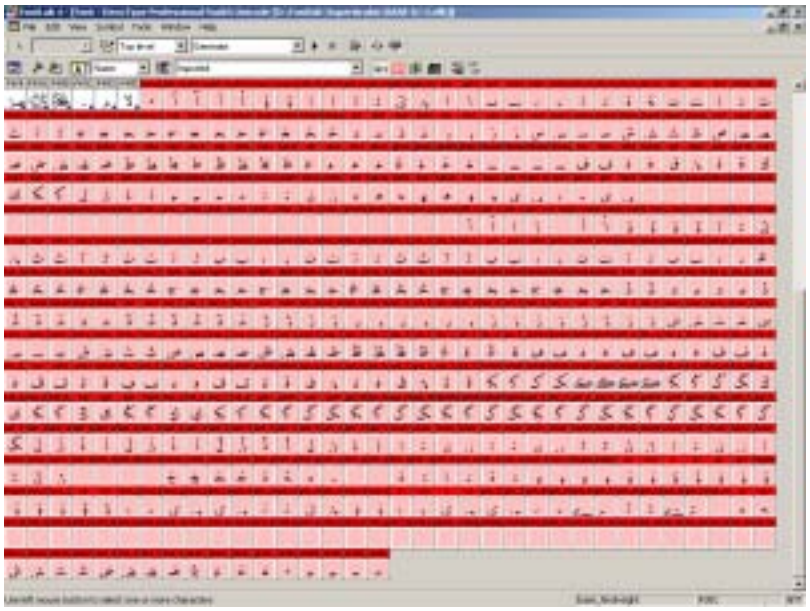




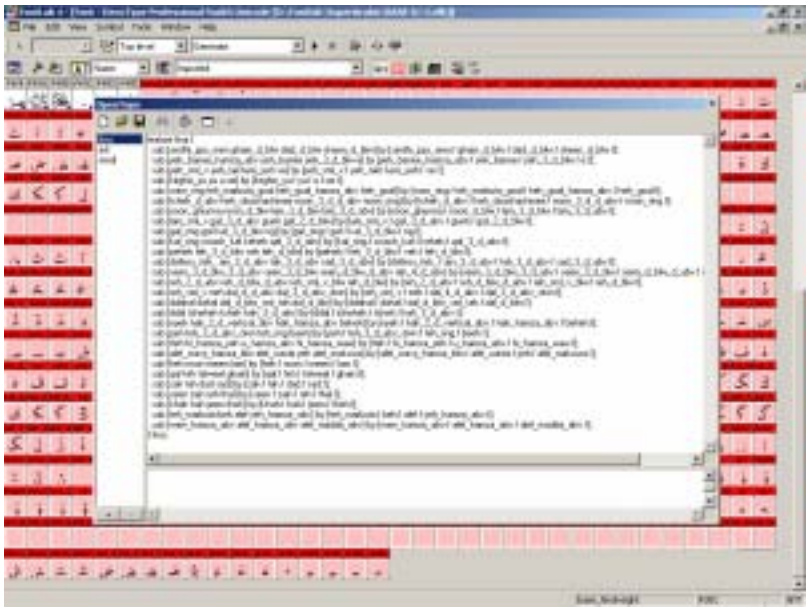
The practical result is an extremely small font, easy to design and to maintain.



From there, tables are used to build a font consisting of references to these base elements. In this process, each grapheme shall be expanded automatically into the full series required for contextual representation:



The font type used for this example (Open Type) needs internal tables for contextual glyph substitution. The new tool generates these tables dynamically.



The method described above shows just one way of benefiting from the archigraphemic structure of Arabic script. The drawback of this table-driven paradigm is that it is archigraphemic only in the design phase.

Conclusion

What is required in Unicode is a variant character-glyph model to handle these issues on the level of the Operating System. But it is not just that. For the end-user it would be a major improvement if Operating Systems in general facilitated the use of the optimal script system for a given script. This means the ability to switch between different Font Rendering and Layout mechanisms. An example of such modularity is Apples Open Font Architecture (OFA).