

LINGUIST – SOFTWARE INTERACTION

1. Introduction

Every year there are more and more computer applications out there for linguists to use. Programs specifically written for linguists range from programs for speech sound analysis to dictionary making software to programs helping with cognate recognition. Much of the software is written by linguists for themselves and other linguists.

I first realized the extent of the potential problems faced by linguists when dealing with software myself when I was involved for two years in the development of a computer application for historical comparative linguistics ([Wordcorr](#), principal investigator: Joseph Grimes). This program was developed under a grant from the National Science Foundation in a cooperative effort between linguists and a professional software company. During my work this project, I noticed that there are certain communication barriers between linguists and software developers. What seems obvious to linguists was often anything but obvious to programmers, and vice versa. If there had not been close and frequent communication between the linguists and the developers, the final product would have been less useable to linguists.

While the ideal case would be to have software for linguists developed by linguists, or in close cooperation between linguists and software developers, not all linguists have sufficient knowledge of computers and software to design or create software, and not all software developers have access to linguists.

When a linguist does develop a computer program for linguistics, some of these problems are reduced. Ideally, the linguist knows what the application should be able to do, what the limits of software are, and how to combine their linguistic knowledge with their own computer know-how. However, even then, if the program is intended for wider use, problems may still persist. A user who has less extensive knowledge of computers than the creator of the program may not be able to learn quickly how the program works, or to ever fully exploit the program's potential.

The purpose of the survey discussed in this paper was to investigate the current situation in linguist-computer interaction. This paper summarizes some of the most common problems and issues associated with software used by linguists. This paper also contains some recommendations for linguists regarding use of computer programs for linguistics, and recommendations for creators of software regarding what kinds of software are needed by linguists.

The survey was conducted in two stages: in the first stage I sent the survey to students and faculty in the Department of Linguistics at the University of Hawaii at Mānoa. In the second stage, the survey was directed towards members of the [Linguist List](#).

2. Methodology

After testing a questionnaire with five subjects, a revised version was emailed out to the Linguistics Department mailing list. The text of the email is given in Appendix II. Once emailed responses were received, the text of the answers was copied and pasted into a separate document, and the original emails were deleted.

Before asking members of the Linguist List to fill out my survey, I changed the questions slightly from the original form used in the department wide survey, based on the responses I had received in the department survey. I posted a query to the Linguist List, directing respondents to my Web page where I had posted the questionnaire. Once emailed responses were received, the text of the answers was copied and pasted into a separate document, and the original emails were deleted. The text of the questionnaire posted on the Web page can be found in Appendix II.

3. Responses

3.1 Summary of Responses to the Department-wide Survey

I received 13 responses from students and faculty who had been or are using some type of computer software in their research. The responses included comments about the following software, in no particular order: Shoebox, Praat, Clan, Pitchwork, Macquirer, PCquirer, SoundEdit, Wordcorr, Childes, Adobe Premier, StatView, Excel, Word (95, 98, and later), Photoshop, iMovie, Emacs, Profiler, and PlotFormants

The most frequent complaints were: (1) incompatible with IPA and with other software, (2) difficult to use/steep learning curve, (3) too expensive.

Some of these are applications designed specifically for linguists, while others are more general types of software that can also be used for linguistics. For more details about the applications listed above, see Appendix I.

Users were most positive about programs that were easy to use or provided free and prompt support by email, were free or inexpensive to acquire, and had many functions needed by the linguist. In line with this, suggestions for improvements of specific applications mostly concentrated on making the software easier to use and more freely available.

The ‘wish list’ for linguistics computer programs contained the following: (1) a Shoebox-like program with a less complicated entry process, (2) software for searching multilingual interlinearized texts, (3) software for organizing data from a field methods class (by phonology, morphology, syntax and lexicon), (4) easier statistics software, (5) “something like audiamus...linked to a dictionary to provide interlinear gloss at same time”, (6) a program for transcription (to help save time).

In general, respondents felt that the way IPA is handled by computers needs to be improved, i.e. its stability across computers and programs, and compatibility with application software. Respondents mentioned that some archives are difficult to search, a problem which might be improved by changing the user interface. Students also felt that it would help if software use were taught in courses, or if there were at least workshops offered.

3.2 Summary of Responses to the Linguist List Survey

Six respondents filled out my survey in response to the query posted on the Linguist List. As with responses from the department-wide survey, responses to the Linguist List survey addressed both specialized linguistics software and more general software.

The responses included comments about the following software (no particular order): LaTeX, Prolog, DCG, Microsoft Word, Microsoft Word 2003 and below, UltraXML, Uniscribe, Open Office Writer, CanIPA, Knorpora, JabRef, Kile, Texniccenter, Perl, Shoebox, Toolbox, MS Paint, MS Word, Praat, Elan, Imdi, SoundEdit Pro16, Sound Studio, Cascadilla Press Mora font, SIL Phonetics keyboard for Mac, FileMaker, Windows, SAL, TECKit, XeTeX, iWork application ‘Pages’ on Mac, Microsoft Publisher, SIL’s Graphite font technology, Font Creator Program 4.0, Mozilla Editor, Emacs, CoolEdit96, MS Office, Star Office, Transcriber, Phono

The properties of software which were listed as most positive overlap with those found in the department-wide responses. The most frequently listed positive properties were the following: (1) flexibility, (2) intuitive interface, (3) simple, (4) good handling of IPA and other ‘unusual’ scripts. Also listed were: (5) fit to linguistic needs, (6) reasonable price, (7) the existence of good help files

The most frequently listed negative property of software was the problematic handling of IPA and other unusual characters. This was the only negative property which was listed by more than one respondent. Responses also contained the following comments regarding which properties are most disliked about software: (1) cannot save encoded txt, (2) expensive, (3) runs only on one platform, (4) incomprehensible manual, (5) unstable, (6) metadata values cannot be set flexibly

The following is a list of responses regarding what kind of software should be developed. Again, useable IPA heads the list.

- be able to use full (Unicode) IPA in different programs
- a program to convert a bracket notation into a branching tree, allowing edits

- software developed by linguists together with software designers
- a tool for rule-based collation
- a tool for annotation of audio and video on different tiers, with lots of capability for interlinking, creation of dictionaries, automatic updating, printing, and well-structured archiving
- easy to learn software

The question about the most serious problem with software for linguists elicited the following responses: (1) lack of portability across platforms and from one application to another, (2) linguist's high expectations, (3) phonetic transcription, (4) linguists afraid of software.

3.3 Discussion of Responses

This section addresses some of the most mentioned points in the responses. Issues such as the difficulty of the software and the cost of software were especially often mentioned in the department-wide survey.

3.3a Handling of IPA and other unusual fonts

This is the most frequently mentioned problem in responses to both stages of the survey.

Even though use of Unicode is getting more wide-spread, not all software used by linguists allows the input of full Unicode IPA (many IPA fonts are non-Unicode and require proprietary keyboard simulators).¹

The handling of different fonts, not just IPA, but the writing system of any language, is something that is absolutely vital for linguists. Copying and pasting, or entering numeric keys, may be acceptable for single words, or maybe even a whole sentence, but it becomes unacceptable as soon as you are dealing with larger pieces of text, word lists, whole dictionaries, long transcriptions, etc. In addition, it is not sufficient to be able to use 'unusual' fonts with just one program, such as a transcription program. The linguist has to be able to export the fonts to other programs. There has to be an agreed upon standard for IPA and other unusual fonts, easily portable between programs and platforms. At the moment, the number of programs that will handle unusual characters satisfactorily is limited. A program that can handle IPA and other unusual fonts, and in addition 'talks' well to other programs and can be used across different platforms, would certainly have an edge over its competitors.

One solution for problems with displaying special fonts which is possible with current technology is embedding fonts into an Adobe document. This can ensure that the characters appear correctly on any machine, because Adobe documents are essentially instructions to browsers and printers about where to put millions of individual dots. However, the document will increase in size quite a bit if many characters are embedded.²

3.3b Steep learning curve

The difficulty of software was addressed predominantly in responses to the department-wide survey.

There are a number of possible solutions to this problem. One solution suggested by Dr. Grimes³ and implemented for the Wordcorr program, is to have the help materials for a program written by a person who is not a programmer, and not familiar with software design. This way, with constant interaction between the software developer and the writer of the help materials, help information should be useable by people who have little experience with software.⁴

¹ Dr. Grimes (personal communication, 3/2005) points out that "The blocks of characters in Unicode that are designated as IPA do not cover the International Phonetic Alphabet standard; they must be supplemented from Greek and extended Latin blocks. There's also the question of whether all software should handle complete Unicode including right-to-left and CJK (Chinese-Japanese-Korean) characters. Wordcorr does, to the delight of our testers in China, who use Chinese as a glossing language."

² I would like to thank Jun Nomura for pointing this solution out, and Dr. Grimes for explaining it to me in more detail.

³ personal communication, March 2005

⁴ This is how the help materials were written for the Wordcorr program. Christine Hansen, a linguist and technical writer, wrote the help materials, interpreting what the programmers did for linguists with little software experience.

It is also helpful to have free and prompt feedback from the software developer or provider. Feedback by email would be sufficient. Shoebox is an example of a program which has prompt and comprehensible support by email. Praat, Pitchworks, PC/Macquirer, and SoudEdit Pro are examples which were labeled as straightforward and user-friendly in responses to my survey.

Another solution, suggested by respondents to the survey, is to provide more training in software use to students in linguistics. This is something which could be implemented on the department level. Possibilities include semester-long courses on a particular piece of software, or a series of workshops for different computer programs. The workshop approach might be more feasible for a small department, which cannot offer semester-long courses for each software product students might need. Having workshops for a number of different programs is also more useful if students in the linguistics department have varied interests. In my opinion, it is quite likely that once a student has attended one software workshop, further learning of the software in question, and even of other software, will be easier. If specific questions arise about an application, and they are not addressed in the help files, users may want to look for program-specific usenets, lists, or newsgroups on the internet. The problem may already have been addressed, or else the user can post a query for knowledgeable people to respond to.

3.3c Cost

This issue was addressed more frequently by respondents to the department-wide survey, although it was also mentioned by one respondent to the Linguist List survey.

Some programs which are used by linguists are quite expensive. I checked some of the prices online, and found applications which cost \$400 and more. This kind of cost is unaffordable for most graduate students. The obvious solution, of course, would be to lower the price of the software. It might also be possible for a department to acquire one copy or multiple copies of the program on an academic license for student use in a department computer lab. In the responses to my survey, Praat received especially positive feedback since it is free, while Shoebox was noted for being cheap (i.e. affordable)⁵

For potential users, another solution is to look for comparable open source software. The most ambitious solution would be for computer-savvy linguists to write the needed program and make it freely available.

3.3d Intuitive Interface

Problems with the interface were mentioned in response to the Linguist List survey.

This is, to a certain degree, a subjective property. However, there are probably certain features of an 'intuitive interface' that linguists would agree on. To create an interface that is useable for linguists, it may be a good idea to involve linguists with little or no programming experience in the design process, incorporating the linguists' feedback before the software is released to the 'general public'. Examples of software with a good user interface which were cited in the responses are JabRef and SoundEditPro16.

3.3e Flexibility and portability

Flexibility and portability were addressed by respondents to both stages of the survey.

This issue has already been addressed in terms of usage of unusual symbols. However, flexibility is a more general issue, including portability of software in general across different platforms, and the flexibility of software to be used for more than just one fixed function. Output of one program is often not useable by other programs. Respondents to my survey cited Shoebox and LaTeX as flexible software.

To solve the problem of portability, it is necessary for all newly developed software to be able to use an output format which is easily accessible by other software, such as plain text, or text in standard UTF-8 encoding, possibly in XML format.

⁵ Its successor Toolbox is free.

33f 'Wish list' for future software

Besides specific ideas about future versions of existing software, it seemed to me that in the department-wide survey two areas of concentration of the 'wish lists' were the following: (1) better capabilities for searching databases, and (2) transcription software.

From personal experience I know that while there are more and more databases available for information about languages, searching these databases can be problematic. The problems stem mainly from three reasons: the material is not organized to be searchable, it is not clear which terminology is used (now referred to as the "ontology" problem, or there is a problem with the user interface.

Non-searchable material may, for example, be documents in PDF format. If databases are set up, care should be taken that all material is searchable. If users have to spend a lot of time looking at each document separately to find what they are looking for, they may give up looking for it before they find it. Terminology can be a problem when a user is looking, for example, for a "focus" language, but the database describes the same language as an "ergative" language. In order to alleviate this problem, it would be helpful to provide a glossary of linguistics terms and their frequent counterparts in other frameworks or theoretical approaches. It would be ideal if terminology could be the same across databases, but I am not sure how realistic this is at the moment.⁶

In general, respondents to the Linguist List survey called for more interaction between linguists and software developers, and for software which can handle IPA and other special characters. More specific programs which would be useful for linguists are programs for converting bracket notation into tree notation, for annotating audio and video, and for rule-based collation.

4. Conclusion

It seems to me that the majority of problems with linguistic software are solvable with current technological capabilities. What is needed is mainly more flexibility on the side of software developers, and more know-how on the side of linguists.

Although the survey on the Linguist List produced responses about a large number of software applications from a small number of respondents, there were clear parallels to the responses from the department-wide survey in terms of the positive and negative features of software that were cited.

By far the most prominent problem with current software seems to be the difficulty associated with special fonts such as IPA, and other non-English fonts. Linguists want to be able to not just use special fonts in any one program, but they need to be able to then look at the data in different programs and on different platforms. It is no good to be able to create a document containing special fonts in a program for interlinearization with audio files, when one is then incapable of importing the same data into a program for dictionary editing. It is equally useless to be able to use special fonts on one's computer at home, when the whole document dissolves into gibberish when using a different computer trying to give a presentation at a conference. In my opinion, software will be more successful the more it 'talks' to other programs. Software makers have to agree upon a standard for special symbols, possibly Unicode, which is portable among different software systems.

It seems that linguists might need more training in software, in order to be aware of the capabilities and limitations of software. Learning to use one application in relative detail often gives the learner at least some idea about the general capabilities of computers.

In addition, non-computer-savvy linguists should be more involved in the development of software for linguists. Not all users of the software will have a thorough background in computer use; in fact, I think the majority of linguists would be helped by manuals and user interfaces which presuppose no more than a working knowledge of word-processing programs, and possibly some passing knowledge of Excel.

⁶ Check out <http://emeld.org/tools/ontology.cfm> for an attempt to standardize linguistic terminology. The focus of this summer's workshop (2005) will be on ontologies (thanks to Dr. Grimes for pointing this out to me).

Appendix I

1. Shoebox – “Shoebox is a computer program that helps field linguists and anthropologists integrate various kinds of text data: lexical, cultural, grammatical, etc. It has flexible options for sorting, selecting, and displaying data. It is especially useful for helping researchers build a dictionary as they use it to analyze and interlinearize text. The name *Shoebox* recalls the use of shoe boxes to hold note cards on which definitions of words were written in the days before researchers could use computers in the field.” (<http://www.sil.org/computing/shoebox/index.html>, 3/20/05)
2. Toolbox – “Toolbox is a data management and analysis tool for field linguists. It is especially useful for maintaining lexical data, and for parsing and interlinearizing text, but it can be used to manage virtually any kind of data.” (<http://www.sil.org/computing/toolbox/> 4/25/2005)
3. PRAAT – “a computer program with which you can analyse, synthesize, and manipulate speech, and create high-quality pictures for your articles and thesis” (<http://www.fon.hum.uva.nl/praat/manual/Intro.html> 3/20/2005)
4. Clan – “The CLAN Programs are downloaded, installed, and used as a single application. Functionally, however, CLAN has two parts. The first part is the CLAN editor which can be used to edit files in either CHAT or CA (Conversation Analysis) format. The editor also provides a wide range of additional functions, such as audio and video playback, linkage to audio and video, fonts for Roman and non-Roman orthographies, data validation, adding codes to files, and shipping data to other programs. The second part of CLAN is the set of data analysis programs. These programs are run from a separate window called the Commands window. The results of the analytic programs are sent to the CLAN Output window.” (<http://chilides.psy.cmu.edu/clan/> 3/20/2005)
5. Pitchworks – used “for pitch, labeling, and other intonation related studies.” (<http://www.sciconrd.com/pworks.htm> 3/20/2005)
6. Macquirer and
7. PCquirer - “capture, edit, and analyze audio signals with a click of a button. The analysis functions include FFT/LPC, Spectrogram, intensity, and pitch.” (<http://www.sciconrd.com/multi.htm> 3/20/2005)
8. SoundEdit – “digital audio editing software that offers a variety of audio recording, mixing, editing, analyzing and effects processing possibilities.” (<http://www.rmbsoft.com/sep.asp> 3/20/2005)
9. Wordcorr – “Wordcorr is a tool for making phonological comparisons among speech varieties. It can handle a large number of matching word lists. It provides for full IPA as well as other notations. Once the raw data are organized, Wordcorr keeps track of your judgments about what forms match each other and how they match. It tabulates your judgments into correspondence sets, the building blocks for a detailed analysis of the history of the speech varieties under study. (That's where the "corr" in the name comes from -- correspondence sets.)” (<http://wordcorr.org/> 3/27/2005)
10. Childes – “Child Language Data Exchange System. The CHILDES system provides tools for studying conversational interactions. These tools include a database of transcripts, programs for computer analysis of transcripts, methods for linguistic coding, and systems for linking transcripts to digitized audio and video.” (<http://chilides.psy.cmu.edu/> 4/20/2005)
11. Adobe Premiere – “real-time video and audio editing tools.” (<http://www.adobe.com/products/premiere/main.html> 4/20/2005)

12. StatView – discontinued as of December 31, 2002 (<http://www.statview.com/> 4/24). “Statview is a general purpose statistical package, fairly easy to use, available for macintosh and windows 95.” (<http://www.ucs.ed.ac.uk/usd/stats/statview.html> 4/23/2005)
13. Excel – software for spreadsheets, capable of doing calculations and producing charts. I have been unable to find a short and comprehensive summary of what Excel does on the web.
14. Word (95, 98 and later)
15. Photoshop – “editing, processing, and file handling.” (<http://www.adobe.com/products/photoshop/overview.html> 4/25/2005)
16. iMovie
17. Emacs – “Emacs is the extensible, customizable, self-documenting real-time display editor... a text editor and more. At its core is an interpreter for Emacs Lisp (“elisp”, for short), a dialect of the Lisp programming language with extensions to support text editing.” (<http://www.gnu.org/software/emacs/emacs.html> 4/25/2005)
18. Profiler
19. PlotFormants – “PlotFormants is a Macintosh program which generates two-dimensional plots of formant frequency data on linear, Mel and Bark scales.” (http://liceu.uab.es/~joaquim/speech_technology/tecnol_parla/software/software.html 4/25/2005)
19. LaTeX – “typesetting system, with features designed for the production of technical and scientific documentation.” (<http://www.latex-project.org/> 4/25/2005)
20. Prolog – “SWI-Prolog offers a comprehensive Free Software Prolog environment, licensed under the Lesser GNU Public License. Together with its graphics toolkit XPCE, its development started in 1987 and has been driven by the needs for real-world applications.” (<http://www.swi-prolog.org/> 4/25/2005)
21. DCG
22. UltraXML – “UltraXML is a native XML WYSIWYG authoring and publishing solution with an integrated high end composition engine that utilizes XSLT technology to enable the formatting and publishing of XML or SGML data to the Web, PDFs, Postscript and others.” (<http://www.xml.com/pub/p/6> 4/25/2005)
23. Uniscribe – “Uniscribe provides a large set of APIs to handle all aspects of text layout for supported scripts, including cursor position and hit testing, advance width calculation, linebreaking, complex script determination, localized digit substitution, etc..” (<http://www.microsoft.com/typography/developers/uniscribe/default.htm> 4/25/2005)
24. Open Office Writer – word processor and desktop publisher (<http://www.openoffice.org/product2/writer.html> 4/25/2005)
25. CanIPA – by Canepari

26. Knorpora – “Knorpora is a modified version of the Knoppix 3.3 Live CD for students of corpus-based computational linguistics. Like Knoppix, the Knorpora CD allows you to run a fully operational Debian/Linux operating system from the CD-ROM drive, without installing anything on the computer. The Knorpora edition of Knoppix contains programs and data files that should be of interest to computational linguistics students (WordNet, the Natural Language Toolkit, taggers, etc.)” (http://sslmit.unibo.it/~baroni/welcome_to_knorpora.html 4/25/2005)
27. JabRef – “JabRef is a graphical application for managing bibliographical databases. The main focus is on BibTeX databases, but other database formats will be supported as well. JabRef runs on all platforms and requires java 1.4.” (<http://sourceforge.net/projects/jabref/> 4/25/2005)
28. Kile – “Kile is a user friendly TeX/LaTeX editor for the KDE desktop environment. KDE is available for many architectures such as PC, PowerPC (Mac for example) and SPARC.” (<http://kile.sourceforge.net/> 4/25/2005)
29. Texniccenter – “TeXnicCenter is a feature rich integrated development environment (IDE) for developing LaTeX-documents on Microsoft Windows (Windows 9x/ME, NT/2000/XP) freely available under GPL.” (http://www.toolscenter.org/front_content.php?idcatart=41&lang=4&client=3 4/25/2005)
30. Perl – “Perl is a language optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information. It's also a good language for many system management tasks. The language is intended to be practical (easy to use, efficient, complete) rather than beautiful (tiny, elegant, minimal). Perl combines...some of the best features of C, sed, awk, and sh, so people familiar with those languages should have little difficulty with it. ...Unlike most Unix utilities, Perl does not arbitrarily limit the size of your data--if you've got the memory, Perl can slurp in your whole file as a single string. Recursion is of unlimited depth. And the tables used by ...associative arrays...grow as necessary to prevent degraded performance. Perl can use sophisticated pattern matching techniques to scan large amounts of data quickly. Although optimized for scanning text, Perl can also deal with binary data, and can make dbm files look like hashes. Setuid Perl scripts are safer than C programs through a dataflow tracing mechanism that prevents many stupid security holes.” (<http://search.cpan.org/dist/perl/pod/perl.pod> 4/25/2005)
31. MS Paint –painting program (<http://www.fayette.k12.il.us/99/paint/paint.htm> 4/25/2005)
32. Elan – “ELAN (EUDICO Linguistic Annotator) is an annotation tool that allows you to create, edit, visualize and search annotations for video and audio data. It was developed at the Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands, with the aim to provide a sound technological basis for the annotation and exploitation of multi-media recordings. ELAN is specifically designed for the analysis of language, sign language, and gesture, but it can be used by everybody who works with media corpora, i.e., with video and/or audio data, for purposes of annotation, analysis and documentation.” (http://liceu.uab.es/~joaquim/phonetics/fon_anal_acus/herram_anal_acus.html#ELAN 4/25/2005)
33. ImdI – A framework used for metadata, more complex than OLAC (<http://www.mpi.nl/IMDI/> (4/25/2005)
34. Sound Studio – “Sound Studio is a Mac OS X audio application that lets you record and edit two-channel audio.” (<http://www.feltp.com/products/soundstudio/> 4/25/2005)
35. Cascadilla Press Mora font

36. SIL Phonetics keyboard for Mac – “The “IPA-SIL” keyboard, developed by Fraser Bennett, is an installable keyboard layout for Mac OS X 10.2.x and above. It is intended to provide a text input method for Unicode-based applications (including most Cocoa-based applications, as well as some others), in order to access IPA characters.”
(http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&item_id=ipa-sil_keyboard 4/25/2005)
37. FileMaker – database software
38. Windows
39. SAL
40. TECKit – “TECKit is a low-level toolkit intended to be used by other applications that need to perform encoding conversions (e.g., when importing legacy data into a Unicode-based application). The primary component of the TECKit package is therefore a library that performs conversions; this is the "TECKit engine". The engine relies on mapping tables in a specific binary format (for which documentation is available); there is a compiler that creates such tables from a human-readable mapping description (a simple text file).”
(http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&cat_id=TECKit 4/25/2005)
41. XeTeX – “a typesetting system based on a merger of Donald Knuth's TeX system with Unicode and Mac OS X font technologies.” (http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=xetex 4/25/2005)
42. iWork application ‘Pages’ on Mac – word processor, “lets you create everything from letters to newsletters to brochures.” (<http://www.apple.com/iwork/> 4/25/2005)
43. Microsoft Publisher – publishing software
(<http://www.microsoft.com/office/publisher/prodinfo/overview.msp> 4/25/2005)
44. SIL Graphite font technology – “Graphite is a project under development within SIL’s Non-Roman Script Initiative and Language Software Development groups to provide rendering capabilities for complex non-Roman writing systems. (Work began in 1997 under the code name WinRend.) The original goal was a system for the Windows platform; a port to Linux is currently underway. Graphite can be used to create "smart fonts" capable of displaying writing systems with various complex behaviors. With respect to the Text Encoding Model, Graphite handles the "Rendering" aspect of writing system implementation.”
(http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&cat_id=RenderingGraphite 4/25/2005)
45. Font Creator Program 4.0 – “The Font Creator Program puts font creation within the grasp of the average PC user, would-be typographers, and graphic designers. With this application you can create and edit TrueType font files. You can use the modified fonts in Windows 3.11 and higher. Features include the ability to convert (scanned) bitmaps (BMP files) to TrueType outlines, thus enabling you to create your own signature, logo, and handwriting. In addition, unlimited undo and redo and repeat options are available. The modified fonts can be saved and then used in popular word processing and illustration programs. The editor lets you easily select any installed font, displaying the entire character set complete with descriptions of every letter, number, and special character. Once you've perfected your work of art, a click on the program's toolbar will install your creation.”
(http://www.softwareeshop.com/Design_Tools/Font_Creator_Program_40.htm 4/25/2005)

46. Mozilla Editor – “The Mozilla editor provides HTML and plain text editing functionality in Gecko-based applications. Internally, the editor code manipulates page content primarily using DOM calls, so it is a true DOM-savvy editor. It also supports IME (international text input), and is mostly accessible from JavaScript.” (http://www.mozilla.org/editor/#What_is_the_editor 4/25/2005)
47. CoolEdit96 – “Syntrillum Software Corporation's CoolEdit96™ graphical voice editor is included with Natural Access. CoolEdit96, which runs on Microsoft Windows 95 and Windows NT, simplifies prompt creation by allowing developers to graphically trim silence, cut-and-paste audio samples, and filter noise. CoolEdit96 also simplifies voice file conversion with support for NMS voice coding formats as well as OKI, .WAV, PCM, and other industry-standard formats.” (<http://www.telogic.com.sg/NMS-NaturalLinux.html> 4/25/2005)
48. MS Office
49. Star Office – “based on open standards. It offers word processing, spreadsheet, presentation, drawing, and database capabilities.” (<http://www.sun.com/software/star/staroffice/index.xml> 4/25/2005)
50. Transcriber – “Transcriber is a tool for assisting the manual annotation of speech signals. It provides a user-friendly graphical user interface for segmenting long duration speech recordings, transcribing them, and labeling speech turns, topic changes and acoustic conditions. It is more specifically designed for the annotation of broadcast news recordings, for creating corpora used in the development of automatic broadcast news transcription systems, but its features might be found useful in other areas of speech research.” (<http://www.etca.fr/CTA/gip/Projets/Transcriber/> 4/25/2005)
51. Phono – “Phono is a linguistic software tool for designing and testing models of regular historical sound change (diachronic phonology). Version 4.1-VB (for Windows, written in Visual Basic) is now in the beta-testing stage.” (<http://mypage.siu.edu/lhartman/> 4/25/2005)

Appendix II: Surveys

A. Department-wide Survey

Aloha. My name is Burgel R. Maria Faehndrich, and I am a Ph.D. student in the Department of Linguistics at UH Manoa. I am doing a research project about linguists and their interaction with computer programs, specifically those designed for linguists. I am doing this project for a LING 699 course about linguists and linguistics software. The product of this survey will be a document describing the current situation in linguist-computer interaction, which may be published at some point in the future. The document will also contain recommendations for linguists regarding usage of computer program for linguistics, and recommendations for creators of software regarding what kinds of software are needed by linguists.

If you participate in this survey, your information will be kept anonymous. Answers to survey questions will be copied and pasted to a word document, and emails will be deleted after that. Participation in the survey is completely voluntary. By responding to this email, you agree that I may use the information provided by you (responses to the questions below) in my research project.

If you have any questions, you can contact me at:

faehndri@hawaii.edu

Moore Hall 566B

Phone: 956-3231

If you have any questions regarding your rights as a research participant, please contact the UH Committee on Human Studies at (808) 956-5007.

QUESTIONNAIRE:

You do not have to answer all the questions. Just answer those which are relevant to you.

1. What computer programs for linguistics have you been using that you really liked?
2. What did you like about these computer programs?
3. What computer programs for linguistics have you been using that you disliked?
4. What did you dislike about them?
5. How do you think these programs could be improved?
6. What kinds of computer programs for linguistics do you think should be developed?
7. What do you think is the major problem with computer programs for linguists?
8. Any other comments:

Please reply by Sunday, February 27, 2005. Thank you for your assistance.

B. Linguist List Survey

Linguist - Software Interaction Survey

Aloha. My name is Burgel R. Maria Faehndrich, and I am a Ph.D. student in the Department of Linguistics at the University of Hawaii at Manoa. I am doing a research project about linguists and their interaction with computer programs, specifically those designed for linguists. I am doing this project for a LING 699 course about linguists and linguistics software. The product of this survey will be a document describing the current situation in linguist-computer interaction, which may be published at some point in the future. The document will also contain recommendations for linguists regarding usage of computer program for linguistics, and recommendations for creators of software regarding what kinds of software are needed by linguists.

If you participate in this survey, your information will be kept anonymous. Answers to survey questions will be copied and pasted to a word document, and information linking the respondents with their responses will be deleted after that. Participation in the survey is completely voluntary. By responding to this survey, you agree that I may use the information provided by you (responses to the questions below) in my research project. This project has received the University's Internal Review Board approval.

If you have any questions, you can contact me at:

faehndri@hawaii.edu

Moore Hall 566B

Phone: 956-3231

If you have any questions regarding your rights as a research participant, please contact the University of Hawaii Committee on Human Studies at (808) 956-5007.

QUESTIONNAIRE:

You do not have to answer all the questions. Just answer those which are relevant to you. In this questionnaire, “software” refers to any software you have use for any type of linguistics analysis, data management, data presentation, etc. It does not have to be software designed specifically for linguists.

1. What software have you been using that you really liked? What did you like about this software?
2. What software have you been using that you disliked? What did you dislike about it? How do you think this software could be improved?
3. What kinds of software for linguistics do you think should be developed?
4. What do you think is the most serious problem with software for linguists?
5. Any other comments:

If you would like to receive a copy of the final paper by email, please let me know. The paper will also be posted on at <http://www2.hawaii.edu/~faehndri/>

Please reply by Saturday, April 23, 2005. Thank you for your assistance.