THE DUKE LANGUAGE TOOLKIT: FOREIGN ALPHABETS FOR THE IBM PC
Jeffrey William Gillette

ABSTRACT

This paper is given as a progress and a goals report for the Duke Language Toolkit program. It is also a solicitation for input for mutual benefit of linguists and humanists.

KEYWORDS: Duke Language Toolkit, foreign character fonts, IBM PC, Victor 9000, IBM 3081, CMS, MVS, IBM enhanced color monitor, P-C Write, BYU Concordance program, Greek, Hebrew, Coptic, Cyrillic, Armenian, Amharic, Arabic, Chinese, Syriac

We who spend our lives studying ancient texts, teaching esoteric languages, or analyzing minute linguistic structures, have one basic requirement of our computers: that they fully and accurately present the alphabets we use. Imagine the engineer's interest in a computer that had no capacity for floating point arithmetic!

At the start of the summer of 1984, Duke University had one computer system which could display the Greek alphabet—an expensive minicomputer without so much as a word processor! Some months before, we had purchased several Victor 9000 microcomputers, which boasted the potential for 4000 customized characters, but had no software to realize this vision. As the summer of 1985 draws to a close, Duke boasts several dozen microcomputers writing Amharic, Arabic, Armenian, Chinese, Coptic, Cyrillic, Greek, Hebrew, and Syriac, and making these languages available to word processors, concordance programs, and even mainframe computers. We believe that 1985-86 will be the year the microcomputer changes the face of humanistic studies at Duke.

The hardware focus of the Duke Toolkit has been the IBM PC, with some experimentation done on the Victor 9000 and the IBM 3081 running CMS and MVS. We would here like to present to the larger CALICO community a progress report on the Toolkit. We will take up three heads: the Toolkit itself (as it currently stands); the interface of the Toolkit to other commonly used software; and the direction of the Toolkit project over the next nine months.

The Duke Language Toolkit

The Duke Language Toolkit is a collection of programs and utilities we have written to provide foreign alphabet capabilities on the IBM PC. Files on the Toolkit diskette divide into two categories: those utilities that make foreign characters available to the IBM PC, and patches, interfaces, and modifications to (commercially) existing software that enables them to take advantage of foreign alphabet capabilities.

The primary requirement for the Toolkit is the IBM Enhanced Graphics Adapter with a monochrome or enhanced color monitor (the standard color monitor does not work with the Toolkit). The EGA has three primary features which, we feel, make it the only adequate display option currently available. First, it allows for RAM-based (i.e., custom generated) character sets, which allow us to create our own fonts and use them with almost all commercial software. Second, it allows two character sets to be combined into one large font, which allow us to handle alphabets of up to 512 characters. Third, it doubles the graphics resolution, effectively making text and graphics modes equal in legibility. While we hope the future promises more exciting alternatives for display hardware, we doubt that foreign alphabet work can be comfortably handled with anything less powerful than the EGA.

The primary programs in the Toolkit are Makefont, a character editing utility with which we produce new alphabets, and Loadfont, the utility which loads a custom alphabet into the Enhanced Graphics Adapter, and prevents other software from "loosing" the font. Two smaller programs, Loff and Lfon, switch between the normal IBM character set and the custom alphabet. These functions can also be accessed from standard interrupts in programming languages like Basic or Pascal. Two experimental utilities we hope to be able to distribute in the near future are Dukecom, a communications program that translates data from a mainframe computer into the foreign language, and Printfont, a print spooler which takes a standard text file and translates it into the code sequences required by graphics and by laser printers.

The alphabets currently available in the Toolkit include Greek (with iota subscripts, breathing marks, and most accents), Hebrew (unpointed), Coptic, and Cyrillic. We hope to see Armenian, Arabic, and Amharic before the end of the year.

By using the EGA, standard IBM interrupts, and arranging our fonts to preserve graphics characters when possible, we have focused on maximum compatibility with already popular PC software. This has created an environment where there is already a large number of programs which run with the Toolkit, and this allows the
Interfacing to Commercial Software

We have yet to discover a commercially available program, which runs on a monochrome IBM PC, that is wholly incompatible with the Duke Language Toolkit. The only real difficulties appear to be those programs which make no provision for displaying characters 128-255 (WordStar is the most popular, and discouraging, example of this). In this section, we would like to discuss a few of our more promising success stories.

For many reasons, PC-Write is, without a doubt, the most promising tool for foreign alphabet word processing available. Bob Wallace has made his program transparent in the way it handles special characters, freeing it from the use of special reserved codes. PC-Write also has the most powerful keyboard redefinition capability short of Emacs. The use of a "dead key" to implement characters which must be "composed" (like vowels with accents) is also crucial. But perhaps the two most significant advantages of PC-Write are the availability of source code (for the inevitable modifications) and the reasonable price—free (although we strongly encourage every user to support Wallace with their $75 registration fee).

By creating a series of "ruler" files (as PC-Write calls them), we have created Greek, Cyrillic, and other keyboards. Switching between English and the foreign keyboard is as simple as hitting a function key, so even the occasional technical term scattered throughout a scholarly paper is relatively painless.

To create useful foreign keyboards a "dead key" is utilized. The "dead key" feature operates by means of three lists of characters. Character-one followed by the "dead key" followed by character-two causes PC-Write to look down list one and list two to find the right combination of characters. Character-one is then replaced with the corresponding character in list three, and character-two is discarded. Thus the sequence 'a,' "dead key," produces 'a'-circumflex. By redefining these lists (through a patch in the executable file, a rather simple process), we now use the "dead key" function to generate Greek breathing marks and accents. This facilitates the rapid input of Greek text without learning a series of strange and difficult key assignments or "escape" sequences. Those who have access to PC-Write source code can enlarge the "dead key" lists (the current limit is 48 combinations), and combining this feature with the keyboard definition capability makes possible fairly simple input, even with complex alphabets like Amharic (where each symbol represents a consonant-vowel combination).

A second striking success for the Toolkit has been the Brigham Young University Concordance program. Although yet in its infancy, and still possessing its share of "bugs," the BYU Concordance holds considerable promise for the quantitative (and qualitative) study of terms, themes, and ideas in texts.

Using the Duke Language Toolkit and our modified PC-Write, we have been able to edit the four Gospels of the New Testament (in Greek) into a file, concord them, and operate the BYU View program on the text. The concord program allows us to define our own sorting order, so the word lists come out in the proper Greek alphabetical sequence. Using Newkey (another "freeware" program), we can define a simple Greek keyboard, and provide both Greek and English input to the concordance program.

One of the pleasant surprises of this project has been the ease with which texts already on campus (used by minicomputer and mainframe computers) can be downloaded to the microcomputer. By running the text through a pair of filter programs, and by using a word processor to clean up the text (with global "search-and-replace" functions), a document like Matthew's Gospel (115 pages in the Greek text—110K in its final form) can be prepared for the BYU Concordance in about 2 hours from start to finish (only half of that time actually requires human intervention). The implications of this lead to our final topic.

Future Directions for the Toolkit

As the Duke Language Toolkit project looks forward, we see five primary goals—four of which are short-range, and one long-term.

Our chief priority is to get colleagues in the various departments using the microcomputers. To do this, our first two goals are placing equipment throughout the university (and teaching colleagues how to use it), and making large quantities of interesting data available for computer-assisted research. We are encouraging the purchase of several "scholar's workstations"—our term for an IBM AT with two Bernoulli drives, an Enhanced Graphics Adapter and Monochrome monitor. We feel this configuration provides adequate computing power, is flexible enough to handle new hardware/software alternatives, and maintains a per-work-station cost which is somewhat compatible with the real world of university budgets.

For the second goal, the conversion of large quantities of text into microcomputer format, we are requesting the cooperation of colleagues in the various departments (both at Duke and at other institutions). We hope to focus on data which already exists in computer usable form, and will supplement these texts with new material prepared using our Kurzweil Data Entry Machine.

The third goal is extending the Toolkit to include more languages. Currently, alphabets containing less than 128 characters (which can be acceptably displayed in 13 x 8 pixel resolution) are incorporated into the Toolkit with no difficulty. We hope to raise that limit to 256, 384, and perhaps even 512 by summer 1986. We are interested in working with scholars who wish to develop new languages, and invite queries.

Fourth, we will be incorporating the Toolkit into Calis, our authoring system for the IBM PC. Duke has already announced plans to provide computer-assisted instruction in modern Greek, Arabic, and Amharic, and we intend to provide delivery on the scholar's work-station described above.

A long-range issue is the question of languages which cannot be accommodated within the limitations of the normal IBM text mode. Chinese is one example of a language with far too many characters to store in RAM-based text fonts, and with far too much detail to be displayed adequately in cells of 8 x 13 pixels. Languages making heavy use of diacritical marks (like "pointed" Hebrew
or Arabic) also fall into this category. These languages raise problems like the tradeoff between the flexibility of graphics mode display, versus the speed required for word processing. There are also problems of standards—in what format will text be stored, or how will keyboard input be managed. Ultimately, the goal is maximum compatibility, so that the core character display software can accommodate new languages with minimum modification. Toward this goal, we would like to encourage some type of forum for the discussion of the technical issues raised by these languages—both from the perspective of the language scholar and from that of the computer engineer.

We hope that sharing some of our experiences from the Duke Language Toolkit project will further the discussion of foreign alphabets and microcomputers. We offer the fruit of our research to the larger academic community as an early and provisional tool which, we hope, will, by your criticism and refinement, evolve into a mature resource for the benefit of all scholars.

The Duke Language Toolkit, though barely a year old, has borne significant fruit for humanities computing. Yet we feel ourselves to be standing at the very beginning of a much larger task—one that will soon change the way scholars look at literature, and may ultimately change the way all of us look at language itself.

[Note: The Duke Language Toolkit is available to institutions and to interested individuals free of charge. To obtain a copy of the Toolkit, send two diskettes to the author at the Divinity School, Duke University, Durham, NC 27706]