Introducing *Phon*: A Software Solution for the Study of Phonological Acquisition

Yvan Rose[1], Brian MacWhinney[2], Rodrigue Byrne[1], Gregory Hedlund[1], Keith Maddocks[1], Philip O'Brien[3] and Todd Wareham[1]
[1]Memorial University of Newfoundland, [2]Carnegie Mellon University, [3]Dalhousie University

1. Introduction

In this paper, we introduce a new software program, *Phon*, which aims at providing unprecedented functionality for the elaboration, analysis and sharing of phonological data. *Phon* is part of a larger initiative, *PhonBank*, the brainchild of the first two authors of this paper, which aims at extending the software and database components of the *CHILDES* project into the realm of phonology. *Phon* can be downloaded from the *CHILDES* web site at: http://childes.psy.cmu.edu/Phon/.

Phon is a new software program designed to support every step involved in the management, analysis and sharing of large corpora of phonological data. The foundation of this software was elaborated in the context of two Honor's B.Sc. theses (Hedlund and O'Brien, 2004; Maddocks, 2005). Hedlund and O'Brien (2004) provided outlines and specifications of the main components of *Phon*. Maddocks (2005) contributed algorithms for segmental and syllabic alignment. This algorithm is an important component of the software, because it enables systematic comparisons between target and actual speech productions from both segmental and prosodic perspectives (see further below). Since May 2004, Gregory Hedlund and Philip O'Brien have continued working on the design and

[*] Prior to its initial release during *BUCLD* 30, the design of *Phon* has been extensively discussed during meetings and conference presentation, held during the *Second Lisbon Meeting on Language Acquisition* (June 2004), *BUCLD* 29 (November 2004), the *Characteristics and Assessment of Stuttered Speech Workshop* held at the University College London (MacWhinney and Rose 2005) and the *International Congress for the Study of Child Language* in Berlin (Rose et al. 2005). A preliminary version was also tested by several colleagues and students internationally. While we cannot name every individual who contributed to this project, we would like to thank them for their generous and immensely useful contribution. This project was funded by the Social Sciences and Humanities Research Council of Canada, the National Science Foundation, the Arts Faculty and the Vice-President Research at Memorial University, the Natural Sciences and Engineering Research Council of Canada, the National Science Foundation and the Canada fund for Innovation.

implementation of additional components of *Phon* with ongoing supervision from Byrne, Wareham and Rose, in close collaboration with MacWhinney's *CHILDES*-TalkBank team of programmers at Carnegie Mellon University.

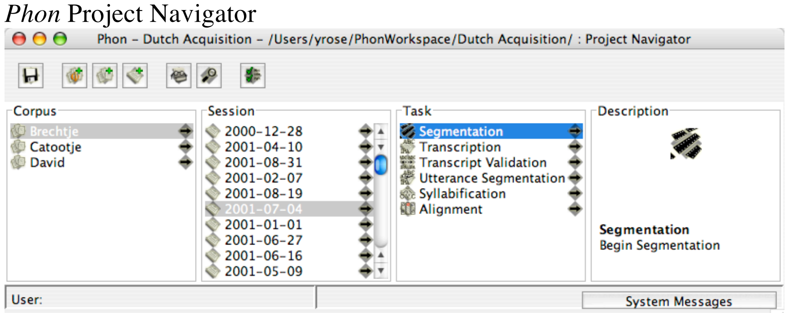
Despite the definite relationship between *Phon* and *PhonBank*, this paper will focus specifically on describing the current implementation of *Phon*, in section 2. We will also mention, in section 3, some future directions in the development of this project. We briefly conclude in section 4 with some of the potential impact that this software may have for the research community.

2. The *Phon* software

Phon is programmed in Java and runs on Macintosh, Windows, and Unix/Linux operating systems. The current version has been tested on *Mac* OS X 10.3 and 10.4, *Windows* 2000, *Windows* XP, and several forms of the *Linux* operating system. *Phon* uses Unicode font encoding, which enables 100% data compatibility across these various platforms, an important condition for data sharing among researchers. Moreover, because it relies on the *TalkBank* XML data encoding schema, *Phon* can interface directly with new *Java WebStart* programs for web-based access and analysis being built by the *TalkBank* and *CHILDES* projects.

Phon is organized into eight inter-related modules. These modules are accessible through the interface illustrated by the screen shot of the Project Navigator in (1). This interface provides hierarchical access to each corpus included in a research project (first column), then to each of the recording sessions included in any given corpus (second column). The list of available tasks, which may vary across users, is provided in the third column, alongside a task description and access button in the fourth column.

(1) *Phon* Project Navigator

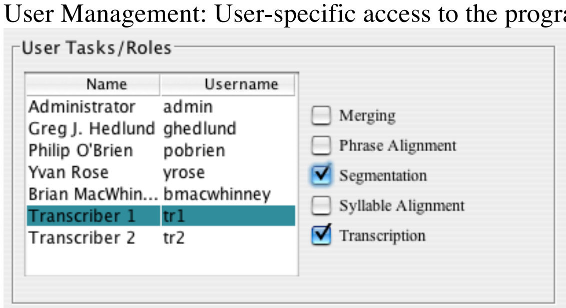


In the subsections below, we describe in order the functions provided by the main modules of *Phon*.

2.1. User Management

At the top level, the program allows project managers to configure password-protected access so that users will only access functionality that is appropriate for the work they need to perform. The user management module can also facilitate control of the workflow as well as incremental access to the various functions of the program. In (2) is the one of the screens in the management module. The user selected here, code-named Transcriber 1, can only access the Media Segmentation and Data Transcription modules (see below for a description of these modules).

(2) User Management: User-specific access to the program's functions

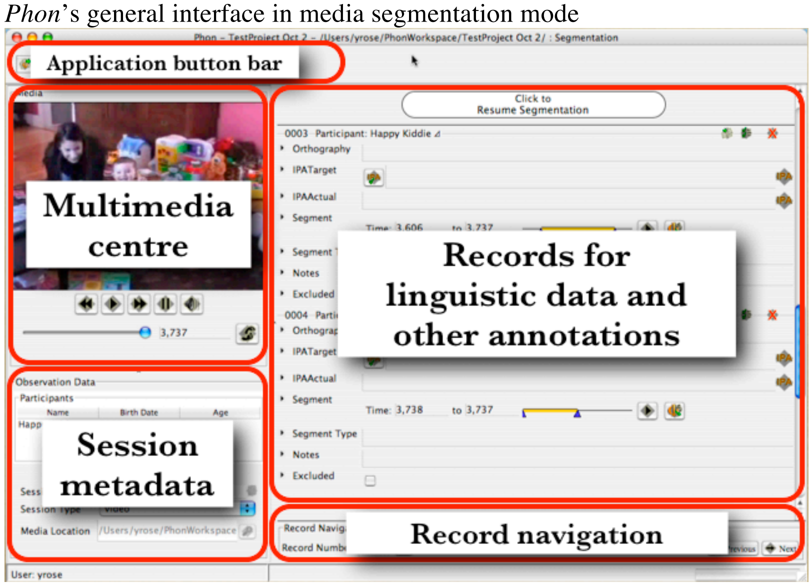


Users and access privileges can be added and removed at will by the project manager in order to enable easy and flexible management of each step in the project.

2.2. Media Segmentation

The second module focuses on media segmentation, the functionality required to identify the portions of the recorded media that are relevant for research. The techniques used here will be familiar to child language researchers, since they are modeled on those available in *CLAN*. The main distinction here is that, in line with the emphasis on usability that has driven the elaboration of this project, all functions are available directly from the graphical user interface. A sample screen of the *Phon* interface for segmentation and linkage of transcripts to a video file is given below.

(3) *Phon*'s general interface in media segmentation mode

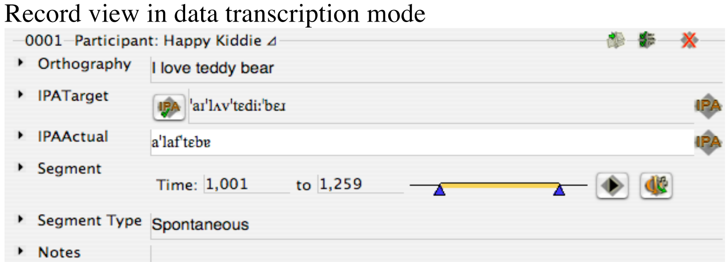


This screen shot also provides a general view of the program's graphical user interface. This interface provides simultaneous access to the recorded media, the recording session metadata (e.g. session date; list of participants and their ages at recording time). All audio and video formats compatible with QuickTime Player (for Macintosh or Windows) are supported.

2.3. Data Transcription

Each media segment identified through the module described above corresponds to a record in the database. By default, records contain fields for orthographic and phonetic transcriptions as well as segment type (e.g. spontaneous or repeated speech) and notes. The application also supports the creation of user-defined fields that can be used both for the creation of new corpora or for the importation of existing corpora. It thus offers a standardized data entry method as well as sufficient flexibility for data coding.

(4) Record view in data transcription mode



Phonetic transcriptions are based on the symbols and conventions of the International Phonetic Association (IPA). An IPA character map is accessible from within the application, to facilitate access to the symbols for which there is no keyboard equivalent. Using this input method, the user enters the target adult words if they are available (i.e. prelinguistic data will not include this feature). Then the citation form IPA pronunciations of these words are inserted from the CHILDES-formatted version of the CMU Pronouncing Dictionary; http://www.speech.cs.cmu.edu/cgi-bin/cmudict (or similar sources for other languages, as available). Once citation forms are accurately characterized, they provide a uniform baseline against which child-based patterns can be identified (e.g. patterns of consonant harmony or of syllable truncation).

Of course, idealized citation forms do not provide accurate fine-grained characterizations of variations in the target language (e.g. dialect-specific pronunciation variants; phonetic details such as degree of aspiration in obstruent stops). In order to capture these finer details, the source dictionaries must be modified to more accurately represent the actual patterns in the input. In some cases, it will be possible to make systematic modifications to sounds patterns in the source dictionaries to reflect global aspects of local dialect variation. This work can be done through text editing outside of *Phon*. In other cases, it will be necessary to create a set of alternative target forms that reflect different degrees of segmental and prosodic reduction in the adult input. It will also be necessary to represent reduction patterns by mergers across lexical items. Currently, *Phon* supports only one target form set. However, future versions will allow users to add additional possible adult target form sets and to apply sandhi, elision, and syncope rules to canonical forms to bring them into closer correspondence with the facts of adult input.

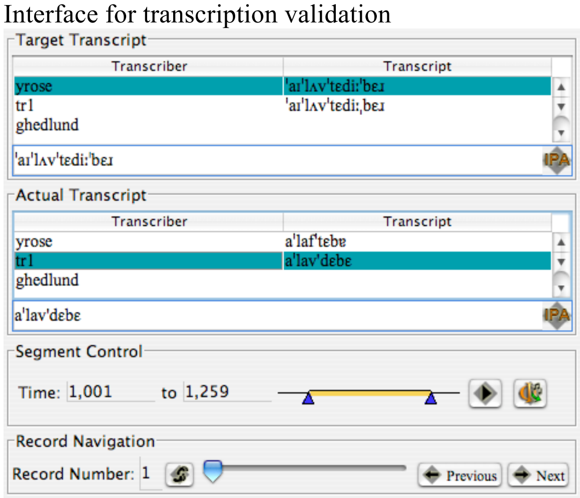
Actual forms produced by the learner must be transcribed manually. To facilitate this process, *Phon* provides direct access to the segmented portions of the media for playback in each record (see the 'Segment' tier in the above screen shot). The beginning and end times of these segments can also be edited directly from the record, which facilitates an accurate circumscription of the relevant portions of the recorded media. Finally, *Phon* can export the segmented portions of the media into a sound file, which enables quick acoustic verifications using sound visualizing software such as *Praat*, *Signalyze*, *WAVES*, *SFS*, or *CSL*.

Target and actual transcriptions are stored internally as strings of phonetic symbols. Each symbol is automatically associated with a set of descriptive features. These features are based on standards used in the fields of phonetics and phonology (e.g. bilabial, alveolar, voiced, voiceless, aspirated) (Ladefoged and Maddieson 1996). These features are extremely useful in the sense that they provide series of descriptive labels to each transcribed symbol. The availability of these labels is essential for research involving the grouping of various sounds into natural classes (e.g. the class of voiced consonants; the class of non-high vowels). The built-in set of features can also be reconfigured as needed to fit special research needs.

2.4. Transcription Validation

With controls specified through the User Management module, the program allows for multiple-blind transcription. Once the individual transcriptions are completed, the Transcription Validation module allows the transcription supervisor (or, in a better set up, a team of supervisors working together) to compare alternative transcriptions and resolve divergences.

(5) Interface for transcription validation

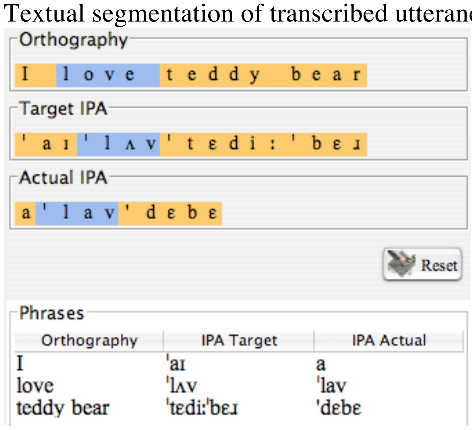


As illustrated in this screen shot, all of the transcriptions produced for any given record are displayed in parallel. Similar to the transcription field, the recorded media can be played from the module's interface or exported for examination in speech analysis software. The preferred transcriptions are then selected for further analysis. Alternative, non-validated transcriptions are preserved for data recoverability and verification purposes.

2.5. Utterance Segmentation

Researchers often wish to divide transcribed utterances into specific domains such as the phrase, the word, the root, the stem, or the affix. *Phon* fulfills this need by incorporating a text segmentation module that enables the identification of strings of symbols corresponding to such morpho-syntactic and phonological domains. For example, using the syllabification module described immediately below, the researcher can test hypotheses about what domains are relevant for resyllabification processes across words (e.g. whether resyllabification takes place between phrases within an utterance).

(6) Textual segmentation of transcribed utterance

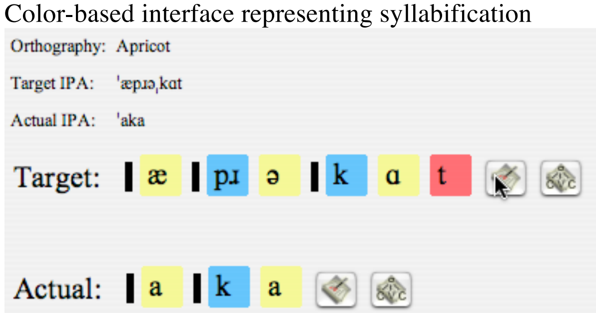


Given the wide variety of possibilities allowed by the Utterance Segmentation module, this task must be handled manually, according to the specific needs of the researcher.

2.6. Automatic Syllabification

Once the researcher has identified the domains of analysis, segmentation at the level of the syllable is performed automatically: Segments are assigned descriptive syllable labels (visually represented with colors) such as 'onset' or 'coda' for consonants and 'nucleus' for vowels. The program also identifies segmental sequences within syllable constituents (e.g. complex onsets or nuclei). Since controversy exists in both phonetic and phonological theory regarding guidelines for syllabification, the algorithm is parameterized to allow for analytical flexibility. The availability of different parameter settings also enables the researcher to test hypotheses on which analysis makes the best prediction for a given dataset.

(7) Color-based interface representing syllabification



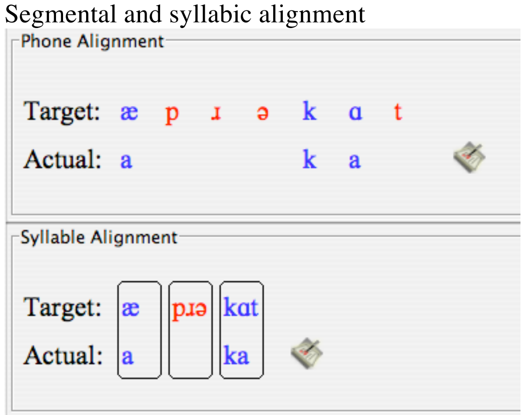
Occasionally, the algorithm may produce spurious results, especially when it encounters strings of segments that pose problems from a syllabification standpoint (e.g. long series of consonants with variable sonority profiles). After the operation of automatic syllabification, the user must go over the transcript to verify all results. Segments that are left unsyllabified will be available for all queries on segmental features and strings of segments, but will not be available for queries referring to aspects of syllabification.

The syllabification labels can then be used in database query (for example, to access specific information about syllable onsets or codas). In addition, because the algorithm is sensitive to main and secondary stress marks and domain edges (i.e. first and final syllables), each syllable identified is given a prosodic status and position index. Using the Database Query module (described below), the researcher can thus use search criteria as precisely defined as, for example, complex onsets realized in word-medial, secondary-stressed syllables. As discussed below, this level of functionality is central to the study of several phenomena in phonological acquisition that are determined by the status of the syllable as stressed or unstressed, or by the position of the syllable within the word.

2.7. Automatic Alignment

After syllabification, a second algorithm performs automatic, segment-by-segment and syllable-by-syllable alignment of target and actual forms. Building on featural similarities and differences between the segments in each syllable and on syllable properties such as stress, this algorithm automatically aligns corresponding segments and syllables in target and actual forms. This algorithm, whose design is described in (Maddocks 2005), is an extension of the method proposed by Kondrak (2003). It provides alignments for both corresponding sounds and syllables. For example, in the target-actual word pair 'apricot' > 'a_cot', the algorithm aligns the first and final syllables of each form, and identifies the middle syllable ('pri') as truncated (see screen shot).

(8) Segmental and syllabic alignment



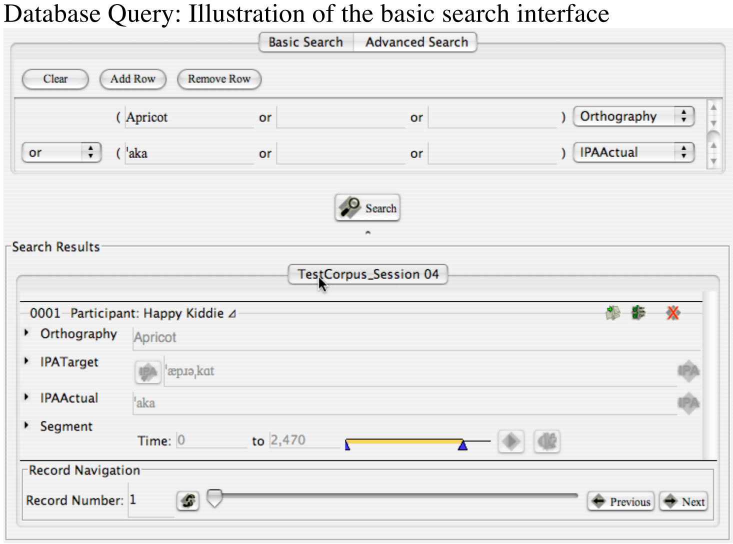
Similarly, in cases of renditions such as ‘blow’ > ‘bolow’ the alignment algorithm relates both syllables of the actual form to the only syllable of the target form and diagnoses a case of syllable (vowel) epenthesis. As documented in (Maddocks 2005) and Hedlund et al. (2005), preliminary tests on attested data from the published literature on Dutch- and English-learning children (e.g. Fikkert 1994; Pater 1997) indicate an accuracy rate above 95% (96% for the Dutch corpus and 98% for the English corpus). As it is the case with the other algorithms included in the program, the user is able to perform manual adjustments of the computer-generated syllable alignments whenever necessary. The alignment algorithm, as well as the data processing steps that precede it (especially, syllabification), are essential to any acquisition study that requires pair-wise comparisons between target and actual forms, from both segmental and syllabic perspectives.

Implicit to the description of the last two modules is a careful approach whereby the algorithms implemented at this stage are used to assist data compilation; because every result generated by the algorithms can be modified by the user, no data analysis directly depends on them. The user thus has complete control on the processing of the data being readied for analysis. After extensive testing on various types of data sets, we will be able to optimize their degree of reliability and then determined how they can be used in truly automated analyses.

2.8. Database Query

Once the user has completed validation of the segmental and syllabic alignment, the data are ready for analysis. Two levels of query are available in *Phon*. The first, Basic Search, is designed for text-based queries using regular expressions such as Boolean connectives and wildcards, which are accessible directly from the interface, as illustrated by the screen shot below.

(9) Database Query: Illustration of the basic search interface



The second, Advanced Search, is an extension of PhonBasic, a query language designed from the ground up (Hedlund and O'Brien 2004). Taking advantage of the Jython scripting language (*Jython* is a *Java* implementation of the *Python* scripting language; see http://www.jython.org/ for more details), the advanced search module enable extremely specific data queries that may, whenever needed, make reference to data generated by the syllabification and alignment algorithms described above.

PhonBasic indeed uses common terms such as "syllable" and "stress" and allows for queries at the level of segments, features, syllabic constituents, syllables, and higher levels of prosodic organization. Using the query language, the user can extract phonological information about child and target forms. These inventories can be used to attain a characterization of productive abilities within and across developmental stages. Pair-wise comparisons between target and actual forms through specific queries enable a quick and accurate detection of the patterns found in the acquisition data such as syllable truncation (e.g. 'banana' > 'nana'), branching onset reduction (e.g. 'train' > 'tain'), consonant harmony (e.g. 'duck' > 'guck') or velar fronting (e.g. 'coffee' > 'toffee'). Because the stress and syllabic information encoded in the data is available in the query language, these patterns can also be detected relative to positions within the string (e.g. in stressed versus unstressed syllables; in word-initial, medial and final positions), within the syllable (e.g. onset versus coda position), and across syllables.

3. Future directions

The application described above offers a strong basis for computer-assisted research in phonology and phonological development. However, this tool, in its current implementation, should by no means be considered a complete solution. Several further developments are planned, some of which are outlined below. First, we will implement an automatic algorithm for adjusting the parameters of the alignment module described above. Based on user-provided examples of segmental and syllabic alignment patterns, this system automatically adjusts the parameters of the alignment algorithm in order to yield the highest possible degree of accuracy. As a result, the more alignments are validated by the user, the higher the degree of overall accuracy of the algorithm. The preliminary version of this optimization system is described in Maddocks (2005) and Hedlund et al. (2005) but remains to be implemented in *Phon*.

Second, we will design an interface for entry of acoustic data measurements obtained with speech analysis software such as *Praat*. While the current design does not incorporate functions for acoustic measurements per se, it allows for the importing of values extracted with speech analysis software into the *Phon* database structure. Using this module, the researcher will be able to systematic assess the acoustic behaviors of speech sounds in precise phonological contexts.

Third, we plan to integrate basic statistical and data reporting functions, in order to facilitate data extraction from large corpora of data. These functions will be implemented using the statistical packages provided in Java software development kits.

In the longer run, we plan to implement additional automated analytical functions, with the aim of making *Phon* a useful report and diagnosis tool for use in developmental and clinical research.

While pursuing the goals overviewed above, we will also provide functionality and interface improvements to streamline the overall usability of the tool based on user feedback.

4. Conclusion

As discussed in the previous sections, *Phon*, once completed, will offer several innovative characteristics enabling researchers to investigate theoretical, clinical and educational issues using more advanced investigation methods. Below is a summary of *Phon*'s most prominent innovative characteristics, which include: Integrated methods for workflow, user access and multiple-blind data transcription and validation; algorithms for data syllabification and for target-actual (segment and syllable) alignment; method for crossing phonological with phonetic data; query functions tailored for extracting phonological and phonetic data; cross-platform compatibility.

The more advanced research methods enabled by a full realization of this software development project will offer a basis for important advances in all theoretical and applied fields of research that relate to the sound systems of natural languages. Once completed, the current development phase will support advances in research in language development documentation and archival (especially, *CHILDES*-related *PhonBank* database project), phonetics (e.g. speech production; speech recognition), theoretical and applied phonology, language development, clinical linguistics (establishment of normative baselines, better understanding of developmental and acquired speech disorders) and education (both official and minority languages). The empirical and theoretical benefits of this project also have spinoffs in other speech-related areas such as genetic research in speech disorders as well as research in artificial intelligence and language learning.

References

Fikkert, Paula (1994). *On the Acquisition of Prosodic Structure*. Dordrecht: ICG Printing.

Hedlund, Gregory, Keith Maddocks, Yvan Rose and Todd Wareham (2005). Pairwise Natural Language Syllabic Alignment: From Conception to Implementation. In *Proceedings of the Fifteenth Annual Newfoundland Electrical and Computer Engineering Conference* (NECEC 2005). To appear.

Hedlund, Gregory and Philip O'Brien (2004). A Software System for Linguistic Data Capture and Analysis. B.Sc. Honours Thesis. Memorial University of Newfoundland.

Ladefoged, Peter and Ian Maddieson (1996). *The Sounds of the World's Languages*. Cambridge, MA: Blackwell.

Maddocks, Keith (2005). An Effective Algorithm for the Alignment of Target and Actual Syllables for the Study of Language Acquisition. B.Sc. Honours Thesis. Memorial University of Newfoundland.

Pater, Joe (1997). Minimal Violation and Phonological Development. *Language Acquisition* 6(3):201-253.

Rose, Yvan and Brian MacWhinney (2005). The *Phon* and PhonBank Initiatives: A Scheme for Sharing and Archiving Child Language Phonology Data. Invited presentation. *The Characteristics and Assessment of Stuttered Speech*. University College London. June.

Rose, Yvan, Brian MacWhinney, Rodrigue Byrne, Gregory Hedlund, Keith Maddocks, Philip O'Brien and Todd Wareham (2005a). *Phon*: A Computerized Database for Research in Phonological Acquisition. *10th International Congress for the Study of Child Language*. Berlin, July.