DOI: 10.69085/linc20261187

John MATHEWSON
(Independent Scholar)

TOWARDS A FULLY CROSS-PLATFORM, UNICODE-COMPLIANT, SOFTWARE ENTRY SYSTEM FOR DIGITISATION OF OLD BULGARIAN TEXTS

Abstract. *This article makes a case for moving from a plethora of ways of digitising Old Bulgarian texts that are incompatible with each other towards a software package that is cross-platform and results in texts that utilise the Unicode standard and are completely cross-compatible. It also describes the programming of just such a software package.*

Keywords: *Old Bulgarian*[1] *texts; digitisation; Unicode; cross-platform software*

Джон Матюсън
(Независим изследовател)

СОФТУЕРЕН ПАКЕТ ЗА ДИГИТАЛИЗИРАНЕ НА БЪЛГАРСКИ ТЕКСТОВЕ

Резюме. *Тази статия обосновава преминаването от множество начини за дигитализиране на старобългрски текстове, които са несъвместими помежду си, към софтуерен пакет, който е междуплатформен и води до текстове, използващи стандарта Unicode и напълно съвместими помежду си. Описва се и програмирането на такъв софтуерен пакет.*

Ключови думи: *старобългарски текстове; дигитализация; Unicode; междуплатформен софтуер*

[1] The author has chosen to us the term Old Bulgarian throughout this paper for the reason that his software is primarily targeted at digitising Old Bulgarian documents. Obviously, this does not preclude the software package’s use for digitising non-Old Bulgarian texts that employ either the Glagolitic or the Old Cyrillic writing systems.

Introduction

Digitisation of texts should be considered in a number of ways, and especially with regard to the linguistic content of texts and how those texts are presented, visually, in their originals as is stated on the Missouri S & T University libraries website:

Depending on the purpose of the collection, different approaches to digitizing text content may be used. In some cases, libraries may only be interested in the information that the text conveys, and the medium of expression is irrelevant. However, in most collections, it is desirable not only to create a digital representation of the information within the text content itself, but also the visual aspects of the text, such as type, formatting, layout, or paper quality. (https://libguides.mst.edu/)

While Missouri S & T University libraries advocate preparing images of texts, the author disagrees with this extremely strongly for several reasons:

1. Images that were prepared of texts in the early 1990s now appear as extremely crude, bitmapped images that, in many cases, are extremely hard to read and/or make out individual characters and diacritic/abbreviational marks.

2. These images are, obviously neither editable texts, nor are they much good for text extraction, OCR (Optical Character Recognition) always opting for “the lowest common denominator” resulting in characters from other writing systems ‘mysteriously’ appearing in digitised texts.

If texts are to be adequate, not just for reading, but for academic access and research, they need to be capable of being:

3. Edited.

4. Commented on in a variety of ways (text insertion, supralinear annotation, etc.).

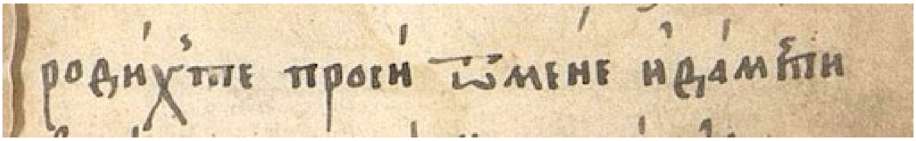


Figure 1. Example of original document

This may be extremely attractive, but what it is not is editable like this:

родн́хѕт•те

Figure 2. Example of digitised version

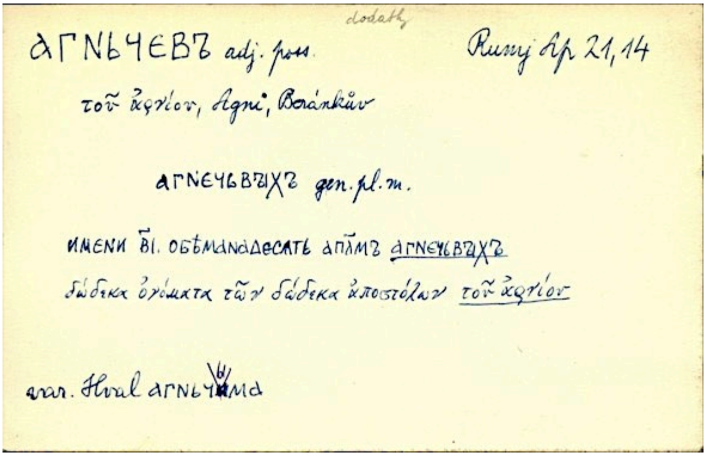
5. Future-proofed (as far as is feasible).

This problem was presented in literature by Iain Banks (Banks 1992) where one of the protagonists has left a body of writing on extremely antiquated floppy disks which it takes an enormous effort to retrieve.

The speed of digital obsolescence is, arguably, accelerating rather than slowing down, as data that was stored on CD-ROM and DVD discs are already becoming increasingly difficult to retrieve as computers are, increasingly, being made without the capability of handling them, and the software to access their contents is only usable of machines that are no longer being maintained.

David Birnbaum discussed “four goals that should not be controversial, and that should govern the way Slavic philologists use electronic texts” in 1995. Stating that these were: “MULTIPLE USE, STRUCTURE, PORTABILITY, and PRESERVATION.” (Birnbaum 1995)

This is completely unacceptable in 2025:



dodatky

АГНЬЧЕВЪ adj. poss. Rumj Ap 21,14

τοῦ ἀρνίου, Agni, Beránkův

АГНЕЧЬВЫХЪ gen. pl. m.

ИМЕНИ ВІ. ОБѢМАНАДЕСАТЬ АП҃ЛМЪ АГНЕЧЬВЫХЪ

δώδεκα ὀνόματα τῶν δώδεκα ἀποστόλων τοῦ ἀρνίου

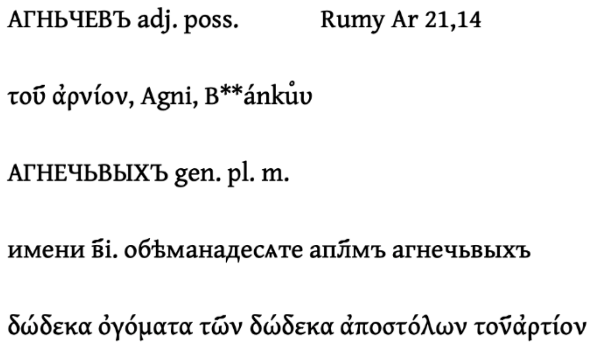
var. Hval АГНЬЧѦМА

Figure 3. Library card. (http://gorazd.org/kartoteka/?envLang=en)

As the font used for the author’s software contains all the Unicode characters for Ancient Greek, the author has introduced an interface for Greek input. However, while the author can see some justification for including Greek input and any additional diacritics used in Greek, the inclusion of the complete Precomposed polytonic Greek range (1F00–1FFF) does not seem justified for 2 reasons:

1. This is meant to be software package aimed at encoding primarily Old Bulgarian texts, and not Greek texts as such.

2. If polytonic Greek symbols are required, on occasion, they can be easily composed with diacritics.



АГНЬЧЕВЪ adj. poss. Rumy Ar 21,14

τοῦ ἀρνίον, Agni, B**ánků̊υ

АГНЕЧЬВЫХЪ gen. pl. m.

имени в҃і. обѣманадесѧте апл҃мъ агнечьвыхъ

δώδεκα ὀγόματα τῶν δώδεκα ἀποστόλων τον͂ἀρτίον

Figure 4. The text as entered into the software package and exported in RTF format (The "**" represents 2 symbols that are unreadable in the original library card.)

Using the software package of the author's digitising the original library card took 15 minutes. As an end-user becomes more familiar with the software package digitisation will involve less time.

Digitised versions are far more acceptable as they are **portable** (insofar as they can be read cross-platform, and can be exported as an HTML document, an RTF document, a PDF document, and in various image formats).

By portability Birnbaum meant "that the format of an electronic document should not restrict the platform on which it can be processed. . ., so that Slavists should be able to access one another's electronic texts even though" working "on different platforms" (Birnbaum, 1995). Of course preservation was also hindered by a lack of portability. Accessing documents typed on a Macintosh computer in 1995 on a computer running Red Hat Linux in 2025 that contain Slavic text is a recipe for disaster.

But, in 1995, even working on a single platform there were many hurdles to access other people's Old Bulgarian texts. One only has to consider the multiplicity of methods of encoding Cyrillic texts current in the 1990s to see the very great difficulty at the time in sharing Old Bulgarian texts.

Here is a short, incomplete listing of some of the 8-bit Cyrillic encodings current then, and still in use in 2025:

KOI8-R
KOI8-U
KOI8-RU
KOI8-F
ISO-IR-111 / ECMA-Cyrillic, KOI8-E, ECMA-113:1986
GOST 19768-87
CP866

RUSCII / IBM CP1125 / x-cp866-u in UUPC/Ache
ISO-8859-5
ISO-IR-153 / GOST_19768-74
CP1251 / windows-1251

These are all based on the extended ASCII method of font encoding that only provide the computer user with 255-character slots. As the ASCII method (American Standard Code for Information Interchange) was initially implemented to cope with only the English alphabet, numbers, various signs, and 33 control characters (taking up slots 1 to 128), the standard was doubled to provide twice as many slots. But a moment's consideration will show how unsatisfactory this was given the multiplicity of writing systems used globally. Even for languages that used a Latin alphabet but also used diacritic marks ASCII was hardly satisfactory: rendering a French ç involved the rigmarole of several key presses on a computer keyboard. ASCII went through a large number of revisions: but the restriction to 255-character slots meant that as soon as anyone wanted to digitise text outside the Latin alphabet they were beset by problems.

A proposal for some sort of universal character encoding system was first mooted in about 1987, and work on the Unicode standard started in 1988. This was pointed out and promoted by Kempgen: "Clearly, the future of encoding is Unicode" (Kempgen 1995).

As of 9 September, 2025 there is a whole lexicon of Old Bulgarian characters available in Unicode version 17.0. This does not mean fonts are not being employed, that while being Unicode compliant to a certain extent, contain characters in the Personal Private Use Areas, which means that, unless one has that font on one's computer or it is transmitted in Portable Document Format (PDF) it may be partly unreadable. For example, the CyrillicaBulgarian10U truetype font contains characters in the Private Use Area Range: E000-F8FF: some of which are now available in the Cyrillic Extended-B Range: A640–A69F table, although it does also include characters that are currently not provided for in Unicode version 17:

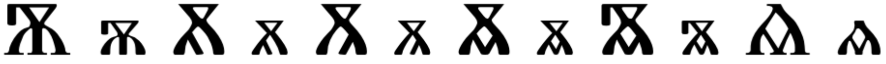


Figure 5. Characters not available in Unicode version 17.0

Using the Unicode set of characters comfortably on any computer system presupposes the following:

1. Access to a font that contains a complete set of the Old Bulgarian characters and diacritics defined in the Unicode standard.

2. An adequate software package that allows end-users to rapidly leverage the Cyriilic character set offered by the Unicode standard: ideally as easily as typing in a modern writing system.

Having realised the lack of an adequate software package that allows end-users to rapidly leverage Indic character sets offered by the Unicode standard for the digitisation of Indic documents (e.g. those written in Sanskrit and various Prakrits) the author developed between 2012 and 2024 a cross-platform software package (for Microsoft Windows, Apple Macintosh, and Linux) to digitise ancient Indic texts, offering the 2 dominant abugida systems used for those texts: Devanagari and Grantha. The software package ("Devawriter Pro") is now widely used for Sanskrit input.

In 1996 the author attempted to create an adequate Old Bulgarian font ("Kotlenski") using the software package Fontographer 4.0 without much success.

It seemed that leveraging the skill the author developed with Devawriter Pro, and Sheba:Makeda (for Ethiopic/Ge'ez digitisation), and his experiments with both bitmap and truetype Old Bulgarian fonts to produce a similarly useful tool for Bulgarian might help to solve some of the problems outlined above.

Devawriter Pro was developed using LiveCode Community edition (Open Source) until the LiveCode company discontinued that in 2021, and subsequently with OpenXTalk, a software package developed on the code base of the last Open Source version of LiveCode. Sheba:Makeda has been developed solely with OpenXTalk.

Towards an Effective Software Package for Old Bulgarian Text Digitisation

The author developed a prototype of a Old Bulgarian software package in 2013, but it was highly unsatisfactory, as was the font the author used used with it. Subsequently the Unicode consortium have expanded their Old Bulgarian/Cyrillic offering, and the author's software development skills have improved considerably.

The goals of any software package for Old Bulgarian digitisation should be:

1. An easily usable interface (as intuitive as possible).

1.1. This software package should, ideally, be both usable and feature-identical on the Windows, Macintosh, and Linux platforms.

2. A system for Glagolitic digitisation.

3. A comprehensive Old Bulgarian Cyrillic input system including the ability to leverage:

3.1. The Old Bulgarian combining letters (Unicode 2DE0 – 2DFF).

3.2. The Old Bulgarian superscript and subscript letters (Unicode 1E030 – 1E08F).

3.3. Ancient Greek letters (Unicode 0370–03FF). This should also allow access to supposedly non-standard forms such as Stigma, Digamma, and Koppa.

3.4. The Old Bulgarian numbers (c.f. Dejić & Dzebić). While these are not included in Unicode 17.0 it is easy enough to compose them from their constituents.

3.5. Roman numbers (Unicode 2150–218F).

3.6. Greek Acrophonic numbers that do not refer to money (Unicode 10140–1018F).

Attempts have been made to provide a subset of these (c.f. https://sites.psu.edu/symbolcodes/languages/europe/cyrillic/cyrillicchart/) as a series of codes that can be (extremely slowly) entered into an office package. All that they provide is a set of Old Bulgarian Cyrillic input codes.

While in *Computer Processing of Medieval Slavic Manuscripts* (1995) Unicode is mentioned multiple times, it is interesting that the report (https://www.obshtezhitie.net/report.htm) does not mention it at all.

The huge range of on-Latin alphabets lead to the idea to create a standard which will provide free space for all of them. Thus, the UNICODE standard was set up [UNICODE 92]. The 'piece' of Cyrillic symbols included into it does not look completely satisfactorily, especially in the presentation of the Medieval Slavic alphabet (Paskelova & Dobreva).

The Unicode standard version 17.0 was released on 9th September 2025 with a vast number of Cyrillic glyphs, and version 18.0 (slated for release within 18 months) containing an additional glyph, demonstrating that the Cyrillic code blocks of the Unicode standard are still very much a work in progress.

A Comprehensive Unicode Font for Old Bulgarian

While a completely comprehensive font for Old Bulgarian input is not currently possible, the author has prepared an Open Source font based on SIL Gentium, a font that contains a subset of the Unicode lexicon of Cyrillic glyphs. This necessitated the inclusion of a complete Glagolitic range (ranges: 2C00–2C5F and 1E000–1E02F), as well as both missing Cyrillic Unicode ranges and the addition of characters that SIL Gentium did not possess. The resulting font (GBP_RM.ttf) contains all the Unicode Cyrillic characters as well as all the Unicode Glagolitic characters, and an extensive number of diacritical marks not present in SIL Gentium.

Graphic Use Interface Decisions

Managing cognitive load – the amount of information people can process-is essential to effective teaching or training. Indeed, bombarding learners with too much information at once, called cognitive overload, is one of the chief obstacles to learning. (Clark 1995)

While this software is not conceived as a teaching instrument (end-users should have some idea of Old Bulgarian), using it may be divided into 2 phases:

1. What Clark terms "cognitive apprenticeship", where the end-user learns how to leverage the software to serve their needs.

Cognitive apprenticeship is designed to build expertise. (Clark 1995)

2. Normal usage; where, once an end-user has become familiar with the software package they are able to use it intuitively just as they would with any other software package they use on a regular basis.

One of the author's main aims is to reduce stage 1 to as small a period as possible by reducing the cognitive load an end-user will experience when first they attempt to get to grips with the software package.

The author also had in mind *Gagne's Nine Events of Learning*, and that all of those events should be present for end-users to effectively get to grips with the software package, and subsequently use it for successfully digitising texts. (https://www.td.org/content/atd-blog/we-think-therefore-we-learn).

Robert Gagne's 'events' are:

1. gain attention
2. state objective
3. recall prior learning
4. present stimulus
5. guide learning (showing examples, coaching)
6. practice
7. feedback
8. assessment
8. transfer.

How these have been manifested in the software is as follows:

1. gain attention: make the software package attractive, but lacking distractions, so it can be seen as businesslike.

2. state objective: the objective is stated boldly on the first window when the software is opened: "Old Bulgarian Digitisation."

3. recall prior learning: as the software may be used directly from the end-user's computer keyboard, and features the ability to use the following inbuilt keyboard layouts (Bulgarian Traditional, Bulgarian Phonetic, and US

English) the end-user does not have to learn obscure key commands to rapidly produce text.

This is also termed a*ndragogy*, a term for tapping into prior experience, as differentiated from 'Pedagogy' which involves teaching skills to (possibly younger) learners who do not have transferrable skills to bring to the learning experience.

4. present stimulus: stimulus is present insofar as an end-user can manage to produce text extremely rapidly.

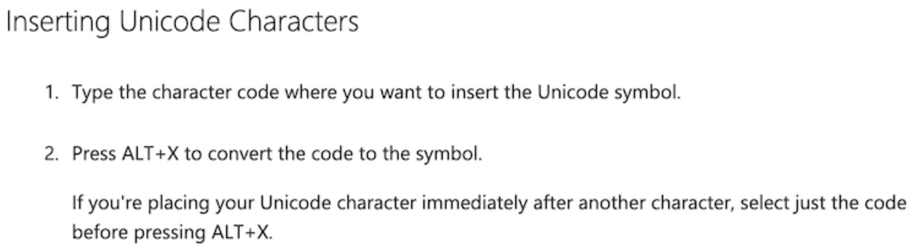
5. guide learning (showing examples, coaching): this is present in the extensive introductory series of screens, and the ability for the end-user to revisit some of the introductory screens from inwith the main input interfaces.

6. practice: that should almost speak for itself.

7. feedback: is present in that the interface is WYSIWYG (What You See is What You Get), so any digitised text is instantly seen in user input.

8. assessment: any text produced via the software can be exported in 3 formats (HTML, RTF, Text) allowing results to be assessed even if the assessor does not have access to the software package.

9. transfer: the end-user will rapidly both transfer skills the have previously learnt to leveraging this software package, as well as acquiring new transferrable skills through its use.



Inserting Unicode Characters

1. Type the character code where you want to insert the Unicode symbol.
2. Press ALT+X to convert the code to the symbol.

   If you're placing your Unicode character immediately after another character, select just the code before pressing ALT+X.

Figure 6. How to insert Unicode characters in Microsoft Word

Rather than expect end-users to either memorise or continually refer to an obscure table of key combinations for character entry, a decision was made to make the entry methods resemble a physical computer keyboard as much as possible:

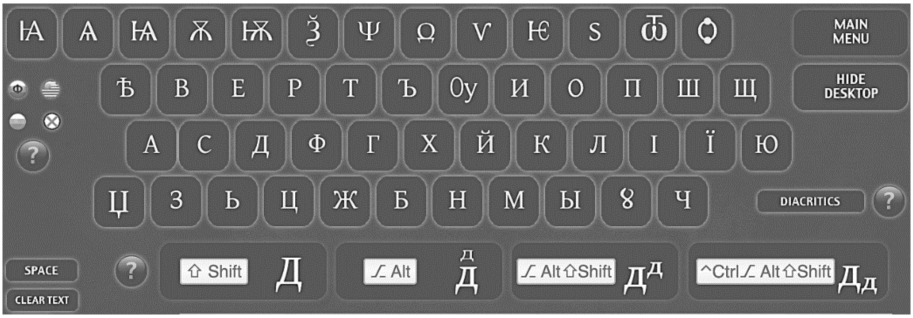


Figure 7. Phonetic Bulgarian keyboard entry method

This should allow a large number of readily transferrable skills the end-user already possesses to come into play to accelerate adoption of the software package.

Usability Decisions

To add diacritics to letters commonly involves multiple awkward key presses. For instance on MacOS 12 to add the cedilla to a 'c' in the French word *Français* one has to keep one's finger down on the 'c' button and then press the '1' button, which will normally involve crossing one's hands on the keyboard. With other words with a variety of possible diacritics the whole system is hardly consistent.

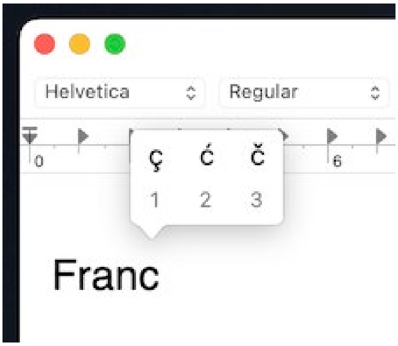


Figure 8. Attempting to type 'ç' in TextEdit on MacOS 12

Computer keyboards currently connected to contemporary computers running Windows, Macintosh, or Linux usually have a selection of modifier keys (although these vary from system to system, there is a common core):

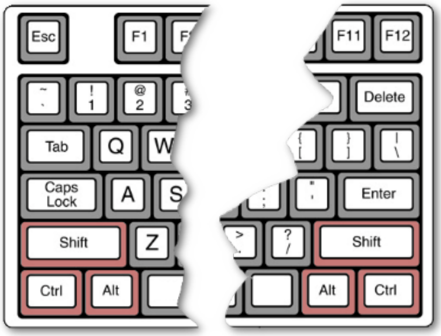


Figure 9. Standard modifier keys on contemporary computer keyboards

Therefore, a decision was taken that glyphs should be entered either by simple key presses, or key presses along with one or more of the cross-platform modifier keys. It was also decided that the effects of using the modifier keys should be visible on-screen at all times. See Figure 7.

The modifier keys are only used to enter upper-case, supersuper script, super script, and subscript variants of letters. As diacritics are all constituted as combining characters in Unicode a decision was taken to input diacritics via a dedicated keyboard layout.

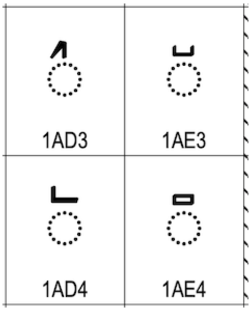

1AD3
1AE3
1AD4
1AE4

Figure 10. Combining diacritics illustrated in the Unicode documentation

The illustration shows how a diacritic would be positioned above a character. (Range 1AB0–1AFF) Each character is denoted by a unique hexadecimal address. Therefore each diacritical mark inside the font is given a zero width so it will be positioned directly above a preceding character.

With the Chudov translation of the New Translation on Mount Athos into East Slavonic the adoption of Greek diacritics into Old Bulgarian seems to have become normative (Knoll, 2018). This clear if we compare an earlier and a later recension of the *Codex Supraliensis*:

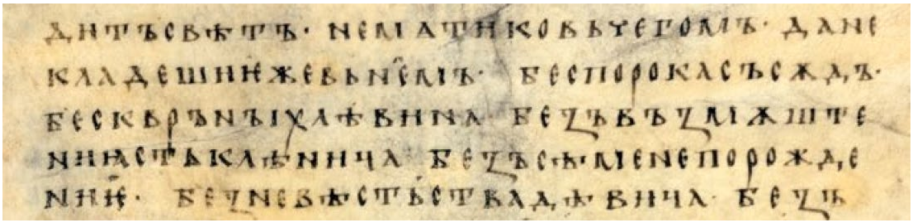


Figure 11. *Codex Supraliensis* early version

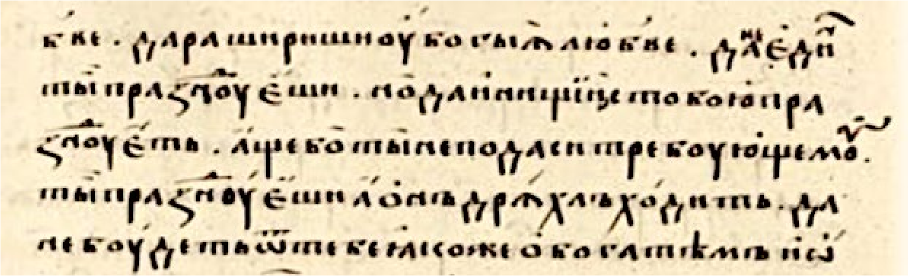


Figure 12. *Codex Supraliensis* later version

As far as can be determined the Unicode standard does not include the following diacritical marks at present:

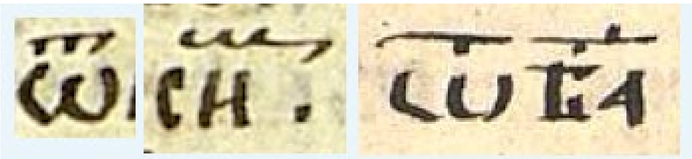


Figure 13. Diacritical marks or superposed characters

A question this raises is, "How *authentic* (in terms of 'look and feel') to original manuscripts do want digitised editions to be?" *versus* questions related to *functionality*.

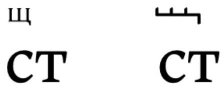


Figure 14. Functionality *versus* authenticity

Comparing the *functional* rendering of a superlinear ‘sht’ on the left with an *authentic* rendering on the right, makes it clear that there will always have to be a level of compromise as to how authentic a digitised text can be, as it is not possible to encode every scribe’s individual quirks that they exhibited in the original texts. This compromise will, inevitably have some unsuspected effects:

> *The digital reproduction of texts, maps, still images, moving images and sound brings about a profound revolution in how we relate to cultural artefacts past and present. What was separated by time and in completely different media and realms, becomes reduced to bits and bytes and travels side by side in fiberoptic networks around the globe, made available to us immediately as we request it. This profoundly not only changes the cultural artefact itself, but also the way we relate to it.*” (https://tei-c.org/Vault/Workgroups/CE/chibs-2002-paper.html)

Ease of Use

A decision was made to enter text, either by keyboard entry, or by point-and-click using the end-user’s Mouse, Trackball, or Trackpad. While point-and-click is useful for single character/word insertions it rapidly becomes inefficient when entering larger units of text.

As there are 2 dominant keyboard layouts for Bulgarian Cyrillic entry: the “Standard” Bulgarian and the “QWERTY” Bulgarian, two alternate methods of Old Bulgarian text entry were made available within the software package.

Although keyboard overlays were also provided for all input methods allowing end-users to easily align their physical keyboards with whichever virtual keyboard they had chosen.

Missing characters and diacritics

The author will be submitting a request for inclusion in Unicode version 19.0 of the following missing characters and diacritical marks:

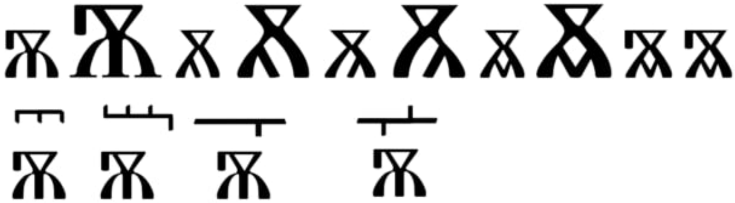


Figure 15. Characters and superposed forms not currently included in the Unicode standard

The author has included these characters and diacritical marks in his font, and has made them accessible via the software package.

OpenXTalk

This is a RAD-IDE (Rapid Application Development – Integrated Development Environment) that is a linear descendent of the HyperCard software package for MacOS 7 – 9.

It allows for the production of software executables on Apple Macintosh, Microsoft Windows, and Linux. The author does his programming on both Macintosh and Debian-derivative Linux.

This is the programming environment the author has used to build the software package.

The author has programmed software packages for a wide variety of institutions (Southern Illinois at Carbondale., The UAE University, Al Ain., St Andrews University, Scotland., His own EFL school., Kauai's Hindu Monastery, Kauai, Hawaii.) as well as several open source language digitisation packages for Sanskrit and associated Indic languages, Ge'ez Ethiopic, and Anglo-Saxon, as well as an educational software package for Scottish schools (Listen Hear).

OpenXTalk employs an object-oriented programming model, and has an WYSIWYG (What You See is What You Get) interface that bypasses the conventional compile-run routine, allowing a direct visual correspondence between the programming interface and the result.

While the **OpenXTalk** interface provides premade several buttons the author has chosen to use images as buttons to guarantee that they will appear the same whether displayed on Linux, Macintosh, or Windows. The images to be used as buttons were made using **OpenXTalk** itself. Additional graphic material was prepared using **GIMP** (https://www.gimp.org/).

The font (GBP_RM.ttf) was authored using **FontForge** (https://fontforge.org/en-US/).

The **GBP-RM.ttf** font is a heavily modified font derived from the **Gentium Bold Plus font** made available by the **Summer Institute of Linguistics** https://software.sil.org/gentium/download/.

Software Interface

The software package uses a **copyleft** licensing system.

Copyleft open-source licenses require derivative works or modifications made to the software be released under the same *licensing agreement. This ensures that the source code remains free and open for general use.* (https://montague.law/blog/understanding-open-source-license-definition-types-and-comparison/)

Introductory Section

The software package features 15 introductory screens to help end-users to become
acquainted with the features of the software package.

Main Menu

Once an end-user is familiar with the software they can easily bypass the introductory screens and proceed directly to the Main Menu screen which offers Traditional Cyrillic entry, Phonetic Cyrillic, and Glagolitic entry methods.

The author has not included a direct link to the Greek input method as Greek is not the main language this software is concerned with. The Greek input method can be accessed from within all of the main entry method screens.

Supplementary windows:

Diacritics:

The diacritic window allows access to a large number of diacritic marks used in Old Bulgarian texts.

Extras:

The 'Extras' window allows access to a wide range on additional Cyrillic characters used in historical Old Bulgarian texts. A decision was taken not to include the whole Unicode Cyrillic range as many of the characters were never used in historical Old Bulgarian texts.

Extras 2:

The 'Extras2' window allows access to further Cyrillic characters used in historical slavic texts as well as diacritical forms. This windows aslo allows access to those characters mentioned above that are not currently included in the Unicode standard.

There is room available for inclusion of further characters and/or diacritics in future versions of the software, should they be required.

Greek:

The Greek window allows access to Greek characters including historically attested forms such as digamma and stigma. Diacritics can be applied via the Diacritic window.

Roman Numerals:

This window allows access to Roman numerals. The modern ('Arabic') number for each Roman Numeral is displayed in a small window onscreen to aid in recognition.

Owing to the large number of Old Bulgarian numeric forms they had to have a separate window to that one from Roman numbers.

Old Bulgarian Numerals:

Owing to the large number of Old Bulgarian numeric forms they had to have a separate window to that one from Roman numbers. The modern ('Arabic') number for each Old Bulgarian Numeral is displayed in a small window onscreen to aid in recognition.

Exporting Text:

It is currently possible to export text in RTF (Rich Text Format), HTML (HyperText Markup Language), or Text formats. PDF (Portable Document Format) and a variety of image formats can be added at a later date should they be required.

Obtaining the Software:

The software package is downloadable here:
https://richmondmathewson.owlstown.net/pages/8572-pismo

REFERENCES

Ambrosiani, 1991: Ambrosiani, P. *On Bulgarian Accentuation*. Stockholm: Almqvist & Wiksell Int. ISBN 91-22-01446-2.

Banks, 1992: Banks, I.M. *The Crow Road*, London: Abacus. ISBN 0349139156

Berdnikov. 1999: Berdnikov, A., Lapko, O. *Old Slavonic and Bulgarian in TEX and Unicode*. https://www.semanticscholar.org/paper/Church-Slavonic-in-TEX-and-Unicode-1-Old-Slavonic-Berdnikov/e528d10c57a92b38f854435ace8ae56cd096dd06, retrieved on 1 October 2025

Birnbaum, 1995: Birnbaum. D.J. How Slavic Philologists Should Use Computers. In: Birnbaum, D.J., Bojažiev, A.T., Dobreva, M. P., Miltenova, A. L. (Eds.): *Computer Processing of Medieval Slavic Manuscripts*. Sofia: Marin Drinov Academic Publishers, 19 – 29. ISBN 954-430-417-7.

Birnbaum, 1995: Birnbaum. D. J. Informational and Presentational Units in Early Cyrillic Writing. In Birnbaum, D. J., Bojažiev, A. T., Dobreva, M. P., Miltenova, A. L. (Eds.): *Computer Processing of Medieval Slavic Manuscripts*. Sofia: Marin Drinov Academic Publishers, 41 – 50. ISBN 954-430-417-7

Clark, 1995: Clark. R. C. Authorware, Multimedia, and Instructional Methods. In *Taking The Plunge*. San Francisco: Macromedia, Inc. 3 – 28

Dejić & Dzebić, 2015: Dejić. M., Dzebić S. Најстарији руски (словенски) математички рукопис, дело монаха Кирика. 2015. Accessed: 1 October 2025. http://www.inovacijeunastavi.rs/wp-content/uploads/Inovacije2-15/07Dejic.pdf

Kempgen, 1995: Kempgen. S. Complex Script Systems on Today's Personal Computer. In Birnbaum, D.J., Bojažiev, A.T., Dobreva, M. P., Miltenova, A. L. (Eds.): *Computer Processing of Medieval Slavic Manuscripts*. Sofia: Marin Drinov Academic Publishers, 68 – 79. ISBN 954-430-417-7

Knoll, 2018: Knoll. V., *Greek minuscule in the 14 century Slavonic manuscripts*. https://www.academia.edu/37585312 Greek_minuscule_in_the_14th_century_Slavonic_manuscripts_2018_, retrieved on 1 October 2025

Mitrevski, 1995:Mitrevski Ljupčo. Fund, Fonts and Character Set of Church Slavic Graphemes. In Birnbaum, D.J., Bojažiev, A.T., Dobreva, M. P., Miltenova, A. L. (Eds.): *Computer Processing of Medieval Slavic Manuscripts*. Sofia: Marin Drinov Academic Publishers, 114 – 119. ISBN 954-430-417-7

Paskaleva & Dobreva 1995: Paskaleva. E.& Dobreva. M., New Tools for Old Languages, Computer Processing of Bulgarian Texts. In Birnbaum, D.J., Bojažiev, A.T., Dobreva, M. P., Miltenova, A. L. (Eds.): *Computer Processing of Medieval Slavic Manuscripts*. Sofia: Marin Drinov Academic Publishers, 30 – 41. ISBN 954-430-417-7

Totomanova & Ganeva 2014: Totomanova. A., Ganeva. G. Digital Tools for Studying Medieval Lexis and Grammar. Sofia. 2014. https://www.kmnc.bg/files/elmanuscript2014_final_.pdf ISBN 978-954-9787-25-2.

ELECTRONIC SOURCES

https://segfault.kiev.ua/cyrillic-encodings/

https://www.unicode.org/history/summary.html

https://richmondmathewson.owlstown.net/

https://fontforge.org/en-US/

https://sites.psu.edu/symbolcodes/languages/ancient/ocslavonic/

https://livecode.com/

https://www.openxtalk.org/

https://online-fonts.com/fonts/cyrillica-bulgarian

https://www.unicode.org/L2/L2025/25112-cyrillic-with-connecting-bar.pdf

https://support.microsoft.com/en-au/office/insert-ascii-or-unicode-character-codes-in-word-e97306f7-00c1-490d-9920-c924ca443f87

https://software.sil.org/gentium/

http://gorazd.org/kartoteka/?envLang=en

https://istoria.bg/booktext/ivan-asen-vtori-tarnovski-nadpis

https://libguides.mst.edu/c.php?g=335435&p=2256780

https://digital.libplovdiv.com/bg/view/642b0b36a556affbd1cd45fc

https://approachingdh.commons.gc.cuny.edu/2018/03/13/the-visualization-digital-studies-and-literary-history/codex_suprasliensis_070811.pdf., Institute for Literature Bulgarian Academy of Sciences. Undated.

https://tei-c.org/Vault/Workgroups/CE/chibs-2002-paper.html

https://www.td.org/content/atd-blog/we-think-therefore-we-learn. Accessed 2025.

https://montague.law/blog/understanding-open-source-license-definition-types-and-comparison/

https://www.gimp.org/

https://fontforge.org/en-US/

https://software.sil.org/gentium/download/

OpenXTalk is downloadable here: https://www.openxtalk.org/

John R. Mathewson, MA, MSc.
Plovdiv, Bulgaria
e-mail: richmondmathewson@gmail.com