Tools for Analyzing Talk

Part 1: The CHAT Transcription Format

Brian MacWhinney
Carnegie Mellon University

July 31, 2019
https://doi.org/10.21415/3mhn-0z89

When citing the use of TalkBank and CHILDES facilities, please use this reference to the last printed version of the CHILDES manual:

MacWhinney, B. (2000). The CHILDES Project: Tools for Analyzing Talk. 3rd Edition. Mahwah, NJ: Lawrence Erlbaum Associates

This allows us to systematically track usage of the programs and data through scholar.google.com.

1 Introduction

This electronic edition of the CHAT manual is being continually revised to keep pace with the growing interests of the language research communities served by the TalkBank and CHILDES communities. The first three editions were published in 1990, 1995, and 2000 by Lawrence Erlbaum Associates. After 2000, we switched to the current electronic publication format. However, in order to easily track usage through systems such as Google Scholar, we ask that users cite the version of the manual published in 2000, when using data and programs in their published work. This is the citation: MacWhinney, B. (2000). *The CHILDES project: Tools for analyzing talk. 3rd edition*. Mahwah, NJ: Lawrence Erlbaum Associates.

In its earlier version, this manual focused exclusively on the use of the programs for child language data in the context of the CHILDES system (https://childes.talkbank.org). However, beginning in 2001 with support from NSF, we introduced the concept of TalkBank (https://talkbank.org)to include a wide variety of language databases. These now include:

1. AphasiaBank (https://aphasia.talkbank.org) for language in aphasia,
2. ASD Bank (https://asd.talkbank.org) for language in autism,
3. BilingBank (https://biling.talkbank.org) for the study of bilingualism and code-switching,
4. CABank (https://ca.talkbank.org) for Conversation Analysis, including the large SCOTUS corpus,
5. CHILDES (https://childes.talkbank.org) for child language acquisition,
6. ClassBank (https://class.talkbank.org) for studies of language in the classroom,
7. DementiaBank (https://dementia.talkbank.org) for language in dementia,
8. FluencyBank(https://fluency.talkbank.org) for the study of childhood fluency development,
9. HomeBank (https://homebank.talkbank.org) for daylong recordings in the home,
10. PhonBank (https://phonbank.talkbank.org) for the study of phonological development,
11. RHDBank (https://rhd.talkbank.org) for language in right hemisphere damage,
12. SamtaleBank (https://samtalebank.talkbank.org) for Danish conversations.
13. SLABank (https://slabank.talkbank.org) for second language acquisition, and
14. TBIBank (https://tbi.talkbank.org) for language in traumatic brain injury,

The current manual maintains some of the earlier emphasis on child language, particularly in the first sections, while extending the treatment to these further areas and formats in terms of new codes and several new sections. We are continually adding corpora to each of these separate collections. In 2018, the size of the text database is 800MB and there is an additional 5TB of media. All of the data in TalkBank are freely open to downloading and analysis with the exception of the data in the clinical language banks which are open to clinical researchers using passwords. The CLAN program and the related morphosyntactic taggers are all free and open-sourced through GitHub.

Fortunately, all of these different language banks make use of the same transcription format (CHAT) and the same set of programs (CLAN). This means that, although most

of the examples in this manual rely on data from the CHILDES database, the principles extend easily to data in all of the TalkBank repositories. TalkBank is the largest open repository of data on spoken language. All of the data in TalkBank are transcribed in the CHAT format which is compatible with the CLAN programs.

Using conversion programs available inside CLAN (see the CLAN manual for details), transcripts in CHAT format can be automatically converted into the formats required for Praat (praat.org), Phon (phonbank.talkbank.org), ELAN (tla.mpi.nl/tools/elan), CoNLL, ANVIL (anvil-software.org), EXMARaLDA (exmaralda.org), LIPP (ihsys.com), SALT (saltsoftware.com), LENA (lenafoundation.org), Transcriber (trans.sourceforge.net), and ANNIS (corpus-tools.org/ANNIS).

TalkBank databases and programs have been used widely in the research literature. CHILDES, which is the oldest and most widely recognized of these databases, has been used in over 7000 published articles. PhonBank has been used in 480 articles and AphasiaBank has been used in 212 presentations and publications. In general, the longer a database has been available to researchers, the more the use of that database has become integrated into the basic research methodology and publication history of the field.

Metadata for the transcripts and media in these various TalkBank databases have been entered into the two major systems for accessing linguistic data: OLAC, and VLO (Virtual Language Observatory). Each transcript and media file has been assigned a PID (permanent ID) using the Handle System (www.handle.net), and each corpus has received an ISBN and DOI (digital object identifier) number.

For ten of the languages in the database, we provide automatic morphosyntactic analysis using a series of programs built into CLAN. These languages are Cantonese, Chinese, Dutch, English, French, German, Hebrew, Japanese, Italian, and Spanish. The codes produced by these programs could eventually be harmonized with the GOLD ontology. In addition, we can compute a dependency grammar analysis for each of these 10 languages. As a result of these efforts, TalkBank has been recognized as a Center in the CLARIN network (clarin.eu) and has received the Data Seal of Approval (datasealofapproval.org). TalkBank data have also been included in the SketchEngine corpus tool (sketchengine.co.uk).

2 The CHILDES Project

Language acquisition research thrives on data collected from spontaneous interactions in naturally occurring situations. You can turn on a tape recorder or videotape, and, before you know it, you will have accumulated a library of dozens or even hundreds of hours of naturalistic interactions. But simply collecting data is only the beginning of a much larger task, because the process of transcribing and analyzing naturalistic samples is extremely time-consuming and often unreliable. In this first volume, we will present a set of computational tools designed to increase the reliability of transcriptions, automate the process of data analysis, and facilitate the sharing of transcript data. These new computational tools have brought about revolutionary changes in the way that research is conducted in the child language field. In addition, they have equally revolutionary potential for the study of second-language learning, adult conversational interactions, sociological content analyses, and language recovery in aphasia. Although the tools are of wide applicability, this volume concentrates on their use in the child language field, in the hope that researchers from other areas can make the necessary analogies to their own topics.

Before turning to a detailed examination of the current system, it may be helpful to take a brief historical tour over some of the major highlights of earlier approaches to the collection of data on language acquisition. These earlier approaches can be grouped into five major historical periods.

2.1 Impressionistic Observation

The first attempt to understand the process of language development appears in a remarkable passage from *The Confessions of St. Augustine* (1952). In this passage, Augustine claims that he remembered how he had learned language:

> This I remember; and have since observed how I learned to speak. It was not that my elders taught me words (as, soon after, other learning) in any set method; but I, longing by cries and broken accents and various motions of my limbs to express my thoughts, that so I might have my will, and yet unable to express all I willed or to whom I willed, did myself, by the understanding which Thou, my God, gavest me, practise the sounds in my memory. When they named anything, and as they spoke turned towards it, I saw and remembered that they called what they would point out by the name they uttered. And that they meant this thing, and no other, was plain from the motion of their body, the natural language, as it were, of all nations, expressed by the countenance, glances of the eye, gestures of the limbs, and tones of the voice, indicating the affections of the mind as it pursues, possesses, rejects, or shuns. And thus by constantly hearing words, as they occurred in various sentences, I collected gradually for what they stood; and, having broken in my mouth to these signs, I thereby gave utterance to my will. Thus I exchanged with those about me these current signs of our wills, and so launched deeper into the stormy intercourse of human life, yet depending on parental authority and the beck of elders.

Augustine's outline of early word learning drew attention to the role of gaze, pointing, intonation, and mutual understanding as fundamental cues to language learning. Modern research in word learning (Bloom, 2000) has supported every point of Augustine's analysis, as well as his emphasis on the role of children's intentions. In this sense, Augustine's somewhat fanciful recollection of his own language acquisition remained the high water mark for child language studies through the Middle Ages and even the Enlightenment. Unfortunately, the method on which these insights were grounded depends on our ability to actually recall the events of early childhood – a gift granted to very few of us.

2.2 Baby Biographies

Charles Darwin provided much of the inspiration for the development of the second major technique for the study of language acquisition. Using note cards and field books to track the distribution of hundreds of species and subspecies in places like the Galapagos and Indonesia, Darwin was able to collect an impressive body of naturalistic data in support of his views on natural selection and evolution. In his study of gestural development in his son, Darwin (1877) showed how these same tools for naturalistic observation could be adopted to the study of human development. By taking detailed daily notes, Darwin showed how researchers could build diaries that could then be converted into biographies documenting virtually any aspect of human development. Following Darwin's lead, scholars such as Ament (1899), Preyer (1882), Gvozdev (1949), Szuman (1955), Stern & Stern (1907), Kenyeres (Kenyeres, 1926, 1938), and Leopold (1939, 1947, 1949a, 1949b) created monumental biographies detailing the language development of their own children.

Darwin's biographical technique also had its effects on the study of adult aphasia. Following in this tradition, studies of the language of particular patients and syndromes were presented by Low (1931) , Pick (1913), Wernicke (1874), and many others.

2.3 Transcripts

The limits of the diary technique were always quite apparent. Even the most highly trained observer could not keep pace with the rapid flow of normal speech production. Anyone who has attempted to follow a child about with a pen and a notebook soon realizes how much detail is missed and how the note-taking process interferes with the ongoing interactions.

The introduction of the tape recorder in the late 1950s provided a way around these limitations and ushered in the third period of observational studies. The effect of the tape recorder on the field of language acquisition was very much like its effect on ethnomusicology, where researchers such as Alan Lomax (Parrish, 1996) were suddenly able to produce high quality field recordings using this new technology. This period was characterized by projects in which groups of investigators collected large data sets of tape recordings from several subjects across a period of 2 or 3 years. Much of the excitement in the 1960s regarding new directions in child language research was fueled directly by the great increase in raw data that was possible through use of tape recordings and typed transcripts.

This increase in the amount of raw data had an additional, seldom discussed, consequence. In the period of the baby biography, the final published accounts closely

resembled the original database of note cards. In this sense, there was no major gap between the observational database and the published database. In the period of typed transcripts, a wider gap emerged. The size of the transcripts produced in the 60s and 70s made it impossible to publish the full corpora. Instead, researchers were forced to publish only high-level analyses based on data that were not available to others. This led to a situation in which the raw empirical database for the field was kept only in private stocks, unavailable for general public examination. Comments and tallies were written into the margins of ditto master copies and new, even less legible copies, were then made by thermal production of new ditto masters. Each investigator devised a project-specific system of transcription and project-specific codes. As we began to compare hand-written and typewritten transcripts, problems in transcription methodology, coding schemes, and cross-investigator reliability became more apparent.

Recognizing this problem, Roger Brown took the lead in attempting to share his transcripts from Adam, Eve, and Sarah (Brown, 1973) with other researchers. These transcripts were typed onto stencils and mimeographed in multiple copies. The extra copies were lent to and analyzed by a wide variety of researchers. In this model, researchers took their copy of the transcript home, developed their own coding scheme, applied it (usually by making pencil markings directly on the transcript), wrote a paper about the results and, if very polite, sent a copy to Roger. Some of these reports (Moerk, 1983) even attempted to disprove the conclusions drawn from those data by Brown himself!

During this early period, the relations between the various coding schemes often remained shrouded in mystery. A fortunate consequence of the unstable nature of coding systems was that researchers were very careful not to throw away their original data, even after it had been coded. Brown himself commented on the impending transition to computers in this passage (Brown, 1973, p. 53):

> It is sensible to ask and we were often asked, "Why not code the sentences for grammatically significant features and put them on a computer so that studies could readily be made by anyone?" My answer always was that I was continually discovering new kinds of information that could be mined from a transcription of conversation and never felt that I knew what the full coding should be. This was certainly the case and indeed it can be said that in the entire decade since 1962 investigators have continued to hit upon new ways of inferring grammatical and semantic knowledge or competence from free conversation. But, for myself, I must, in candor, add that there was also a factor of research style. I have little patience with prolonged "tooling up" for research. I always want to get started. A better scientist would probably have done more planning and used the computer. He can do so today, in any case, with considerable confidence that he knows what to code.

With the experience of three more decades of computerized analysis behind us, we now know that the idea of reducing child language data to a set of codes and then throwing away the original data is simply wrong. Instead, our goal must be to computerize the data in a way that allows us to continually enhance it with new codes and annotations. It is fortunate that Brown preserved his transcript data in a form that

allowed us to continue to work on it. It is unfortunate, however, that the original audiotapes were not kept.

2.4 Computers

Just as these data analysis problems were coming to light, a major technological opportunity was emerging in the shape of the powerful, affordable microcomputer. Microcomputer word-processing systems and database programs allowed researchers to enter transcript data into computer files that could then be easily duplicated, edited, and analyzed by standard data-processing techniques. In 1981, when the Child Language Data Exchange System (CHILDES) Project was first conceived, researchers basically thought of computer systems as large notepads. Although researchers were aware of the ways in which databases could be searched and tabulated, the full analytic and comparative power of the computer systems themselves was not yet fully understood.

Rather than serving only as an "archive" or historical record, a focus on a shared database can lead to advances in methodology and theory. However, to achieve these additional advances, researchers first needed to move beyond the idea of a simple data repository. At first, the possibility of utilizing shared transcription formats, shared codes, and shared analysis programs shone only as a faint glimmer on the horizon, against the fog and gloom of handwritten tallies, fuzzy dittos, and idiosyncratic coding schemes. Slowly, against this backdrop, the idea of a computerized data exchange system began to emerge. It was against this conceptual background that CHILDES (the name uses a one-syllable pronunciation) was conceived. The origin of the system can be traced back to the summer of 1981 when Dan Slobin, Willem Levelt, Susan Ervin-Tripp, and Brian MacWhinney discussed the possibility of creating an archive for typed, handwritten, and computerized transcripts to be located at the Max-Planck-Institut für Psycholinguistik in Nijmegen. In 1983, the MacArthur Foundation funded meetings of developmental researchers in which Elizabeth Bates, Brian MacWhinney, Catherine Snow, and other child language researchers discussed the possibility of soliciting MacArthur funds to support a data exchange system. In January of 1984, the MacArthur Foundation awarded a two-year grant to Brian MacWhinney and Catherine Snow for the establishment of the Child Language Data Exchange System. These funds provided for the entry of data into the system and for the convening of a meeting of an advisory board. Twenty child language researchers met for three days in Concord, Massachusetts and agreed on a basic framework for the CHILDES system, which Catherine Snow and Brian MacWhinney would then proceed to implement.

2.5 Connectivity

Since 1984, when the CHILDES Project began in earnest, the world of computers has gone through a series of remarkable revolutions, each introducing new opportunities and challenges. The processing power of the home computer now dwarfs the power of the mainframe of the 1980s; new machines are now shipped with built-in audiovisual capabilities; and devices such as CD-ROMs and optical disks offer enormous storage capacity at reasonable prices. This new hardware has now opened up the possibility for multimedia access to digitized audio and video from links inside the written transcripts. In effect, a transcript is now the starting point for a new exploratory reality in which the whole interaction is accessible from the transcript. Although researchers have just now

begun to make use of these new tools, the current shape of the CHILDES system reflects many of these new realities. In the pages that follow, you will learn about how we are using this new technology to provide rapid access to the database and to permit the linkage of transcripts to digitized audio and video records, even over the Internet.

3 From CHILDES to TalkBank

Beginning in 2001, with support from an NSF Infrastructure grant, we began the extension of the CHILDES database concept to a series of additional fields listed in the Introduction. These extensions have led to the need for additional features in the CHAT coding system to support CA notation, phonological analysis, and gesture coding. As we develop new tools for each of these areas and increase the interoperability between tools, the power of the system continues to grow. As a result, we can now refer to this work as the TalkBank Project.

3.1 Three Tools

The reasons for developing a computerized exchange system for language data are immediately obvious to anyone who has produced or analyzed transcripts. With such a system, we can:

automate the process of data analysis,
obtain better data in a consistent, fully-documented transcription system, and
provide more data for more children from more ages, speaking more languages.

The TalkBank Project has addressed each of these goals by developing three separate, but integrated, tools. The first tool is the CHAT transcription and coding format. The second tool is the CLAN analysis program, and the third tool is the database. These three tools are like the legs of a three-legged stool. The transcripts in the database have all been put into the CHAT transcription system. The program is designed to make full use of the CHAT format to facilitate a wide variety of searches and analyses. Many research groups are now using the CLAN programs to enter new data sets. Eventually, these new data sets will be available to other researchers as a part of the growing TalkBank databases. In this way, CHAT, CLAN, and the database function as an integrated set of tools. There are manuals for each of these TalkBank tools.

1. Part 1 of the TalkBank manual, which you are now reading, describes the conventions and principles of CHAT transcription.
2. Part 2 describes the use of the basic CLAN computer programs that you can use to transcribe, annotate, and analyze language interactions.
3. Part 3 describes the use of additional CLAN program for morphosyntactic analysis.
4. The final section of the manuals, which describes the contents of the databases, is broken out as a collection of index and documentation files on the web. For example, if want to survey the shape of the Dutch child language corpora, you first go to https://childes.talkbank.org. There is also a link to that site from the overall index at https://talkbank.org/. From that homepage you click on **Index to Corpora** and then Dutch. For there, you might want to read about the contents of the CLPF corpus for early phonological development in Dutch. You then click on the CLPF link and it takes you to the fuller corpus description with photos from the contributors. From links on that page you can either browse the corpus, download the transcripts, or download the media.

In addition to these basic manual resources, there are these further facilities for learning CHAT and CLAN, all of which can be downloaded from the talkbank.org and childes.talkbank.org server sites:

1. Nan Bernstein Ratner and Shelley Brundage have contributed a manual designed specifically for clinical practitioners called the SLP's Guide to CLAN.
2. There are versions of the manuals in Japanese and Chinese.
3. Davida Fromm has produced a series of screencasts describing how to use basic features of CLAN.

3.2 Shaping CHAT

We received a great deal of extremely helpful input during the years between 1984 and 1988 when the CHAT system was being formulated. Some of the most detailed comments came from George Allen, Elizabeth Bates, Nan Bernstein Ratner, Giuseppe Cappelli, Annick De Houwer, Jane Desimone, Jane Edwards, Julia Evans, Judi Fenson, Paul Fletcher, Steven Gillis, Kristen Keefe, Mary MacWhinney, Jon Miller, Barbara Pan, Lucia Pfanner, Kim Plunkett, Kelley Sacco, Catherine Snow, Jeff Sokolov, Leonid Spektor, Joseph Stemberger, Frank Wijnen, and Antonio Zampolli. Comments developed in Edwards (1992) were useful in shaping core aspects of CHAT. George Allen (1988) helped developed the UNIBET and PHONASCII systems. The workers in the LIPPS Group (LIPPS, 2000) have developed extensions of CHAT to cover code-switching phenomena. Adaptations of CHAT to deal with data on disfluencies are developed in Bernstein-Ratner, Rooney, and MacWhinney (1996). The exercises in the CLAN manual are based on materials originally developed by Barbara Pan for Chapter 2 of Sokolov & Snow (1994)

In the period between 2001 and 2004, we converted much of the CHILDES system to work with the new XML Internet data format. This work was begun by Romeo Anghelache and completed by Franklin Chen. Support for this major reformatting and the related tightening of the CHAT format came from the NSF TalkBank Infrastructure project which involved a major collaboration with Steven Bird and Mark Liberman of the Linguistic Data Consortium.

3.3 Building CLAN

The CLAN program is the brainchild of Leonid Spektor. Ideas for particular analysis commands came from several sources. Bill Tuthill's HUM package provided ideas about concordance analyses. The SALT system of Miller & Chapman (1983) provided guidelines regarding basic practices in transcription and analysis. Clifton Pye's PAL program provided ideas for the MODREP and PHONFREQ commands.

Darius Clynes ported CLAN to the Macintosh. Jeffrey Sokolov wrote the CHIP program. Mitzi Morris designed the MOR analyzer using specifications provided by Roland Hauser of Erlangen University. Norio Naka and Susanne Miyata developed a MOR rule system for Japanese; and Monica Sanz-Torrent helped develop the MOR system for Spanish. Julia Evans provided recommendations for the design of the audio and visual capabilities of the editor. Johannes Wagner and Spencer Hazel helped show us how we could modify CLAN to permit transcription in the Conversation Analysis framework. Steven Gillis provided suggestions for aspects of MODREP. Christophe Parisse built the

POST and POSTTRAIN programs (Parisse & Le Normand, 2000). Brian Richards contributed the VOCD program (Malvern, Richards, Chipere, & Purán, 2004). Julia Evans helped specify TIMEDUR and worked on the details of DSS. Catherine Snow designed CHAINS, KEYMAP, and STATFREQ. Nan Bernstein Ratner specified aspects of PHONFREQ and plans for additional programs for phonological analysis.

3.4 Constructing the Database

The primary reason for the success of the TalkBank databases has been the generosity of over 300 researchers who have contributed their corpora. Each of these corpora represents hundreds, often thousands, of hours spent in careful collection, transcription, and checking of data. All researchers in child language should be proud of the way researchers have generously shared their valuable data with the whole research community. The growing size of the database for language impairments, adult aphasia, and second-language acquisition indicates that these related areas have also begun to understand the value of data sharing.

Many of the corpora contributed to the system were transcribed before the formulation of CHAT. In order to create a uniform database, we had to reformat these corpora into CHAT. Jane Desimone, Mary MacWhinney, Jane Morrison, Kim Roth, Kelley Sacco, Lillian Jarold, Anthony Kelly, Andrew Yankes, and Gergely Sikuta worked many long hours on this task. Steven Gillis, Helmut Feldweg, Susan Powers, and Heike Behrens supervised a parallel effort with the German and Dutch data sets.

Because of the continually changing shape of the programs and the database, keeping this manual up to date has been an ongoing activity. In this process, I received help from Mike Blackwell, Julia Evans, Kris Loh, Mary MacWhinney, Lucy Hewson, Kelley Sacco, and Gergely Sikuta. Barbara Pan, Jeff Sokolov, and Pam Rollins also provided a reading of the final draft of the 1995 version of the manual.

3.5 Dissemination

Since the beginning of the project, Catherine Snow has continually played a pivotal role in shaping policy, building the database, organizing workshops, and determining the shape of CHAT and CLAN. Catherine Snow collaborated with Jeffrey Sokolov, Pam Rollins, and Barbara Pan to construct a series of tutorial exercises and demonstration analyses that appeared in Sokolov & Snow (1994). Those exercises form the basis for similar tutorial sections in the current manual. Catherine Snow has contributed six major corpora to the database and has conducted CHILDES workshops in a dozen countries.

Several other colleagues have helped disseminate the CHILDES system through workshops, visits, and Internet facilities. Hidetosi Sirai established a CHILDES file server mirror at Chukyo University in Japan and Steven Gillis established a mirror at the University of Antwerp. Steven Gillis, Kim Plunkett, Johannes Wagner, and Sven Strömqvist helped propagate the CHILDES system at universities in Northern and Central Europe. Susanne Miyata has brought together a vital group of child language researchers using CHILDES to study the acquisition of Japanese and has supervised the translation of the current manual into Japanese. In Italy, Elena Pizzuto organized symposia for developing the CHILDES system and has supervised the translation of the manual into Italian. Magdalena Smoczynska in Krakow and Wolfgang Dressler in Vienna have helped new researchers who are learning to use CHILDES for languages spoken in

Eastern Europe. Miquel Serra has supported a series of CHILDES workshops in Barcelona. Zhou Jing organized a workshop in Nanjing and Chien-ju Chang organized a workshop in Taipei.

The establishment and promotion of additional segments of TalkBank now relies on a wide array of inputs. Yvan Rose has spearheaded the creation of PhonBank. Nan Bernstein Ratner has led the development of FluencyBank. Audrey Holland, Davida Fromm, and Margie Forbes have worked to create AphasiaBank. Johannes Wagner has created SamtaleBank and segments of CABank. Jerry Goldman developed the SCOTUS segment of CABank. Roy Pea contributed to the development of ClassBank. Within each of these communities, scores of other scholars have helped with donations of corpora, analyses, and ideas.

3.6 Funding

From 1984 to 1988, the John D. and Catherine T. MacArthur Foundation supported the CHILDES Project. In 1988, the National Science Foundation provided an equipment grant that allowed us to put the database on the Internet and on CD-ROMs. From 1989, the CHILDES project has been supported by an ongoing grant from the National Institutes of Health (NICHHD). In 1998, the National Science Foundation Linguistics Program provided additional support to improve the programs for morphosyntactic analysis of the database. In 1999, NSF funded the TalkBank project. In 2002, NSF provided support for the development of the GRASP system for parsing of the corpora. In 2002, NIH provided additional support for the development of PhonBank for child language phonology and AphasiaBank for the study of communication in aphasia. Currently (2017), NICHD is providing support for CHILDES and PhonBank; NIDCD provides support for AphasiaBank and FluencyBank, NSF provides support for HomeBank and FluencyBank, and NEH provides support for LangBank. Beginning in 2014, TalkBank also became a member of the CLARIN federation (clarin.eu), a system designed to coordinate resources for language computation in the Humanities and Social Sciences.

3.7 How to Use These Manuals

Each of the three parts of the TalkBank system is described in separate sections of the TalkBank manual. The CHAT manual describes the conventions and principles of CHAT transcription. The CLAN manual describes the use of the editor and the analytic commands. The database manual is a set of over a dozen smaller documents, each describing a separate segment of the database.

To learn the TalkBank system, you should begin by downloading and installing the CLAN program. Next, you should download and start to read the current manual (CHAT Manual) and the CLAN manual (Part 2 of the TalkBank manual). Before proceeding too far into the CHAT manual, you will want to walk through the tutorial section at the beginning of the CHAT manual. After finishing the tutorial, try working a bit with each of the CLAN commands to get a feel for the overall scope of the system. You can then learn more about CHAT by transcribing a small sample of your data in a short test file. Run the CHECK program at frequent intervals to verify the accuracy of your coding. Once you have finished transcribing a small segment of your data, try out the various

analysis programs you plan to use, to make sure that they provide the types of results you need for your work.

If you are primarily interested in analyzing data already stored in TalkBank, you do not need to learn the CHAT transcription format in much detail and you will only need to use the editor to open and read files. In that case, you may wish to focus your efforts on learning to use the CLAN programs. If you plan to transcribe new data, then you also need to work with the current manual to learn to use CHAT.

Teachers will also want to pay particular attention to the sections of the CLAN manual that present a tutorial introduction. Using some of the examples given there, you can construct additional materials to encourage students to explore the database to test out particular hypotheses.

The TalkBank system was not intended to address all issues in the study of language learning, or to be used by all students of spontaneous interactions. The CHAT system is comprehensive, but it is not ideal for all purposes. The programs are powerful, but they cannot solve all analytic problems. It is not the goal of TalkBank to provide facilities for all research endeavors or to force all research into some uniform mold. On the contrary, the programs are designed to offer support for alternative analytic frameworks. For example, the editor now supports the various codes of Conversation Analysis (CA) format, as alternatives and supplements to CHAT format.. Moreover, we have developed programs that convert between CHAT format and other common formats, because we know that users often need to run analyses in these other formats.

3.8 Changes

The TalkBank tools have been extensively tested for ease of application, accuracy, and reliability. However, change is fundamental to any research enterprise. Researchers are constantly pursuing better ways of coding and analyzing data. It is important that the tools keep progress with these changing requirements. For this reason, there will be revisions to CHAT, the programs, and the database as long as the TalkBank Project is active.

4 Principles

The CHAT system provides a standardized format for producing computerized transcripts of face-to-face conversational interactions. These interactions may involve children and parents, doctors and patients, or teachers and second-language learners. Despite the differences between these interactions, there are enough common features to allow for the creation of a single general transcription system. The system described here is designed for use with both normal and disordered populations. It can be used with learners of all types, including children, second-language learners, and adults recovering from aphasic disorders. The system provides options for basic discourse transcription as well as detailed phonological and morphological analysis. The system bears the acronym "CHAT," which stands for Codes for the Human Analysis of Transcripts. CHAT is the standard transcription system for the TalkBank and CHILDES (Child Language Data Exchange System) Projects. All of the transcripts in the TalkBank databases are in CHAT format.

What makes CHAT particularly powerful is the fact that files transcribed in CHAT can also be analyzed by the CLAN programs that are described in the CLAN manual, which is an electronic companion piece to this manual. The CHAT programs can track a wide variety of structures, compute automatic indices, and analyze morphosyntax. Moreover, because all CHAT files can now also be translated to a highly structured form of XML (a language used for text documents on the web), they are now also compatible with a wide range of other powerful computer programs such as ELAN, Praat, EXMARaLDA, Phon, Transcriber, and so on.

The TalkBank system has had a major impact on the study of child language. At the time of the last monitoring in 2016, there were over 7000 published articles that had made use of the programs and database. In 2016, the size of the database had grown to over 110 million words, making it by far the largest database of conversational interactions available anywhere. The total number of researchers who have joined as members across the length of the project is now over 5000. Of course, not all of these people are making active use of the tools at all times. However, it is safe to say that, at any given point in time, well over 100 groups of researchers around the world are involved in new data collection and transcription using the CHAT system. Eventually the data collected in these various projects will all be contributed to the database.

4.1 Computerization

Public inspection of experimental data is a crucial prerequisite for serious scientific progress. Imagine how genetics would function if every experimenter had his or her own individual strain of peas or drosophila and refused to allow them to be tested by other experimenters. What would happen in geology, if every scientist kept his or her own set of rock specimens and refused to compare them with those of other researchers? In some fields the basic phenomena in question are so clearly open to public inspection that this is not a problem. The basic facts of planetary motion are open for all to see, as are the basic facts underlying Newtonian mechanics.

Unfortunately, in language studies, a free and open sharing and exchange of data has not always been the norm. In earlier decades, researchers jealously guarded their field

notes from a particular language community of subject type, refusing to share them openly with the broader community. Various justifications were given for this practice. It was sometimes claimed that other researchers would not fully appreciate the nature of the data or that they might misrepresent crucial patterns. Sometimes, it was claimed that only someone who had actually participated in the community or the interaction could understand the nature of the language and the interactions. In some cases, these limitations were real and important. However, all such restrictions on the sharing of data inevitably impede the progress of the scientific study of language learning.

Within the field of language acquisition studies it is now understood that the advantages of sharing data outweigh the potential dangers. The question is no longer whether data should be shared, but rather how they can be shared in a reliable and responsible fashion. The computerization of transcripts opens up the possibility for many types of data sharing and analysis that otherwise would have been impossible. However, the full exploitation of this opportunity requires the development of a standardized system for data transcription and analysis.

4.2 Words of Caution

Before examining the CHAT system, we need to consider some dangers involved in computerized transcriptions. These dangers arise from the need to compress a complex set of verbal and nonverbal messages into the extremely narrow channel required for the computer. In most cases, these dangers also exist when one creates a typewritten or handwritten transcript. Let us look at some of the dangers surrounding the enterprise of transcription.

4.2.1 The Dominance of the Written Word

Perhaps the greatest danger facing the transcriber is the tendency to treat spoken language as if it were written language. The decision to write out stretches of vocal material using the forms of written language can trigger a variety of theoretical commitments. As Ochs (1979) showed so clearly, these decisions will inevitably turn transcription into a theoretical enterprise. The most difficult bias to overcome is the tendency to map every form spoken by a learner – be it a child, an aphasic, or a second-language learner – onto a set of standard lexical items in the adult language. Transcribers tend to assimilate nonstandard learner strings to standard forms of the adult language. For example, when a child says “put on my jamas,” the transcriber may instead enter “put on my pajamas,” reasoning unconsciously that “jamas” is simply a childish form of “pajamas.” This type of regularization of the child form to the adult lexical norm can lead to misunderstanding of the shape of the child's lexicon. For example, it could be the case that the child uses “jamas” and “pajamas” to refer to two very different things (Clark, 1987; MacWhinney, 1989).

There are two types of errors possible here. One involves mapping a learner's spoken form onto an adult form when, in fact, there was no real correspondence. This is the problem of overnormalization. The second type of error involves failing to map a learner's spoken form onto an adult form when, in fact, there is a correspondence. This is the problem of undernormalization. The goal of transcribers should be to avoid both the Scylla of overnormalization and the Charybdis of undernormalization. Steering a course

between these two dangers is no easy matter. A transcription system can provide devices to aid in this process, but it cannot guarantee safe passage.

Transcribers also often tend to assimilate the shape of sounds spoken by the learner to the shapes that are dictated by morphosyntactic patterns. For example, Fletcher (1985) noted that both children and adults generally produce "have" as "uv" before main verbs. As a result, forms like "might have gone" assimilate to "mightuv gone." Fletcher believed that younger children have not yet learned to associate the full auxiliary "have" with the contracted form. If we write the children's forms as "might have," we then end up mischaracterizing the structure of their lexicon. To take another example, we can note that, in French, the various endings of the verb in the present tense are distinguished in spelling, whereas they are homophonous in speech. If a child says /mʌnʒ/ "eat," are we to transcribe it as first person singular *mange*, as second person singular *manges*, or as the imperative *mange*? If the child says /mãʒe/, should we transcribe it as the infinitive *manger*, the participle *mangé*, or the second person formal *mangez*?

CHAT deals with these problems in three ways. First, it uses IPA as a uniform way of transcribing discourse phonetically. Second, the editor allows the user to link the digitized audio record of the interaction directly to the transcript. This is the system called "sonic CHAT." With these sonic CHAT links, it is possible to double-click on a sentence and hear its sound immediately. Having the actual sound produced by the child directly available in the transcript takes some of the burden off of the transcription system. However, whenever computerized analyses are based not on the original audio signal but on transcribed orthographic forms, one must continue to understand the limits of transcription conventions. Third, for those who wish to avoid the work involved in IPA transcription or sonic CHAT, that is a system for using nonstandard lexical forms, that the form "might (h)ave" would be universally recognized as the spelling of "mightof", the contracted form of "might have." More extreme cases of phonological variation can be annotated as in this example: popo [: hippopotamus].

4.2.2 The Misuse of Standard Punctuation

Transcribers have a tendency to write out spoken language with the punctuation conventions of written language. Written language is organized into clauses and sentences delimited by commas, periods, and other marks of punctuation. Spoken language, on the other hand, is organized into tone units clustered about a tonal nucleus and delineated by pauses and tonal contours (Crystal, 1969, 1979; Halliday, 1966, 1967, 1968). Work on the discourse basis of sentence production (Chafe, 1980; Jefferson, 1984) has demonstrated a close link between tone units and ideational units. Retracings, pauses, stress, and all forms of intonational contours are crucial markers of aspects of the utterance planning process. Moreover, these features also convey important sociolinguistic information. Within special markings or conventions, there is no way to directly indicate these important aspects of interactions.

4.2.3 Working With Video

Whatever form a transcript may take, it will never contain a fully accurate record of what went on in an interaction. A transcript of an interaction can never fully replace an audiotape, because an audio recording of the interaction will always be more accurate in

terms of preserving the actual details of what transpired. By the same token, an audio recording can never preserve as much detail as a video recording with a high-quality audio track. Audio recordings record none of the nonverbal interactions that often form the backbone of a conversational interaction. Hence, they systematically exclude a source of information that is crucial for a full interpretation of the interaction. Although there are biases involved even in a video recording, it is still the most accurate record of an interaction that we have available. For those who are trying to use transcription to capture the full detailed character of an interaction, it is imperative that transcription be done from a video recording which should be repeatedly consulted during all phases of analysis.

When the CLAN editor is used to link transcripts to audio recordings, we refer to this as sonic CHAT. When the system is used to link transcripts to video recordings, we refer to this as video CHAT. The CLAN manual explains how to link digital audio and video to transcripts.

4.3 Problems With Forced Decisions

Transcription and coding systems often force the user to make difficult distinctions. For example, a system might make a distinction between grammatical ellipsis and ungrammatical omission. However, it may often be the case that the user cannot decide whether an omission is grammatical or not. In that case, it may be helpful to have some way of blurring the distinction. CHAT has certain symbols that can be used when a categorization cannot be made. It is important to remember that many of the CHAT symbols are entirely optional. Whenever you feel that you are being forced to make a distinction, check the manual to see whether the particular coding choice is actually required. If it is not required, then simply omit the code altogether.

4.4 Transcription and Coding

It is important to recognize the difference between *transcription* and *coding*. Transcription focuses on the production of a written record that can lead us to understand, albeit only vaguely, the flow of the original interaction. Transcription must be done directly off an audiotape or, preferably, a videotape. Coding, on the other hand, is the process of recognizing, analyzing, and taking note of phenomena in transcribed speech. Coding can often be done by referring only to a written transcript. For example, the coding of parts of speech can be done directly from a transcript without listening to the audiotape. For other types of coding, such as speech act coding, it is imperative that coding be done while watching the original videotape.

The CHAT system includes conventions for both transcription and coding. When first learning the system, it is best to focus on learning how to transcribe. The CHAT system offers the transcriber a large array of coding options. Although few transcribers will need to use all of the options, everyone needs to understand how basic transcription is done on the "main line." Additional coding is done principally on the secondary or "dependent" tiers. As transcribers work more with their data, they will include further options from the secondary or "dependent" tiers. However, the beginning user should focus first on learning to correctly use the conventions for the main line. The manual includes several sample transcripts to help the beginner in learning the transcription system.

4.5 Three Goals

Like other forms of communication, transcription systems are subjected to a variety of communicative pressures. The view of language structure developed by Slobin (1977) sees structure as emerging from the pressure of three conflicting charges or goals. On the one hand, language is designed to be **clear**. On the other hand, it is designed to be **processible** by the listener and quick and **easy** for the speaker. Unfortunately, ease of production often comes in conflict with clarity of marking. The competition between these three motives leads to a variety of imperfect solutions that satisfy each goal only partially. Such imperfect and unstable solutions characterize the grammar and phonology of human language (Bates & MacWhinney, 1982). Only rarely does a solution succeed in fully achieving all three goals.

Slobin's view of the pressures shaping human language can be extended to analyze the pressures shaping a transcription system. In many regards, a transcription system is much like any human language. It needs to be clear in its markings of categories, and still preserve readability and ease of transcription. However, transcripts address rather different audiences. One audience is the human audience of transcribers, analysts, and readers. The other audience is the digital computer and its programs. To deal with these two audiences, a system for computerized transcription needs to achieve the following goals:

Clarity: Every symbol used in the coding system should have some clear and definable real-world referent. Symbols that mark particular words should always be spelled in a consistent manner. Symbols that mark particular conversational patterns should refer to consistently observable patterns. Codes must steer between the Scylla of overregularization and the Charybdis of underregularization discussed earlier. Distinctions must avoid being either too fine or too coarse. Another way of looking at clarity is through the notion of systematicity. Codes, words, and symbols must be used in a consistent manner across transcripts. Ideally, each code should always have a unique meaning independent of the presence of other codes or the particular transcript in which it is located. If interactions are necessary, as in hierarchical coding systems, these interactions need to be systematically described.

Readability: Just as human language needs to be easy to process, so transcripts need to be easy to read. This goal often runs directly counter to the first goal. In the TalkBank system, we have attempted to provide a variety of CHAT options that will allow a user to maximize the readability of a transcript. We have also provided clan tools that will allow a reader to suppress the less readable aspects in transcript when the goal of readability is more important than the goal of clarity of marking.

Ease of data entry: As distinctions proliferate within a transcription system, data entry becomes increasingly difficult and error-prone. There are two ways of dealing with this problem. One method attempts to simplify the coding scheme and its categories. The problem with this approach is that it sacrifices clarity. The second method attempts to help the transcriber by providing computational aids. The CLAN programs follow this path. They provide systems for the automatic checking of transcription accuracy, methods for the automatic analysis of morphology and syntax, and tools for the semiautomatic entry of codes. However, the basic process of transcription has not been automated and remains the major task during data entry.

5 minCHAT

CHAT provides both basic and advanced formats for transcription and coding. The basic level of CHAT is called minCHAT. New users should start by learning minCHAT. This system looks much like other intuitive transcription systems that are in general use in the fields of child language and discourse analysis. However, eventually users will find that there is something they want to be able to code that goes beyond minCHAT. At that point, they should move on to learning the additional features of CHAT that are relevant for the type of working they are doing.

5.1 minCHAT – the Form of Files

There are several minimum standards for the form of a minCHAT file. These standards must be followed for the CLAN commands to run successfully on CHAT files:

1. Every line must end with a carriage return.
2. The first line in the file must be an @Begin header line.
3. The second line in the file must be an @Languages header line. The languages entered here use a three-letter ISO 639-3 code, such as "eng" for English.
4. The third line must be an @Participants header line listing three-letter codes for each participant, the participant's name, and the participant's role.
5. After the @Participants header come a set of @ID headers providing further details for each speaker. These will be inserted automatically for you when you run CHECK using escape-L.
6. The last line in the file must be an @End header line.
7. Lines beginning with * indicate what was actually said. These are called "main lines." Each main line should code one and only one utterance. When a speaker produces several utterances in a row, code each with a new main line.
8. After the asterisk on the main line comes a three-letter code in upper case letters for the participant who was the speaker of the utterance being coded. After the three-letter code comes a colon and then a tab.
9. What was actually said is entered starting in the ninth column.
10. Lines beginning with the % symbol can contain codes and commentary regarding what was said. They are called "dependent tier" lines. The % symbol is followed by a three-letter code in lowercase letters for the dependent tier type, such as "pho" for phonology; a colon; and then a tab. The text of the dependent tier begins after the tab.
11. Continuations of main lines and dependent tier lines begin with a tab which is inserted automatically by the CLAN editor.

5.2 minCHAT – Words and Utterances

In addition to these minimum requirements for the form of the file, there are certain minimum ways in which utterances and words should be written on the main line:

1. Utterances must end with an utterance terminator. The basic utterance terminators are the period, the exclamation mark, and the question mark. These can be preceded by a space, but the space is not required.

2. Commas can be used as needed to mark phrasal junctions, but they are not used by the programs and have no sharp prosodic definition.
3. Use upper case letters only for proper nouns and the word "I." Do not use upper-case letters for the first words of sentences. This will facilitate the identification of proper nouns.
4. To facilitate recognition of proper nouns and avoid misspellings, words should not contain capital letters except at their beginning. Words should not contain numbers, unless these mark tones.
5. Unintelligible words with an unclear phonetic shape should be transcribed as **xxx**.
6. If you wish to note the phonological form of an incomplete or unintelligible phonological string, write it out with an ampersand, as in **&guga**.
7. Incomplete words can be written with the omitted material in parentheses, as in **(be)cause** and **(a)bout**.

Here is a sample that illustrates these principles. This file is syntactically correct and uses the minimum number of CHAT conventions while still maintaining compatibility with the CLAN commands.

```
@Begin
@Languages:     eng
@Participants: CHI Ross Child, FAT Brian Father
@ID:      eng|macwhinney|CHI|2;10.10||||Target_Child|||
@ID:      eng|macwhinney|FAT|35;2.||||Target_Child|||
*ROS:     why isn't Mommy coming?
%com:     Mother usually picks Ross up around 4 PM.
*FAT:     don't worry.
*FAT:     she'll be here soon.
*CHI:     good.
@End
```

5.3 Analyzing One Small File

For researchers who are just now beginning to use CHAT and CLAN, there is one single suggestion that can potentially save literally hundreds of hours of wasted time. The suggestion is to transcribe and analyze one single small file completely and perfectly before launching a major effort in transcription and analysis. The idea is that you should learn just enough about minCHAT and minCLAN to see your path through these four crucial steps:

1. entry of a small set of your data into a CHAT file,
2. successful running of the CHECK command inside the editor to guarantee accuracy in your CHAT file,
3. development of a series of codes that will interface with the particular CLAN commands most appropriate for your analysis, and
4. running of the relevant CLAN commands, so that you can be sure that the results you will get will properly test the hypotheses you wish to develop.

If you go through these steps first, you can guarantee in advance the successful outcome of your project. You can avoid ending up in a situation in which you have transcribed hundreds of hours of data in a way that does not match correctly with the input requirements for CLAN.

5.4 Next Steps

After having learned minCHAT, you are ready to learn the basics of CLAN. To do this, you will want to work through the first chapters of the CLAN manual focusing in particular on the CLAN tutorial. These chapters will take you up to the level of minCLAN, which corresponds to the minCHAT level.

Once you have learned minCHAT and minCLAN, you are ready to move on to learning the rest of the system. You should next work through the chapters on words, utterances, and scoped symbols. Depending on the shape of your particular project, you may then need to study additional chapters in this manual. For people working on large projects that last many months, it is a good idea to eventually read all of the current manual, although some sections that seem less relevant to the project can be skimmed.

5.5 Checking Syntactic Accuracy

Each CLAN command runs a very superficial check to see if a file conforms to min-CHAT. This check looks only to see that each line begins with either @, *, %, a tab or a space. This is the minimum that the CLAN commands must have to function. However, the correct functioning of many of the functions of CLAN depends on adherence to further standards for minCHAT. In order to make sure that a file matches these minimum requirements for correct analysis through CLAN, researchers should run each file through the CHECK program. The CHECK command can be run directly inside the editor, so that you can verify the accuracy of your transcription as you are producing it. CHECK will detect errors such as failure to start lines with the correct symbols, use of incorrect speaker codes, or missing @Begin and @End symbols. CHECK can also be used to find errors in CHAT coding beyond those discussed in this chapter. Using CHECK is like brushing your teeth. It may be hard at first to remember to use the command, but the more you use it the easier it becomes and the better the final results.

6 Corpus Organization

6.1 File Naming

Each TalkBank database consists of a collection of corpora, organized into larger folders by languages and language groups. For example, there is a top-level folder called Romance in which one finds subfolders for Spanish, French, and other Romance languages. Within the Spanish folder, there are then dozens of further folders, each of which has a single corpus. With a corpus, files may be further grouped by individual children or groups of children. For longitudinal corpora, we recommend that file names use the age of the child followed by a letter if there are several recordings from a given day. For example, the transcript from the fourth taping session when the child was 2;3;22 would be called 20322d.cha. It is better to use ages for file names, rather than dates or other material.

6.2 Metadata

Increasingly, researchers rely on Internet systems to locate and retrieve language data and resources. There are currently several systems designed to facilitate this process and we have adapted the indexing and registration of materials in the CHILDES and TalkBank systems to provide information that can be incorporated into these systems. The two systems designed specifically to deal with linguistic data are OLAC (Online Language Archives Community at www.language-archives.org) and VLO (Virtual Language Observatory at vlo.clarin.eu). These systems allow researchers to search for whole corpora or single files, using terms such as *Cantonese, video, gesture,* or *aphasia.* In order to publish or register TalkBank data within these systems, we create a 0metadata.cdc file at the top level of each corpus in TalkBank. Some of the fields in this metadata file are designed for indexing in OLAC and some are designed for the CMDI system used by VLO and the related facility called The Language Archive (tla.mpi.nl). Because of the highly specific nature of the terms and the software used for regular harvesting and publication of these data, we do not require users to create the 0metadata.cdc files. The following table explains what keywords are expected within each field of these files. The first fields listed are for OLAC and the later ones are for CMDI. For CMDI, the values *unknown* and *unspecified* are also available for most of the fields.

Field	Example	Values
CMDI_PID:	11312/c-00041631-1	Set by Handle Server system
Title:	Bilingual AarsenBos Corpus	open
Creator:	Aarssen, Jeroen	open
Creator:	Bos, Petra	open
Subject:	child language development	
Subject.olac:linguistic-field:	language_acquisition	
Subject.olac:language:	ndl	ISO-639
Subject.olac:language:	tur	ISO-639
Subject.olac:language:	ara	ISO-639
Subject.childes:participant:	age="4 - 10"	open
Description:		open

Publisher:	TalkBank	open
Contributor:	Aarssen, Jeroen	open
Date:	2004-03-30	YEAR-MM-DD
Type:	Text	text, video,
Type.olac:linguistic-type:	primary_text	lexicon, primary_text, language_description
Type.olac:discourse-type:	dialogue	dialogue, drama, formulaic, ludic, oratory, narrative, procedural, report, singing, unintelligible speech
Format:		
Identifier:	1-59642-132-0	ISBN
Language:		ISO-639
Relation:		open
Coverage:		open
Rights:		open
IMDI_Genre:	discourse	
IMDI_Interactivity:	interactive	interactive, non-interactive, semi-interactive
IMDI_PlanningType:	spontaneous	spontaneous, semi-spontaneous, planned
IMDI_Involvement:	non-elicited	elicited, non-elicited, no-observer
IMDI_SocialContext:	family	family, private, public, controlled environment, talkshow, shopping, face_to_face, lecture, legal, religious, sports, tutorial, classroom, medical work, meeting, clinic, telechat, phonecall, computer, constructed
IMDI_EventStructure:	conversation	monologue, dialogue, conversation, not a natural format
IMDI_Task:	unspecified	open
IMDI_Modalities:	speech	open
IMDI_Subject:	unspecified	open
IMDI_EthnicGroup:	unspecified	open
IMDI_RecordingConditions:	unspecified	open
IMDI_AccessAvailability:	open access	open
IMDI_Continent:	Europe	Dublin Core
IMDI_Country:	Netherlands	Dublin Core
IMDI_ProjectDescription:		open
IMDI_MediaFileDescription:		open
IMDI_WrittenResourceSubType:		open

For the CMDI/VLO/CLARIN system, there must be a cmdi.xml file for each transcript. To create these several thousand files, we use a CLAN program that takes the information from the 0metadata.cdc files and from the header lines in each transcript. The information in the @ID field is particularly important in this process. It also relies on the fact that we use an isomorphic file system for indexing media files. Fortunately, users do not need to concern themselves with all these many additional technical details.

6.3 The Documentation File

CHAT files typically record a conversational sample collected from a particular set of speakers on a particular day. Sometimes researchers study a small set of children repeatedly over a long period of time. Corpora created using this method are referred to as longitudinal studies. For such studies, it is best to break up CHAT files into one collection for each child. This can be done just by creating file names that begin with the three letter code for the child, as in lea001.cha or eve15.cha. Each collection of files from the children involved in a given study constitutes a corpus. A corpus can also be composed of a group of files from different groups of speakers when the focus is on a cross-sectional sampling of larger numbers of language learners from various age groups. In either case, each corpus should have a documentation file. This "readme" file should contain a basic set of facts that are indispensable for the proper interpretation of the data by other researchers. The minimum set of facts that should be in each readme file are the following.

Acknowledgments. There should be a statement that asks the user to cite some particular reference when using the corpus. For example, researchers using the Adam, Eve, and Sarah corpora from Roger Brown and his colleagues are asked to cite Brown (1973). In addition, all users can cite this current manual as the source for the TalkBank system in general.

Restrictions. If the data are being contributed to TalkBank, contributors can set particular restrictions on the use of their data. For example, researchers may ask that they be sent copies of articles that make use of their data. Many researchers have chosen to set no limitations at all on the use of their data.

Warnings. This documentation file should also warn other researchers about limitations on the use of the data. For example, if an investigator paid no attention to correct transcription of speech errors, this should be noted.

Pseudonyms. The readme file should also include information on whether informants gave informed consent for the use of their data and whether pseudonyms have been used to preserve informant anonymity. In general, real names should be replaced by pseudonyms. Anonymization is not necessary when the subject of the transcriptions is the researcher's own child, as long as the child grants permission for the use of the data.

History. There should be detailed information on the history of the project. How was funding obtained? What were the goals of the project? How was data collected? What was the sampling procedure? How was transcription done? What was ignored in transcription? Were transcribers trained? Was reliability checked? Was coding done? What codes were used? Was the material computerized? How?

Codes. If there are project-specific codes, these should be described.

Biographical data. Where possible, extensive demographic, dialectological, and psychometric data should be provided for each informant. There should be information on topics such as age, gender, siblings, schooling, social class, occupation, previous residences, religion, interests, friends, and so forth. Information on where the parents grew up and the various residences of the family is particularly important in attempting to understand sociolinguistic issues regarding language change, regionalism, and dialect. Without detailed information about specific dialect features, it is difficult to know whether these particular markers are being used throughout the language or just in certain regions.

Situational descriptions. The readme file should include descriptions of the contexts of the recordings, such as the layout of the child's home and bedroom or the nature of the activities being recorded. Additional specific situational information should be included in the @Situation and @Comment fields in each file.

7 File Headers

The three major components of a CHAT transcript are the file headers, the main tier, and the dependent tiers. In this chapter we discuss creating the first major component – the file headers. A computerized transcript in CHAT format begins with a series of "header" lines, which tells us about things such as the date of the recording, the names of the participants, the ages of the participants, the setting of the interaction, and so forth.

A header is a line of text that gives information about the participants and the setting. All headers begin with the "@" sign. Some headers require nothing more than the @ sign and the header name. These are "bare" headers such as @Begin or @New Episode. However, most headers require that there be some additional material. This additional material is called an "entry." Headers that take entries must have a colon, which is then followed by one or two tabs and the required entry. By default, tabs are usually understood to be placed at eight-character intervals. The material up to the colon is called the "header name." In the example following, "@Media" and "@Date" are both header names

```
@Media:   abe88, video
@Date: 25-JAN-1983
```

The text that follows the header name is called the "header entry." Here, "abe88 movie" and "25-JAN-1983" are the header entries. The header name and the header entry together are called the "header line." The header line should never have a punctuation mark at the end. In CHAT, only utterances actually spoken by the subjects receive final punctuation.

This chapter presents a set of headers that researchers have considered important. Except for the @Begin, @Languages, @Participants, @ID, and @End headers, none of the headers are required and you should feel free to use only those headers that you feel are needed for the accurate documentation of your corpus.

7.1 Hidden Headers

CHAT uses five types of headers: hidden, initial, participant-specific, constant, and changeable. In the editor, CHAT files appear to begin with the @Begin header. However, there are actually four hidden headers that appear before this header. These are the @Font header, the @UTF8 header, the @PID header, and the optional @ColorWords header which appear in that order.

@Font

This header is used to set the default font for the file. This line appears at the beginning of the file and its presence is hidden in the CLAN editor. When this header is missing, CLAN tries to determine which font is most appropriate for use with the current file by examining information in the @Languages and @Options headers. If CLAN's choice is not appropriate for the file, then the user will have to change the font. After this is done, the font information will be stored in this header line. Files that are retrieved

from the database often do not have this header included, thereby allowing CLAN and the user to decide which font is most appropriate for viewing the current file.

@UTF8

This hidden header follows after the @Font header. All files in the database use this header to mark the fact that they are encoded in UTF8. If the file was produced outside of CLAN and this header is missing, CLAN will complain and ask the user to verify whether the file should be read in UTF8. Often this means that the user should run the CP2UTF program to convert the file to UTF8.

@PID

This hidden header follows after the @UTF header and it declares the value of the transcript for the Handle System (www.handle.net) that allows for persistent identification of the location of digital objects. These numbers are then further processed using the CMDI metadata scheme for publication and harvesting over the web through the CLARIN (www.clarin.eu) schema that creates access through TLA (The Language Archive; https://tla.mpi.nl/) and the VLO (Virtual Language Observatory; https://vlo.clarin.edu), as well as parallel methods from OLAC (Online Language Archives Community).

These values can be entered into any system that resolves PIDs to locate the required resource, such as the server at https://128.2.71.222:8000. For example one of the files from the MacWhinney corpus has this number 11312/c-00044068-1 which refers to the CMDI metadata file for that transcript. If you change the -1 to -2, then it refers to the transcript itself. If you change the -1 to -3, it refers to the media, if that exists. There are also PID numbers in the 0metadata.cdc file that accompanies each corpus. When those numbers end in -1, they refer to the CMDI file associated with the corpus. If you change that -1 to -2, it refers to the .zip file that you can download for the corpus.

@ColorWords

This hidden header stores the color values that users create when using the Color Keywords dialog.

7.2 Initial Headers

CHAT has seven initial headers. The first six of these – @Begin, @Languages, @Participants, @Options, @ID, and @Media – appear in this order as the first lines of the file. The last one @End appears at the end of the file as the last line.

@Begin

This header is always the first visible header placed at the beginning of the file. It is needed to guarantee that no material has been lost at the beginning of the file. This is a "bare" header that takes no entry and uses no colon.

@Languages:

This is the second visible header; it tells the programs which language is being used in the dialogues. Here is an example of this line for a bilingual transcript using Swedish and Portuguese.

```
@Languages:     swe, por
```

The language codes come from the international ISO 639-3 standard. For the languages currently in the database, these three-letter codes and extended codes are used:

Language	Code	Language	Code	Language	Code
Afrikaans	afr	German	deu	Polish	pol
Arabic	ara	Greek	ell	Portuguese	por
Basque	eus	Hebrew	heb	Punjabi	pan
Cantonese	zho-yue	Hungarian	hun	Romanian	ron
Catalan	cat	Icelandic	isl	Russian	rus
Chinese	zho	Indonesian	ind	Sesotho	sot
Cree	crl	Irish	gle	Spanish	spa
Croatian	hrv	Italian	ita	Swahili	swa
Czech	ces	Japanese	jpn	Swedish	swe
Danish	dan	Javanese	jav	Tagalog	tag
Dutch	nld	Kannada	kan	Taiwanese	zho-min
English	eng	Kikuyu	kik	Tamil	tam
Estonian	est	Korean	kor	Thai	tha
Farsi	fas	Lithuanian	lit	Turkish	tur
Finnish	sun	Norwegian	nor	Vietnamese	vie
French	fra			Welsh	cym
Galician	glg			Yiddish	yid

We continually update this list, and CLAN relies on a file in the lib/fixes directory called ISO-639.cut that lists the current languages. There are special conditions for certain languages. For example, tone languages like Cantonese, Mandarin, and Thai are allowed to have Romanized word forms that include tone numbers. In addition, Chinese words in non-Roman characters can use numbers to disambiguate homonyms.

In multilingual corpora, several codes can be combined on the @Languages line. The first code given is for the language used most frequently in the transcript. Individual utterances in a second or third most frequent languages can be marked with precodes as in this example:

```
*CHI:     [- eng] this is my juguete@s .
```

In this example, Spanish is the most frequent language, but the particular sentence is marked as English. The @Languages header lists spa for Spanish, and then eng for

English. Within this English sentence, the use of a Spanish word is then marked as @s. When the @s is used in the main body of the transcript without the [- eng], then it indicates a shift to English, rather than to Spanish. Please see the section on code-switching annotation for further details on the use of these codes for interactions with code-switching.

@Participants:

This is the third visible header. Like the @Begin and @Participants headers, it is obligatory. It lists all of the speakers within the file. The format for this header is XXX Name Role, XXX Name Role, XXX Name Role. XXX stands for the three-letter speaker ID. Here is an example of a completed @Participants header line:

```
@Participants: SAR Sue_Day Target_Child, CAR Carol Mother
```

Participants are identified by three elements: their speaker ID, their name and their role.

Speaker ID. The speaker ID is usually composed of three letters. The code may be based either on the participant's name, as in *ROS or *BIL, or on her role, as in *CHI or *MOT. In corpora studying single children, the form *CHI should always be used for the Target_Child, as in this example.

```
@Participants: CHI Mark Target_Child, MOT Mary Mother
```

Several different Target_Child participants can be indicated as *CH1, *CH2, *CH3. However, if one is primary, it should be *CHI. There are several CHILDES corpora that use first name abbreviations for target children, because they are studies of siblings or group sessions. These corpora include FernAguado, Koine, Becasesno, Palasis, Luque, Weissenborn, Gathercole, Guilfoyle, Garvey, Evans, Levy, Navracsics, MCF, and Ionin.

Name and/or Specific Role. The speaker's name can be placed in the field after the 3-letter ID. However, this field can be omitted, particularly if it is important to deidentify the data. If CLAN finds only a three-letter ID and a role, it will assume that the name has been omitted. In order to preserve anonymity, it is often useful to include a pseudonym for the name, because the pseudonym will also be used in the body of the transcript. For CLAN to correctly parse the participants line, multiple-word name definitions such as "Sue Day" need to be joined in the form "Sue_Day." Instead of putting in the name, you can put in a specific role, such as Maternal_Grandmother. The name can be combined with the specific role in this way:

```
@Participants:  ROS Rose_Maternal_Grandmother Grandparent
```

Standard Role. After the ID and name, the last field gives the standard role of the speaker. There is a fixed set of standard roles specified in the depfile.cut file used by CHECK. You will also see this same list of possible roles in the "role" segment of the "ID Headers" dialog box. All of these roles are hard-wired into the depfile.cut file used by CHECK. If one of these standard roles does not work, it would be best to use one of the generic age-related roles, like Adult, Child, or Teenager.

Further details regarding the specific role can be put in the place of the name in the field before the role, as in these examples:

```
@Participants:  TBO Toll_Booth_Operator Adult,AIR Airport_Attendant
    Adult, SI1 First_Sibling Sibling, SI2 Second_Sibling Sibling, COM
    Computer_Talk Media
```

Note that the terms in the second field, such as *Toll_Booth_Operator* or *Second_Sibling* must be written as a single word by using underscores to link separate words. The following is a list of the roles currently in depfile.cut. Although we occasionally add more roles, we try to limit this by using the following standard roles:

Target_Child: Use of this role is very important for CHILDES and PhonBank transcripts, because it allows users to search and analyze the output from the children who are the focus of many of the studies.

Target_Adult: This role serves a similar function to Target_Child by making it clear who which speaker was at the focus of the data collection.

Child: This role is used mostly in transcripts studying large groups of children, when it is not easy to determine whether a child is a boy or girl or perhaps a relative.

Mother: This should be the mother of the Target_Child.

Father: This should be the father of the Target_Child.

Brother: This should be a brother of the Target_Child.

Sister: This should be a sister of the Target_Child.

Sibling: This should be a sibling of the Target_Child.

Grandfather: This should be the grandfather of the Target_Child. Further details such as Paternal_Grandfather can be placed into the Specific Role field.

Grandmother: This should be the grandmother of the Target_Child. Further details such as Paternal_Grandmother can be placed into the Specific Role field.

Relative: This role is designed to include all other relations, including *Aunt, Uncle, Cousin, Father_in_Law* etc. which can then be entered into the Specific Role field.

Participant: This is the generic role for adult participants in interviews and other conversations. Usually, these are coded as having a Participant and an Investigator. Other forms of this role include Patient, Informant, and Subject which can be listed in the Specific Role field or else just omitted.

Investigator: Other terms for this role can be listed in the Specific Roles. These include Researcher, Clinician, Therapist, Camera_Operator, and so on.

Partner: This is the role for the person accompanying the Participant to the interview or conversation.

Boy: This is a generic role.

Girl: This is a generic role.

Adult: This is a very generic role for use when little else is known.

Teenager: This is a generic role.

Male: Use this role when all we know is that the participant is an adult male.

Female: Use this role when all we know is that the participant is an adult female.

Visitor: This role assumes that the visitor is coming to a conversation in the home.

Friend: This is a role for a Friend of the target participants.

Playmate: This is a role for a child that the Target_Child plays with.

Caretaker: This person takes care of the child. Other names for the Specific Role field include Housekeeper, Nursemaid, or Babysitter.

Environment: This role is used in the SBCSAE corpus.
Group: This role is used when transcribing simultaneous productions from a whole group.
Unidentified: This is a role for unidentifiable participants.
Uncertain: This role can be used when it is not clear who produced an utterance.
Other: This is a generic role. When it is used, there should be further specification in the Specific Role field. Roles defined by jobs such as Technician, Patron, Policeman, etc can be listed as Other and the details given in the Specific Role field.
Text: This role is used for written segments of TalkBank.
Media: This role is used for speech from televisions, computers, or talking toys.
PlayRole: This role is used when speakers pretend to be something, such as an animal or another person.
LENA: This role is used in HomeBank LENA recordings. The specific LENA role is then listed in the Specific Role field.
Justice: This is role is used in the SCOTUS corpus. It also includes the role of Judge.
Attorney: This is the general role for attorneys, lawyers, prosecutors, etc.
Doctor: This is the general role for doctors.
Nurse: This is the general role for nurses.
Student: Specific forms of this general role include Graduate Student, Senior, High_Schooler, and so on.
Teacher: This is the general role for Teachers. Specific forms of this general role include Instuctor, Advisor, Faculty, Professor, Tutor, or T_A.
Host: Specific forms of this general role include ShowHost, Interviewer, and CallTaker.
Guest: Specific forms of this general role include ShowGuest, Interviewee, and Caller.
Leader: Specific forms of this general role include Group_Leader, Panel_Moderator, Committee_Chair, Facilitator, Tour_Guide, Tour_Leader, Peer_Leader, Chair, or Discussion_Leader.
Member: Specific forms of this general role include Committee_Member, Group_Member, Panelist, and Tour_Participant.
Narrator: This is a role for presentations of stories.
Speaker: Specific forms of this general role include Lecturer, Presenter, Introducer, Welcomer, and Main_Speaker.
Audience: This is the general role for single audience members.

@Options:

This header is not obligatory, but it is frequently needed. When it occurs, it must follow the @Participants line. This header allows the checking programs (CHECK and the XML validator) to suspend certain checking rules for certain file types. The spelling of these options is case-sensitive.

1. **heritage**: Use of this option tells CHECK and the validator not to look at the content of the main lines at all. This radical blockage of the function of CHECK is only recommended for people working with CA files done in the traditional Jeffersonian format. When this option is used, text may be placed into italics, as in traditional CA.
2. **IPA**: Use of this option permits the use of IPA notation on the main line.

3. **CA**: Use of this option suspends the usual requirement for utterance terminators to accommodate Conversation Analysis transcripts.
4. **CA-Unicode**: This option is needed for CA transcripts using East Asian scripts in order to automatically load Arial Unicode instead of CAFont. Unfortunately, overlap alignment is accurate for a variable-width font like Arial Unicode.
5. **multi:**. Use of this option tells CHECK and Chatter to expect multiple bullets on a single line. This can be used for data that come from programs like Praat that mark time for each word.
6. **bullets**: This option turns off the requirement that each time-marking bullet should begin after the previous one.
7. **dummy**: This option is used in files that point to media that do not yet have any transcription.
8. **notarget:** This option tells CHECK not to look for the presence of a Target_Child in the @Participants line.

@ID:

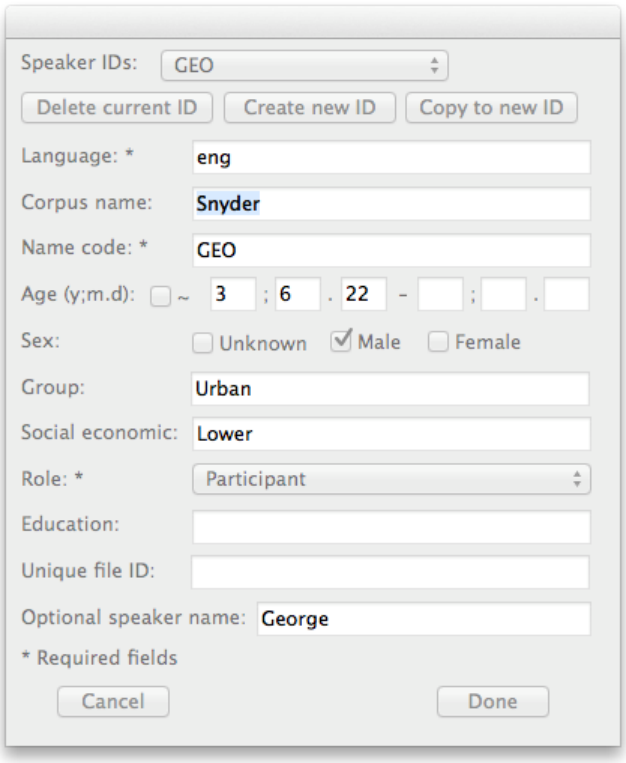
This header is used to control programs such as STATFREQ, output to Excel, and new programs based on XML. The form of this line is:

```
@ID: language|corpus|code|age|sex|group|SES|role|education|custom|
```

There must be one @ID field for each participant. Often you will not care to encode all of this information. In that case, you can leave some of these fields empty. Here is a typical @ID header.

```
@ID:	en|macwhinney|CHI|2;10.10||||Target_Child|||
```

To facilitate typing of these headers, you can run the CHECK program on a new CHAT file. If CHECK does not see @ID headers, it will use the @Participants line to insert a set of @ID headers to which you can then add further information. Alternatively, you can use the INSERT program to create these fields automatically from the information in the @Participants line. For even more complete control over creation of these @ID headers, you can use the dialog system that comes up when you have an open CHAT file and select “ID Headers” under the Tiers Menu pulldown. Here is a sample version of this dialog box:



Here are some further specifications of the codes in the fields for the @ID header.

Language:	as in the ISO codes table given above
Corpus:	a one-word label for the corpus in lowercase
Code:	the three-letter code for the speaker in capitals
Age:	the age of the speaker (see below)
Sex:	either “male” or “female” in lowercase
Group:	any single word label. Common abbreviations include: LI language impaired, LT late talker, SLI specific language impaired, TD typically developing, ASD, RHD, AD, TBI.
Eth, SES:	Ethnicity (Asian, Black, Latino, Multiple, Native, Pacific, Unknown, White), SES (WC for working class, UC for upper class, MC for middle class, LI for limited income) Note: if both Ethnicity and SES are given, there is a comma separating them. If Ethnicity is not listed, it is assumed to be White.
Role:	the role as given in the @Participants line
Education:	educational level of the speaker (or the parent): Elem, HS, UG, Grad, Doc
Custom:	any additional information needed for a given project

It is important to use the correct format for the Target_Child’s age. This field uses the form years;months.days as in 2;11.17 for 2 years, 11 months, and 17 days. The fields for the months and days should always have two places. Using this format is important when it comes to ordering data by age in spreadsheet systems such as Excel. This often

means that you need to add leading zeroes, as in 2;05.06 and 5;09.01. However, you do not need to add any leading zeroes before the years. If you do not know the child's age in days, you can simply use years and months, as in 6;04. with a period after the months. If you do not know the months, you can use the form 6; with the semicolon after the years. If you only know the child's birthdate and the date of the transcript, you can use the DATES program to compute the child's age.

@Media:

This header is used to tell CLAN how to locate and play back media that are linked to transcripts. The first field in this header specifies the name of the media file. Extensions should be omitted. If the media file is abe88.wav, then just enter "abe88". Then declare the format as "sound" or "video". It is also possible to add one of the three terms *missing, notrans*, or *unlinked* after the media type. The term *missing* is used when the media is missing from the collection. The term *unlinked* is used for transcripts that have not yet been linked to media. The term *notrans* is used for media that have not yet been transcribed. So the line has this shape:

```
@Media:   abe88, sound, missing
```

@Videos:

This header is used specifically by the function that allows you to shift between different camera angles on the same interaction, as illustrated and described in the section on "Multiple Video Playback" in the CLAN manual.

@End

Like the @Begin header, this header uses no colon and takes no entry. It is placed at the end of the file as the very last line. Adding this header provides a safeguard against the danger of undetected file truncation during copying.

7.3 Participant-Specific Headers

The third set of headers provides information specific to each participant. Most of the participant-specific information is in the @ID tier. That information can be entered by using the ID headers option in CLAN's Tiers menu. The exceptions are for these tiers:

```
@Birth of #:
@Birthplace of #:
@L1 of #:
```

7.4 Constant Headers

Currently, the constant headers follow the participant-specific headers. However, once the participant-specific headers have been merged into the @ID fields, the constant headers will follow the @Media field. These headers, which are all optional, describe various general facts about the file.

@Location:

This header should include the city, state or province, and country in which the interaction took place. Here is an example of a completed header line:

```
@Location: Boston, MA, USA
```

@Number:

The possible entries here include: 1 2 3 4 5 more audience

@Recording Quality:

Possible entries here are: 1, 2, 3, 4, 5 with 5 being the highest quality.

@Room Layout:

This header outlines room configuration and positioning of furniture. This is especially useful for experimental settings. The entry should be a description of the room and its contents. Here is an example of the completed header line:

```
@Room Layout:   Kitchen; Table in center of room with window on west
    wall, door to outside on north wall
```

@Tape Location:

This header indicates the specific tape ID, side and footage. This is very important for identifying the spot on the analog tape from which the transcription was made. The entry for this header should include the tape ID, side and footage. Here is an example of this header:

```
@Tape Location: tape74, side a, 104
```

@Time Duration:

It is often necessary to indicate the time at which the audiotaping began and the amount of time that passed during the course of the taping, as in the following header:

```
@Time Duration: 12:30-13:30
```

This header provides the absolute time during which the taping occurred. For most projects what is important is not the absolute time, but the time of individual events relative to each other. This sort of relative timing can provided by coding on the %tim dependent tier in conjunction with the @Time Start header described next. However, this type of coding is really only needed for older transcripts for which there is no media. None of the CLAN programs make use of this information.

@Time Start:

If you are tracking elapsed time on the %tim tier, the @Time Start header can be used to indicate the absolute time at which the timing marks begin. If a new @Time Start header is placed in the middle of the transcript, this "restarts" the clock. This method is really only appropriate for older transcripts for which there is no media. Transcripts linked to media will not need this information. None of the CLAN programs make use of this information.

```
@Time Start: 12:30
```

@Transcriber: *

This line identifies the people who transcribed and coded the file. Having this indicated is often helpful later, when questions arise. It also provides a way of acknowledging the people who have taken the time to make the data available for further study.

@Transcription:

The possible entries here are: eye_dialect, partial, full, detailed, coarse, checked

@Warning:

This header is used to warn the user about certain defects or peculiarities in the collection and transcription of the data in the file. Some typical warnings are as follows:

1. These data are not useful for the analysis of overlaps, because overlapping was not accurately transcribed.
2. These data contain no information regarding the context. Therefore they will be inappropriate for many types of analysis.
3. Retracings and hesitation phenomena have not been accurately transcribed in these data.
4. These data have been transcribed, but the transcription has not yet been double-checked.
5. This file has not yet passed successfully through CHECK.

7.5 Changeable Headers

Changeable headers can occur either at the beginning of the file along with the constant headers or else in the body of the file. Changeable headers contain information that can change within the file. For example, if the file contains material that was recorded on only one day, the @Date header would occur only once at the beginning of the file. However, if the file contains some material from a later day, the @Date header would be used again later in the file to indicate the next date. These changeable headers appear, then, at the point within the file where the information changes. The list that follows is alphabetical.

@Activities:

This header describes the activities involved in the situation. The entry is a list of component activities in the situation. Suppose the @Situation header reads, "Getting ready to go out." The @Activities header would then list what was involved in this, such as putting on coats, gathering school books, and saying good-bye.

@Bck:

Diary material that was not originally transcribed in the CHAT format often has explanatory or background material placed before a child's utterance. When converting this material to the CHAT format, it is sometimes impossible to decide whether this background material occurs before, during, or after the utterance. In order to avoid having to make these decisions after the fact, one can simply enter it in an @Bck header.

```
@Bck:     Rachel was fussing and pointing toward the cabinet where
    the cookies are stored.
*RAC:     cookie [/] cookie.
```

@Bg and @Bg:

These headers are used to mark the beginning of a "gem" for analysis by GEM. If there is a colon, you must follow the colon with a tab and then one or more code words.

@Blank

This header is created by the TEXTIN program. It is used to represent the fact that some written text includes a blank line or new paragraph. It should not be used for transcripts of spoken language.

@Comment:

This header can be used as an all-purpose comment line. Any type of comment can be entered on an @Comment line. When the comment refers to a particular utterance, use the %com line. When the comment refers to more general material, use the @Comment header. If the comment is intended to apply to the file as a whole, place the @Comment header along with the constant headers before the first utterance. Instead of trying to make up a new coding tier name such as "@Gestational Age" for a special purpose type of information, it is best to use the @Comment field, as in this example:

```
@Comment:  Gestational age of MAR is 7 months
@Comment:  Birthweight of MAR is 6 lbs. 4 oz
```

Another example of a special @Comment field is used in the diary notes of the MacWhinney corpus, where they have this shape:

```
@Comment:  Diary-Brian – Ross said "I don't need to throw my blocks
    out the window anymore."
```

@Date:

This header indicates the date of the interaction. The entry for this header is given in the form day-month-year. The date is abbreviated in the same way as in the @Birth header entry. Here is an example of a completed @Date header line:

```
@Date: 01-JUL-1965
```

Because we have some corpora going back over a century, it is important to include the full value for the year. Also, because the days of the month should always have two digits, it is necessary to add a leading "0" for days such as "01".

@Eg and @Eg:

These headers are used to mark the end of a "gem" for analysis by the GEM command. If there is a colon, you must follow the colon with a tab and then one or more code words. Each @Eg must have a matching @Bg. If the @Eg: form is used, then the text following it must exactly match the text in the corresponding @Bg: You can nest one set of @Bg-@Eg markers inside another, but double embedding is not allowed. You can also begin a new pair before finishing the current one, but again this cannot be done for three beginnings.

@G:

This header is used in conjunction with the GEM program, which is described in the CLAN manual. It marks the beginning of "gems" when no nesting or overlapping of gems occurs. Each gem is defined as material that begins with an @G marker and ends with the next @G marker. We refer to these markers as "lazy" gem markers, because they are easier to use than the @Bg: and @Eg: markers. To use this feature, you need to also use the +n switch in GEM. You may nest at most one @Bg-@Eg pair inside a series of @G headers. As with the @Bg and @Eg markers, this code can either be used alone without a colon or else used with a colon followed by a tab and some following code for later specific retrieval.

@New Episode

This header simply marks the fact that there has been a break in the recording and that a new episode has started. It is a "bare" header that is used without a colon, because it takes no entry. There is no need to mark the end of the episode because the @New Episode header indicates both the end of one episode and the beginning of another.

@Page:

This header is used to indicate the number of the page from which some text is taken. It should not be used for spoken texts.

@Situation:

This changeable header describes the general setting of the interaction. It applies to all the material that follows it until a new @Situation header appears. The entry for this header is a standard description of the situation. Try to use standard situations such as: “breakfast,” “outing,” “bath,” “working,” “visiting playmates,” “school,” or “getting ready to go out.” Here is an example of the completed header line:

```
@Situation:  Tim and Bill are playing with toys in the hallway.
```

There should be enough situational information given to allow the user to reconstruct the situation as much as possible. Who is present? What is the layout of the room or other space? What is the social role of those present? Who is usually the caregiver? What activity is in progress? Is the activity routinized and, if so, what is the nature of the routine? Is the routine occurring in its standard time, place, and personnel configuration? What objects are present that affect or assist the interaction? It will also be important to include relevant ethnographic information that would make the interaction interpretable to the user of the database. For example, if the text is parent-child interaction before an observer, what is the culture's evaluation of behaviors such as silence, talking a lot, displaying formulaic skills, defending against challenges, and so forth?

8 Words

Words are the basic building blocks for all sentential and discourse structures. By studying the development of word use, we can learn an enormous amount about the growth of syntax, discourse, morphology, and conceptual structure. However, in order to realize the full potential of computational analysis of word usage, we need to follow certain basic rules. In particular, we need to make sure that we spell words in a consistent manner. If we sometimes use the form *doughnut* and sometimes use the form *donut*, we are being inconsistent in our representation of this particular word. If such inconsistencies are repeated throughout the lexicon, computerized analysis will become inaccurate and misleading. One of the major goals of CHAT analysis is to maximize systematicity and minimize inconsistency. In the Introduction, we discussed some of the problems involved in mapping the speech of language learners onto standard adult forms. This chapter spells out some rules and heuristics designed to achieve the goal of consistency for word-level transcription.

One solution to this problem would be to avoid the use of words altogether by transcribing everything in phonetic or phonemic notation. But this solution would make the transcript difficult to read and analyze. A great deal of work in language learning is based on searches for words and combinations of words. If we want to conduct these lexical analyses, we have to try to match up the child's production to actual words. Work in the analysis of syntactic development also requires that the text be analyzed in terms of lexical items. Without a clear representation of lexical items and the ways that they diverge from the adult standard, it would be impossible to conduct lexical and syntactic analyses computationally. Even for those researchers who do not plan to conduct lexical analyses, it is extremely difficult to understand the flow of a transcript if no attempt is made to relate the learner's sounds to items in the adult language.

At the same time, attempts to force adult lexical forms onto learner forms can seriously misrepresent the data. The solution to this problem is to devise ways to indicate the various types of divergences between learner forms and adult standard forms. Note that we use the term “divergences” rather than “error.” Although both learners (MacWhinney & Osser, 1977) and adults (Stemberger, 1985) clearly do make errors, most of the divergences between learner forms and adult forms are due to structural aspects of the learner's system.

This chapter discusses the various tools that CHAT provides to mark some of these divergences of child forms from adult standards. The basic types of codes for divergences that we discuss are:

special learner-form markers,

codes for unidentifiable material,

1. codes for incomplete words,
2. ways of treating formulaic use of words, and
3. conventions for standardized spellings.

For languages such as English, Spanish, and Japanese, we now have complete MOR grammars. The lexicons used by these grammars constitute the definitive current CHAT standard for words. Please take a look at the relevant lexical files, since they illustrate in great detail the overall principles we are describing in this chapter.

8.1 The Main Line

The word forms we will be discussing here are the principal components of the "main line." This line gives the basic transcription of what the speaker said. The structure of main lines in CHAT is fairly simple. Each main tier line begins with an asterisk. After the asterisk, there is a three-letter speaker ID, a colon and a tab. The transcription of what was said begins in the ninth column, after the tab, because the tab stop in the editor is set for the eighth column. The remainder of the main tier line is composed primarily of a series of words. Words are defined as a series of ASCII characters separated by spaces. In this chapter, we discuss the principles governing the transcription of words.

In CLAN, all characters that are not punctuation markers are potentially parts of words. The default punctuation set includes the space and these characters:

```
, . ; ? ! [ ] < >
```

None of these characters or the space can be used within words. This punctuation set applies to the main lines and all coding lines with the exception of the %pho and %mod lines which use the system described in the chapter on Dependent Tiers. Because those systems make use of punctuation markers for special characters, only the space can be used as a delimiter on the %pho and %mod lines. As the CLAN manual explains, this default punctuation set can be changed for particular analyses.

Other non-letter characters can be used within words to express special meanings. These include the various marks in the section on CA coding, as well as these:

```
+  _  - @ ( )
```

8.2 Basic Words

Main lines are composed of words and other markers. Words are pronounceable forms, surrounded by spaces. Most words are entered just as they are found in the dictionary. The first word of a sentence is not capitalized, unless it is a proper noun or a word normally capitalized by itself, such as a noun in German or the word "I" in English.

8.3 Special Form Markers

Special form markers can be placed at the end of a word. To do this, the symbol "@" is used in conjunction with one or two additional letters. Here is an example of the use of the @ symbol:

```
*SAR:     I got a bingbing@c.
```

Here the child has invented the form *bingbing* to refer to a toy. The word *bingbing* is not in the dictionary and must be treated as a special form. To further clarify the use of these @c forms, the transcriber should create a file called "0lexicon.cdc" that provides glosses for such forms.

The @c form illustrated in this example is only one of many possible special form markers that can be devised. The following table lists some of these markers that we have found useful. However, this categorization system is meant only to be suggestive, not exhaustive. Researchers may wish to add further distinctions or ignore some of the categories listed. The particular choice of markers and the decision to code a word with a marker form is one that is made by the transcriber, not by CHAT. The basic idea is that

CLAN will treat words marked with the special learner-form markers as words and not as fragments. In addition, the MOR program will not attempt to analyze special forms for part of speech, as indicated in the final column in this table.

Special Form Markers

Letters	Categories	Example	Meaning	POS
@a	addition	xxx@a	unintelligible	w
@b	babbling	abame@b	-	bab
@c	child-invented form	gumma@c	sticky	chi
@d	dialect form	younz@d	you	dia
@e	echolalia, repetition	want@e more@e	want more	skip
@f	family-specific form	bunko@f	broken	fam
@g	general special form	gongga@g	-	skip
@i	interjection, interaction	uhhuh@i	-	co
@k	multiple letters	ka@k	Japanese “ka”	n:let
@l	letter	b@l	letter b	n:let
@n	neologism	breaked@n	broke	neo
@o	onomatopoeia	woofwoof@o	dog barking	on
@p	phonol. consistent form	aga@p	-	phon
@q	metalinguistic use	no if@q-s or but@q-s	when citing words	meta
@s:*	second-language form	istenem@s:hu	Hungarian word	L2
@s$n	second-language noun	perro@s$n	Spanish noun	n|
@si	singing	lalala@si	singing	sing
@sl	signed language	apple@sl	apple	sign
@sas	sign & speech	apple@sas	apple and sign	sas
@t	test word	wug@t	test	test
@u	Unibet transcription	binga@u	-	uni
@wp	word play	goobarumba@wp	-	wp
@x	excluded words	stuff@x	excluded	unk
@z:xxx	user-defined code	word@z:rtfd	any user code	

We can define these special markers in the following ways:

Addition can be used to mark an unintelligible string as a word for inclusion on the %mor line. MOR then recognizes xxx@a as w|xxx. It also recognizes xxx@a$n as, for example n|xxx. Adding this feature will still not allow inclusion of sentences with unintelligible words for MLU and DSS, because the rules for those indices prohibit this. In most cases, researchers prefer to simply mark unintelligible forms as xxx without the additional @a.

Babbling can be used to mark both low-level early babbling. These forms have no obvious meaning and are used just to have fun with sound.

Child-invented forms are words created by the child sometimes from other words without obvious derivational morphology. Sometimes they appear to be sound variants of other words. Sometimes their origin is obscure. However, the child appears to be

convinced that they have meaning and adults sometimes come to use these forms themselves.

Dialect form is often an interesting general property of a transcript. However, the coding of phonological dialect variations on the word level should be minimized, because it often makes transcripts more difficult to read and analyze. Instead, general patterns of phonological variation can be noted in the readme file.

Echolalia form can be marked for individual words. If a whole utterance is echoed, then it is better to use the [+ imit] postcode.

Family-specific forms are much like child-invented forms that have been taken over by the whole family. Sometimes the source of these forms are children, but they can also be older members of the family. Sometimes the forms come from variations of words in another language. An example might be the use of *undertoad* to refer to some mysterious being in the surf, although the word was simply *undertow* initially.

General special form marking with @g can be used when all of the above fail. However, its use should generally be avoided. Marking with the @ without a following letter is not accepted by CHECK.

Interjections can be indicated in standard ways, making the use of the @i notation usually not necessary. Instead of transcribing "ahem@i," one can simply transcribe *ahem* following the conventions listed later.

Letters can either be transcribed with the @l marker or simply as single-character words. If it is necessary to mark a letter name as plural, it is possible to add a suffix, as in m@l-s.

Multiple letters or strings of letters are marked as @k (as in "kana").

Neologisms are meant to refer to morphological coinages. If the novel form is monomorphemic, then it should be characterized as a child-invented form (@c), family-specific form (@f), or a test word (@t). Note that this usage is only really sanctioned for CHILDES corpora. For AphasiaBank corpora, neologisms are considered to be forms that have no real word source, as is typical in jargon aphasia. If you want to indicate the part of speech for a neologism, you can use a coding like dumpf@n$v to indicate that *dumpf* is intended to be a verb. This can be helpful for %mor coding.

Nonvoiced forms are produced typically by hearing-impaired children or their parents who are mouthing words without making their sounds.

Onomatopoeias include animal sounds and attempts to imitate natural sounds.

Phonological consistent forms (PCFs) are early forms that are phonologically consistent, but whose meaning is unclear to the transcriber. Often these forms are protomorphemes.

Quoting or Metalinguistic reference can be used to either cite or quote single standard words or special child forms.

Second-language forms derive from some language not usually used in the home. These are marked with a second letter for the first letter of the second language, as in @s:zh for Mandarin words inside an English sentence.

Part of speech codes. You can also mark the part of speech of a second language word by using the form @s$ as in perro@s$n to indicate that the Spanish word *perro* (dog) is a noun. You can use the same method without the @s for L1 words. Thus, the form goodbyes$n will be recognized as n|goodbyes. Also, you can use this method with

other special form markers. So, bimp@c$adj would indicate that *bimp* is a child-invented form that is functioning as an adjective.

Sign language use can be indicated by the @sl.

Sign and speech use involves making a sign or informal sign in parallel with saying the word.

Singing can be marked with @si. Sometimes the phrase that is being sung involves nonwords, as in lalaleloo@si. In other cases, it involves words that can be joined by underscores. However, if a larger passage is sung, it is best to transcribe it as speech and just mark it as being sung through a comment line.

Test words are nonce forms generated by the investigators to test the productivity of the child's grammar.

Unibet transcription can be given on the main line by using the @u marker. However, if many such forms are being noted, it may be better to construct a @pho line. With the advent of IPA Unicode, we now prefer to avoid the use of Unibet, relying instead directly on IPA.

Word play in older children produces forms that may sound much like the forms of babbling, but which arise from a slightly different process. It is best to use the @b for forms produced by children younger than 2;0 and @wp for older children.

Excluded forms can be marked with @x.

User-defined special forms can be marked with @z followed by up to five letters of a user-defined code, such as in word@z:rftd. This format should be used carefully, because it will be difficult for the MOR program to evaluate words with these codes unless additional detailed information is added to the sf.cut file.

The @b, @u, and @wp markers allow the transcriber to represent words and babbling words phonologically on the main line and have CLAN treat them as full lexical items. This should only be done when the analysis requires that the phonological string be treated as a word and it is unclear which standard morpheme corresponds to the word. If a phonological string should not be treated as a full word, it should be marked by a beginning &, and the @b or @u endings should not be used. Also, if the transcript includes a complete %pho line for each word and the data are intended for phonological analysis, it is better to use yy (see the next section) on the main line and then give the phonological form on the %pho line. If you wish to omit coding of an item on the %pho line, you can insert the horizontal ellipsis character … (Unicode character number 2026). This is a single character, not three periods, and it is not the ellipsis character used by MS-Word.

Family-specific forms are special words used only by the family. These are often derived from child forms that are adopted by all family members. They also include certain "caregiverese" forms that are not easily recognized by the majority of adult speakers but which may be common to some areas or some families. Family-specific forms can be used by either adults or children.

The @n marker is intended for morphological neologisms and over-regularizations, whereas the @c marker is intended to mark nonce creation of stems. Of course, this distinction is somewhat arbitrary and incomplete. Whenever a child-invented form is clearly onomatopoeic, use the @o coding instead of the @c coding. A fuller

characterization of neologisms can be provided by the error coding system presented in a separate chapter.

8.4 Unidentifiable Material

Sometimes it is difficult to map a sound or group of sounds onto either a conventional word or a non-conventional word. This can occur when the audio signal is so weak or garbled that you cannot even identify the sounds being used. At other times, you can recognize the sounds that the speaker is using, but cannot map the sounds onto words. Sometimes you may choose not to transcribe a passage, because it is irrelevant to the interaction. Sometimes the person makes a noise or performs an action instead of speaking, and sometimes a person breaks off before completing a recognizable word. All of these problems can be dealt with by using certain special symbols for those items that cannot be easily related to words. These symbols are typed in lower case and are preceded and followed by spaces. When standing alone on a text tier, they should be followed by a period, unless it is clear that the utterance was a question or a command.

Unintelligible Speech xxx

Use the symbol xxx when you cannot hear or understand what the speaker is saying. If you believe you can distinguish the number of unintelligible words, you may use several xxx strings in a row. Here is an example of the use of the xxx symbol:

```
*SAR:     xxx .
*MOT:     what ?
*SAR:     I want xxx .
```

Sarah's first utterance is fully unintelligible. Her second utterance includes some unintelligible material along with some intelligible material. The MLU and MLT commands will ignore the xxx symbol when computing mean length of utterance and other statistics. If you want to have several words included, use as many occurrences of xxx as you wish.

Phonological Coding yyy

Use the symbol yyy when you plan to code all material phonologically on a %pho line. If you are not consistently creating a %pho line in which each word is transcribed in IPA in the order of the main line, you should use the @u or & notations instead. Here is an example of the use of yyy:

```
*SAR:     yyy yyy a ball .
%pho:     ta gə ə bal
```

The first two words cannot be matched to particular words, but their phonological form is given on the %pho line.

Untranscribed Material www

This symbol must be used in conjunction with an %exp tier which is discussed in the chapter on dependent tiers. This symbol is used on the main line to indicate material that

a transcriber does not know how to transcribe or does not want to transcribe. For example, it could be that the material is in a language that the transcriber does not know. This symbol can also be used when a speaker says something that has no relevance to the interactions taking place and the experimenter would rather ignore it. For example, www could indicate a long conversation between adults that would be superfluous to transcribe. Here is an example of the use of this symbol:

```
*MOT:     www.
%exp:     talks to neighbor on the telephone
```

Phonological Fragments **&**

Disfluencies such as fillers, phonological fragments, and repeated segments are all coded by a preceding &. More specifically, &- may be used for fillers and &+ for fragments (please see the chapter on disfluency coding for the details). Material following the ampersand symbol will be ignored by certain CLAN commands, such as MLU, which computes the mean length of the utterance in a transcript. If you want a command such as FREQ to count all of the instances of phonological fragments, you would have to add a switch such as +s"&*" (or +s"&+*").

8.5 Incomplete and Omitted Words

Words may also be incomplete or even fully omitted. We can judge a word to be incomplete when enough of it is produced for us to be sure what was intended. Judging a word to be omitted is often much more difficult.

Noncompletion of a Word **text(text)text**

When a word is incomplete, but the intended meaning seems clear, insert the missing material within parentheses. Do not use this notation for fully omitted words, only for words with partial omissions. This notation can also be used to derive a consistent spelling for commonly shortened words, such as *(un)til* and *(be)cause*. CLAN will treat items that are coded in this way as full words. For programs such as FREQ, the parentheses will essentially be ignored and *(be)cause* will be treated as if it were *because*. The CLAN programs also provide ways of either including or excluding the material in the parentheses, depending on the goals of the analysis.

```
*RAL:     I been sit(ting) all day .
```

The inclusion or exclusion of material enclosed in parentheses is well supported by CLAN and this same notation can also be used for other purposes when necessary. For example, studies of fluency may find it convenient to code the number of times that a word is repeated directly on that word, as in this example with three repetitions of the word *dog*.

```
JEF:      that's a dog [x 3].
```

By default, the programs will remove the [x 3] form and the sentence will be treated as a three word utterance. This behavior can be modified by using the +r switch.

Omitted Word **0word**

The coding of word omissions is a difficult and unreliable process. Many researchers will prefer not to even open up this particular can of worms. On the other hand, researchers in language disorders and aphasia often find that the coding of word omissions is crucial to particular theoretical issues. In such cases, it is important that the coding of omitted words be done in as clear a manner as possible.

To code an omission, the zero symbol is placed before a word on the text tier. The full omission of a word always be coded in this way and not through the use of parentheses. If what is important is not the actual word omitted, but the part of speech, then a code for the part of speech can follow the zero. Similarly, the identity of the omitted word is always a guess. The best guess is placed on the main line. Here is an example of its use:

```
*EVE:     I want 0to go.
```

It is very difficult to know when a word has been omitted. However, the following criteria can be used to help make this decision for English data:

1. 0det: Unless there is a missing plural, a common noun without an article is coded as 0det.
2. 0v: Sentences with no verbs can be coded as having missing verbs. Of course, often the omission of a verb can be viewed as a grammatical use of ellipsis.
3. 0aux: In standard English, sentences like "he running" clearly have a missing auxiliary.
4. 0subj: In English, every finite verb requires a subject.

In English, there are seldom solid grounds for assigning codes like 0adj, 0adv, 0obj, or 0prep. However, these codes are possible. In addition, some researchers think that in some contexts they can know exactly what words are being omitted. For example, they may mark forms such as 0person, 0spot, and so on. Making such markings is possible, although we would rather see codes at the level of 0v or 0det. Items marked as omitted are not included in the MLU count.

8.6 Standardized Spellings

There are a number of common words in the English language that cannot be found in the dictionary or whose lexical status is vague. For example, how should letters be spelled? What about numbers and titles? What is the best spelling – *doggy* or *doggie*, *yeah* or *yah*, and *pst* or *pss*? If we can increase the consistency with which such forms are transcribed, we can improve the quality of automatic lexical analyses. CLAN commands such as FREQ and COMBO provide output based on searches for particular word strings. If a word is spelled in an indeterminate number of variant ways, researchers who attempt to analyze the occurrence of that word will inevitably end up with inaccurate results. For example, if a researcher wants to trace the use of the pronoun *you*, it might be necessary to search not only for *you*, *ya*, and *yah*, but also for all the assimilations of the pronouns with verbs such as *didya/dicha/didcha* or *couldya/couldcha/coucha*. Without a standard set of rules for the transcription of such forms, accurate lexical searches could become impossible. On the other hand, there is no reason to avoid using these forms if a

set of standards can be established for their use. Other programs rely on the use of dictionaries of words. If the spellings of words are indeterminate, the analyses produced will be equally indeterminate. For that reason, it is helpful to specify a set of standard spellings for marginal words. If you have doubts about the spellings of certain words, you can look in the 0allwords.cdc file this is included in the /lex folder of the MOR gramar for each language. The words there are listed in alphabetical order..

8.6.1 Letters

To transcribe letters, use the @l symbol after the letter. For example, the letter "b" would be b@l. Here is an example of the spelling of a letter sequence.

```
*MOT:     could you please spell your name ?
*MAR:     it's m@l a@l r@l k@l .
```

The dictionary says that "abc" is a standard word, so that is accepted without the @l marking. In Japanese, many letters refer to whole syllables or "kana" such as *ro* or *ka*. To represent this as well as strings of letters in English, use the @k symbol, as in ka@k or jklmn@k. Using this form, the above example could better be coded as:

```
*MOT:     could you please spell your name?
*MAR:     it's mark@k.
```

However, in this case, the spelling is counted as one word, not four.

8.6.2 Compounds and Linkages

Languages use a variety of methods for combining words into larger lexical items. One method involves inflectional processes, such as cliticization and affixation, that will be discussed later. Here we consider compounds and linkages. In earlier versions of CLAN, it was necessary to write compounds in the form of *bird+house* and *baby+sitter*, but now the plus is no longer necessary. You can just write *birdhouse* and *babysitter* and the correct form will be inserted into the %mor line by the MOR program.

A second level of concatenation involves the use of an underscore to indicate the fact that a phrasal combination is not really a compound, but what we call a "linkage". Common examples here include titles of books such as Green_Eggs_and_Ham, appellations such as Little_Bo_Beep or Santa_Claus, lines from songs such as The_Farmer_in_the_Dell, and places such as Hong_Kong_University. For these forms, the underscore is used to emphasize the fact that, although the form is collocational, it does not obey standard rules of compound formation. Because these forms all begin with a capital letter, the morphological analyzer will recognize them as proper nouns.

The underscore is used for two other purposes. First, it can be used for irregular combinations, such as how_about and how_come. Second, it can be used on the %mor line to represent a multiword English gloss for a single stem, as in "lose_flowers" for *defleurir*. When acronyms are themselves proper nouns, they can be written with full capitalization, as in TGV or CIA without any need to add underscores, because words beginning with capital letters are always assumed to be proper nouns.

The third form of concatenation involves the use of hyphens in words such as cul-de-sac or hi-fi. These words are customarily written with hyphens and that is the way they should be transcribed on the main line. For English, these words are listed in files such as

n-hyphen.cut. Hyphens should only be used if the words involved are customarily written with hyphens.

Unfortunately, the hyphen is also used on the %mor line to indicate suffixation, as in n|dog-PL for *dogs*. To eliminate this confusion, when MOR runs, it changes the hyphens that would otherwise appear in words on the %mor line to an en-dash (Unicode 0x2013) on the fly. This change is done for both the stem words and the words in English translation between the = signs.

8.6.3 Capitalization and Acronyms

The MOR program depends on capitalization of the first letter to identify a word as a proper noun. Earlier versions of CHAT only allowed capital letters at the beginnings of words. This meant that transcribers had to write acronyms such as FBI as F_B_I. However, that restriction has now been lifted and writing FBI is now correct. Other examples include MIT, CMU, USA, MTV, ET, and IU. More complicated acronyms may require underscores, as in C_three_PO and R_two_D_two. The recommended way of transcribing the common name for television is just *tv*. This form is not capitalized, since it is not a proper noun. Similarly, we can write *cd, vcr, tv*, and *dvd*. The underscore is the best mark for combinations that are not true compounds such as *m_and_m-s* for the M&M candy.

Acronyms that are not actually spelled out when produced in conversation should be written as words. Thus *UNESCO* would be written as *Unesco*. The capitalization of the first letter is used to indicate the fact that it is a proper noun. There must be no periods inside acronyms and titles, because these can be confused with utterance delimiters.

8.6.4 Numbers and Titles

Numbers should be written out in words. For example, the number 256 could be written as “two hundred and fifty six,” “two hundred fifty six,” “two five six,” or “two fifty six,” depending on how it was pronounced. It is best to use the form “fifty six” rather than “fifty-six,” because the hyphen is used in CHAT to indicate morphemicization. Other strings with numbers are monetary amounts, percentages, times, fractions, logarithms, and so on. All should be written out in words, as in “eight thousand two hundred and twenty dollars” for $8220, “twenty nine point five percent” for 29.5%, “seven fifteen” for 7:15, “ten o'clock a@l m@l” for 10:00 AM, and “four and three fifths.”

Titles such as *Dr.* or *Mr.* should be written out in their full capitalized form as *Doctor* or *Mister*, as in “Doctor Spock” and “Mister Rogers.” For “Mrs.” use the form “Missus.”

8.6.5 Kinship Forms

The following table lists some of the most important kinship address forms in standard American English. The forms with asterisks cannot be found in *Webster's Third New International Dictionary*.

Kinship Forms

Child	Formal	Child	Formal

Da(da)	Father	Mommy	Mother
Daddy	Father	Nan	Grandmother
Gram(s)	Grandmother	Nana	Grandmother
Grammy	Grandmother	*Nonny	Grandmother
Gramp(s)	Grandfather	Pa	Father
*Grampy	Grandfather	Pap	Father
Grandma	Grandmother	Papa	Father
Grandpa	Grandfather	Pappy	Father
Ma	Mother	Pop	Father
Mama	Mother	Poppa	Father
Momma	Mother	*Poppy	Father
Mom	Mother		

8.6.6 Shortenings

One of the biggest problems that the transcriber faces is the tendency of speakers to drop sounds out of words. For example, a speaker may leave the initial "a" off of "about," saying instead " 'bout." In CHAT, this shortened form appears as (a)bout. clan can easily ignore the parentheses and treat the word as "about." Alternatively, there is a CLAN option to allow the commands to treat the word as a spelling variant. Many common words have standard shortened forms. Some of the most frequent are given in the table that follows. The basic notational principle illustrated in that table can be extended to other words as needed. All of these words can be found in *Webster's Third New International Dictionary*.

More extreme types of shortenings include: "(what)s (th)at" which becomes "sat," "y(ou) are" which becomes "yar," and "d(o) you" which becomes "dyou." Representing these forms as shortenings rather than as nonstandard words facilitates standardization and the automatic analysis of transcripts.

Two sets of contractions that cause particular problems for morphological analysis in English are final apostrophe s and apostrophe d, as in John's and you'd. If you transcribe these as John (ha)s and you (woul)d, then the MOR program will work much more efficiently.

Shortenings

Examples of Shortenings			
(a)bout	don('t)	(h)is	(re)frigerator
an(d)	(e)nough	(h)isself	(re)member
(a)n(d)	(e)spress(o)	-in(g)	sec(ond)
(a)fraid	(e)spresso	nothin(g)	s(up)pose
(a)gain	(es)presso	(i)n	(th)e
(a)nother	(ex)cept	(in)stead	(th)em
(a)round	(ex)cuse	Jag(uar)	(th)emselves
ave(nue)	(ex)cused	lib(r)ary	(th)ere
(a)way	(e)xcuse	Mass(achusetts)	(th)ese

(be)cause	(e)xcused	micro(phone)	(th)ey
(be)fore	(h)e	(pa)jamas	(to)gether
(be)hind	(h)er	(o)k	(to)mato
b(e)long	(h)ere	o(v)er	(to)morrow
b(e)longs	(h)erself	(po)tato	(to)night
Cad(illac)	(h)im	prob(ab)ly	(un)til
doc(tor)	(h)imself	(re)corder	wan(t)

The marking of shortened forms such as (a)bout in this way greatly facilitates the later analysis of the transcript, while still preserving readability and phonological accuracy. Learning to make effective use of this form of transcription is an important part of mastering use of CHAT. Underuse of this feature is a common error made by beginning users of CHAT.

8.6.7 Assimilations and Cliticizations

Words such as “gonna” for “going to” and “whynt cha” for “why don't you” involve complex sound changes, often with assimilations between auxiliaries and the infinitive or a pronoun. None of these forms can be found in *Webster's Third New International Dictionary*. However, to facilitate both phonological and grammatical analysis, it is best to transcribe these forms as they are pronounced, which means as cliticizations. In the corpora in TalkBank, we have tried to always use this form. This means that we always have *coulda* instead of *could have* and so on. The exceptions to this are for *gonna* and *gotta*. Although we recommend using these forms instead of *going to* and *got to*, we were not able to use a global replace function for these two forms, because often *going to* is used in patterns such as *going to Chicago* and *got to* can be used in a form like *he got to my house early*. However, when doing new transcription, it is very helpful to use *gonna* and *gotta* when there is a real cliticization.

Cliticizations

Mod~Aux	**Standard**	**Mod~Inf**	**Standard**	**V~Inf**	**Standard**
coulda	could have	gotta	got to	wanna	want to
mighta	might have	hadta	had to	needa	need to
musta	must have	hafta	have to	gonna	going to
shoulda	should have	hasta	has to	sposta	supposed to
woulda	would have	oughta	ought to		
		useta	used to		

In addition to these cliticizations, other common assimilations include forms listed in this table.

Assimilations

Assimilation	**Standard**	**Assimilation**	**Standard**
dunno	don't know	kinda	kind of
dyou	do you	sorta	sort of

gimme	give me	whyntcha	why didn't you
lemme	let me	wassup	what's up
lotsa	lots of	whaddya	what did you

Unlike the mod:aux group, further types of assimilations are nearly limitless. Some of the most common assimilations are listed in the v-clit.cut file in MOR. However, it is not possible to list all possible assimilations or to assign them to particular parts of speech. Moreover, these other assimilations need to be treated as two or more morphemes. To do this, you should use the replacement notation, as in

```
*CHI: lemme [: let me]
```

If you do this, MOR and the other programs will work on the material in the square brackets, rather than the *lemme* form. An even simpler way of representing some of these forms is by noting omitted letters with parentheses as in: “gi(ve) me” for “gimme,” “le(t) me” for “lemme,” or “d(o) you” for “dyou.”

8.6.8 Communicators and Interjections

Communicators such as *uh* and *nope* and interjections, such as *ugh* and *gosh,* are very frequent. In earlier versions of CHAT and MOR, we distinguished interjections from communicators. However, now we treat these all as communicators. Because their phonological shape varies so much, these forms often have an unclear lexical status. The co.cut file in the MOR lexicon for English provides the shapes for these words that will be recognized by the MOR grammar for English. For consistency, these forms should be used even when the actual phonological form diverges from the standardizing convention, as long as the variant is perceived as related to the standard. Rather than creating new forms for variations in vowel length, it is better to use forms such as *a:h* for *aah.* The English MOR program uses a standard set of forms in the co.cut, co-rhymes.cut, co-under.cut, and co-voc.cut files in the /lex folder that you will need to consult.

Filled pauses are treated in a different way. They are preceded by the ampersand-hyphen mark (&-) which allows them to be ignored as words. Specifically these forms are used to mark the various forms of filled pauses: &-ah, &-eh, &-er, &-ew, &-hm, &-mm, &-uh, &-uhm, and &-um.

8.6.9 Spelling Variants

A number of words have frequent spelling variants. These include *altho* for *although*, *donut* for *doughnut*, *tho* for *though*, *thru* for *through*, and *abc's* for *abcs*. Transcribers should use the spellings for these words used by the files in the English MOR grammar. In general, it is best to avoid the use of monomorphemic words with apostrophes. For example, it is better to use the form *mam* than the form *ma'am*. However, apostrophes must be used in English for multimorphemic contractions such as *I'm* or *don't*.

8.6.10 Colloquial Forms

Colloquial and slang forms are often listed in the dictionary. Examples include *telly* for television and *rad* for *radical*. The following table lists some such colloquial forms

with their corresponding standard forms. Words that are marked with an asterisk cannot be found in *Webster's Third New International Dictionary.*

Colloquial Forms

Form	Meaning	Form	Meaning
Doggone	problematic	okeydokey	all right
fuddy+duddy	old-fashioned person	telly	television
grabby	grasping (adj)	thingumabob	thing
hon	honey(name)	thingumajig	thing
humongous	huge	tinker+toy	toy
Looka	look	who(se)jigger	thing
Lookit	look!	whatchamacallit	thing

8.6.11 Dialectal Variations

Other variant pronunciations, such as *dat* for *that*, involve standard dialectal sound substitutions without deletions. Unfortunately, using these forms can make lexical retrieval very difficult. For example, a researcher interested in the word *together* will seldom remember to include *tagether* in the search string. One solution to this problem is to follow each variant form with the standard form, as given below using the [: replacement] notation. Another solution is to create a full phonological transcription of the whole interaction linked to a full sonic CHAT digitized audio record. In transcripts where the speakers have strong dialectal influences, this is probably the best solution. A third solution is to ignore the dialectal variation and simply transcribe the standard form. If this is being done, the practice must be clearly noted in the readme file. None of these forms are in *Webster's Third New International Dictionary*.

Dialectal Variants

Variant	Standard	Variant	Standard
caint	can't	hows about	how about
da	the	nutin	nothing
dan	than	sumpin	something
dat	that	ta	to
de	the	tagether	together
dese	these	tamorrow	tomorrow
deir	their	weunz	we
deirselves	themselves	whad	what
dem	them	wif	with
demselves	themselves	ya	you
den	then	yall	you all
dere	there	yer	your
dey	they	youse	you all
dis	this	yinz	you all

dose	those	younz	you all
fer	for	ze	the
git	get	zis	this
gon	going	zat	that
hisself	himself		

8.6.12 Baby Talk

Baby talk or "caregiverese" forms include onomatopoeic words, such as *choochoo*, and diminutives, such as *froggie* or *thingie*. In the following table, diminutives are given in final "-ie" except for the six common forms *doggy, kitty, piggy, potty, tummy*, and *dolly*. Wherever possible, use the suffix "-ie" for the diminutive and the suffix "-y" for the adjectivalizer. The following table does not include the hundreds of possible diminutives with the "-ie" suffix simply attached to the stem, as in *eggie, footie, horsie*, and so on. Nor does it attempt to list forms such as *poopy*, which use the adjectivalizer "-y" attached directly to the stem. Words that are marked with an asterisk cannot be found in *Webster's Third New International Dictionary*.

Baby Talk

Baby Talk	Standard	Baby Talk	Standard
beddie(bye)	go to sleep	nunu	hurt
blankie	blanket	night(ie)+night	good night
booboo	injury, hurt	owie	hurt
boom	fall	pantie	underpants
byebye	good-bye	pee	urine, urinate
choochoo	train	peekaboo	looking game
cootchykoo	tickle	peepee	urine, urinate
dark+time	night, evening	peeyou	smelly
doggy	dog	poo(p)	defecation, defecate
dolly	doll	poopoo	defecation, defecate
doodoo	feces	potty	toilet
dumdum	stupid	rockabye	sleep
ew	unpleasant	scrunch	crunch
footie+ballie	football	smoosh	smash
gidd(y)up	get moving	(t)eensy(w)eensy	little
goody	delight	(t)eeny(w)eeny	little
guck	unpleasant	teetee	urine, urinate
jammie	pajamas	titty	breast
kiki	cat	tippytoe	on tips of toes
kitty	cat	tummy	stomach, belly
lookee	look yee!	ugh	unpleasant
moo+cow	cow	(wh)oopsadaisy	surprise or mistake

8.6.13 Word separation in Japanese

Many analyses with CLAN rely on words as items. However, in Japanese script (Kana, Kanji), words are traditionally not divided by spaces. When transcribing Japanese data in Latin script (Romaji) as well as in Japanese script (Kana Kanji), you should add spaces to identify words. The WAKACHI02 system can be downloaded as a part of the complete JPN grammar from https://talkbank.org/morgrams.This web page summarizes the rules for word separation (Wakachigaki). It is crucial to follow these rules in order to get correct results from MOR (automatical morphological analysis) or DSS (Developmental Sentence Score).

8.6.14 Abbreviations in Dutch

Dutch makes extensive use of abbreviations in which vowels are often omitted leaving single consonants, which are merged with nearby words. For consistency of morphological analysis, it is best to transcribe these shortenings using the parenthesis notation, as follows:

Abbreviations in Dutch

Abbreviation	CHAT form	Abbreviation	CHAT form
'k	(i)k	nie	nie(t)
'm	(he)m	es	e(en)s
'r	(e)r	'n	(ee)n
z'n	z(ij)n	's	(i)s
'b	(he)b	't	(he)t
'ns	(ee)ns	wa	wa(t)
'rin	(e)rin	da	da(t)
'raf	(e)raf	'weest	(ge)weest
'ruit	(e)ruit		
'rop	(e)rop		

Some forms that should probably remain with their standard apostrophes include 'smorgens, 'sochtends, 'savonds, 'snachts, and the apostrophe-s plural form.

9 Utterances

The basic units of CHAT transcription are the morpheme, the word, and the utterance. In the previous two chapters we examined principles for transcribing words and morphemes. In this chapter we principles for delimiting utterances.

9.1 One Utterance or Many?

Early child language is rich with repetitions. For example, a child may often say the same word or group of words eight times in a row without changes. The CHAT system provides mechanisms for coding these repetitions into single utterances. However, at the earliest stages, it may be misleading to try to compact these multiple attempts into a single line. Consider five alternative ways of transcribing a series of repeated words.

1. Simple transcription of the words as several items in a single utterance:

```
*CHI:     milk milk milk milk.
```

2. Transcription of the words as items in a single utterance, separated by commas:

```
*CHI:     milk, milk, milk, milk.
```

3. Transcription as four repetitions of a single word.

```
*CHI:     milk [x 4].
```

4. Treatment of the words as a series of attempts to repeat the single word:

```
*CHI:     milk [/] milk [/] milk [/] milk.
```

5. Treatment of the words as separate utterances:

```
*CHI:     milk.
*CHI:     milk.
*CHI:     milk.
*CHI:     milk.
```

These five forms of transcription will lead to markedly different analytic outcomes for programs such as MLU (mean length of utterance). The first two forms will all be counted as having one utterance with four morphemes for an MLU of 4.0. The third and fourth forms will be counted as having one utterance with one morpheme for an MLU of 1.0. The fifth form will be counted as having four utterances each with one morpheme for an MLU of 1.0.

Of course, not all analyses depend crucially on the computation of MLU, but problems with deciding how to compute MLU point to deeper issues in transcription and analysis. In order to compute MLU, one has to decide what is a word and what is an utterance and these are two of the biggest decisions that one has to make when transcribing and analyzing child language. In this sense, the computation of MLU serves as a methodological trip wire for the consideration of these two deeper issues. Other analyses, including lexical, syntactic, and discourse analyses also require that these decisions be made clearly and consistently. However, because of its conceptual simplicity, the MLU index places these problems into the sharpest focus.

The first two forms of transcription all make the basic assumption that there is a single utterance with four morphemes. Given the absence of any clear syntactic relation between the four words, it seems difficult to defend use of this form of transcription.

The third and fourth forms of transcription treat the successive productions of the word "milk" as repeated attempts to produce a single word. This form of transcription makes sense if the child was simply perseverating. If the third form of transcription is used, the commands will, by default, treat the utterance as having only one morpheme. For the fourth form of transcription, CLAN provides two possibilities. The default is to treat the fourth type as a variant of the third form. However, there is also a CLAN option that allows the user to override this default and treat each word as a separate morpheme. This then allows the researcher to compute two different MLU values. The analysis with repetitions excluded could be viewed as the one that emphasizes syntactic structure and the one with repetitions included could be viewed as the one that emphasizes productivity.

Finally, if there is evidence that the word is not simply a repetition, it would seem best to use the fifth form of transcription. This is particularly true if the intonation pattern indicates repeated insistence on a basic single-word message.

The example we have been discussing involves a simple case of word repetition. In other cases, researchers may want to group together non-repeated words for which there is only partial evidence of syntactic or semantic combination. Consider the contrast between these next two examples. In the first example, the presence of the conjunction "and" motivates treatment of the words as a syntactic combination:

```
*CHI:     red, yellow, blue, and white.
```

However, without the conjunction or other intonational evidence, the words are best treated as separate utterances:

```
*CHI:     red.
*CHI:     yellow.
*CHI:     blue.
*CHI:     white.
```

As the child gets older, the solidification of intonational patterns and syntactic structures will give the transcriber more reason to group words together into utterances and to code retracings and repetitions as parts of larger utterances.

9.2 Satellite Markers

Segmentation into utterances can be facilitated through careful treatment of interactional markers and other "communicators" such as "yes," "sure," "well," and "now." These markers should be grouped together with the utterances to which they are most closely bound in terms of intonation. This grouping can be marked with commas, or more explicitly through the use of prefixed (F2+v to enter ‡) and suffixed (F2+t to enter „) interactional markers, as in these examples.

```
*CHI:     no ‡ Mommy no go.
*CHI:     Mommy ‡ I want some.
*CHI:     you need it „ right?
*CHI:     you need it „ don't you?
```

The types of elements that occur as initial satellites include vocatives and communicators (*well*, *but, sure, gosh*). Elements that occur as final satellites include question markers (*okay*?, *see*?) and sentence final particles, as well as vocatives and communicators. The use of these prefixing and suffixing interactional markers is particularly important for Asian languages that use sentence final particles. These satellite markers should be surrounded by spaces, since they will be treated as separate word forms by MOR and GRASP. Use of these markers helps improve syntactic analysis, and provides a more realistic characterization of utterances.

9.3 Discourse Repetition

Earlier, we discussed problems involved in deciding whether a group of words should be viewed as one utterance or as several. This issue moves into the background when the word repetitions are broken up by the conversational interactions or by the child's own actions. Consider this example:

```
*MOT:     what do you drink for breakfast?
*CHI:     milk.
*MOT:     and what do you drink for lunch?
*CHI:     milk.
*MOT:     how about for dinner?
*CHI:     milk.
*MOT:     and what is your favorite thing to drink at bedtime?
*CHI:     milk.
```

Or the child may use a single utterance repeatedly, but each time with a slightly different purpose. For example, when putting together a puzzle, the child may pick up a piece and ask:

```
*CHI:     where does this piece go?
```

This may happen nine times in succession. In both of these examples, it seems unfair from a discourse point of view to treat each utterance as a mere repetition. Instead, each is functioning independently as a full communication. One may want to mark the fact that the lexical material is repeated, but this should not affect other quantitative measures.

9.4 C-Units, sentences, utterances, and run-ons

There is a tendency in the literature to avoid the use of the term “sentence” to refer to the units of spoken language. To avoid this problem, researchers use the terms “utterance” and “c-unit” or conversational unit. The latter is defined as a main clause along with its dependent (subordinate or coordinate) clauses. However, when defined in this way, a c-unit is really not too different from a sentence. The major difference is that a c-unit may be incomplete and may include disfluencies, retraces, etc. which would not be present in written language.

In the past, some transcribers have tended to group all of the words in a turn into a single sentence with only one final delimiter. This is a mistake. Utterances can include main clauses with associated depending clauses, but they should not include multiple main clauses. Sometimes children will string together multiple utterances with “and … and”. In such cases, each utterance with a new “and” should be placed on a new tier, as a

new utterance. However, clauses that are joined by other conjunctions should be treated as a single utterance.

9.5 Retracing

When a speaker abandons an utterance, sometimes another speaker will take a turn. In that case, the first utterance can be marked with a trailing off terminator, as discussed below. However, if the first speaker continues, then the transcriber has a choice to make. One possibility is to mark the abandoned and retraced material with the [//] symbol along with some scoping markers. However, if the abandonment of the first segment is followed by a significant pause, then it would be better to consider it as a trailing off and then to begin a new utterance with the following material. In either case, it is a good idea to mark the fact that there was a long pause by inserting a pause marker with a time value, such as (3.5) for 3.5 seconds.

9.6 Basic Utterance Terminators

The basic CHAT utterance terminators are the period, the question mark, and the exclamation mark. CHAT requires that there be only one utterance on each main line. In order to mark this, each utterance must end with one of these three utterance terminators. It is possible to use the comma on the main line, but it is not treated as a terminator. However, a single main line utterance may extend for several computer lines, as in this example:

```
*CHI:     this.
*MOT:     if this is the one you want, you will have to take your
    spoon out of the other one.
```

The utterance in this main tier extends for two lines in the computer file. When it is necessary to continue an utterance on the main tier onto a second line, the second line *must begin with a tab.* CLAN is set to expect no more than 2000 characters in each main tier, dependent tier, or header line.

Period **.**

A period marks the end of an unmarked (declarative) utterance. Here are some examples of unmarked utterances:

```
*SAR:     I got cold.
*SAR:     pickle.
*SAR:     no.
```

For correct functioning of CLAN, periods should be eliminated from abbreviations. Thus “Mrs.” should be written as *Mrs* and *E.T.* should become *E+T*. Only proper nouns and the word “I” and its contractions are capitalized. Words that begin sentences are not capitalized.

Question Mark **?**

The question mark indicates the end of a question. A question is an utterance that uses a wh-question word, subject- verb inversion, or a tag question ending. Here is an example of a question:

```
*FAT: is that a carrot ?
```

The question mark can also be used after a declarative sentence when it is spoken with the rising intonation of a question.

Exclamation Point !

An exclamation point marks the end of an imperative or emphatic utterance. Here is an example of an exclamation:

```
*MOT:     sit down!
```

If this utterance were to be conveyed with final rising contour, it would instead be:

```
*MOT:     sit down ?
```

9.7 Separators

CHAT allows for the use of several conventional punctuation features that have no formal role in the transcription system. We call these "separators" and distinguish them from terminators, which have a formal role, and the various CA intonation marks.

Comma ,

The comma is used widely throughout CHAT transcripts to represent a combination of features such as pause, syntactic juncture, intonational drop, and others. Although it has no formal definition or systematic characterization, it is fine to use this symbol. The use of comma to mark level intonation in CA is replaced by the use of the mark →.

Semicolon ;

The semicolon is used primarily to mark syntactic structures in corpora such as the SCOTUS oral arguments from the Supreme Court. Most conversational transcripts do not need to use this mark. The use of semicolon to mark a light final drop in CA is replaced by the use of the mark ↘.

Colon :

In order to use the colon as a separator, it must be surrounded by spaces. The colon is also used within words to mark lengthening.

Other

Transcribers should avoid using other separators, because most of them have special meanings in CHAT.

9.8 Tone Direction

Earlier versions of CHAT had used a special set of terminating tone units, such as -? and -! . In order to bring CHAT more into accord with standard practice, we have shifted to a reliance on marks such as ↑ for rising ↓. All the other CA marks can also be used in CHAT files. However, unlike CA, CHAT requires that every utterance have a final delimiter. This means that CA and CHAT are in agreement in assuming that final question mark includes a rising intonation, final exclamation mark represents emphatic intonation, and that final period represents a final fall. In addition, CHAT assumes that the question mark is used with questions, that the exclamation mark is used with exclamations, and that the period terminates declarative sentences. Sometimes questions do not end in a rising intonation. In that case, the actual intonation used can be marked with the falling mark ↓ after the final word, then followed by the question mark, as in this example:

```
*MOT:     Are you going to store↓ ?
```

Final rise fall contour can be represented with ↑↓ and final fall-rise can be represented with ↓↑ .

9.9 Prosody Within Words

CHAT also provides codes for marking lengthening, and pausing within words. For marking features such as stressing and pitch rise and fall, transcribers should rely on the CHAT-CA marks indicated above and provided in the chapter on CA coding. In addition to those symbols, the following symbols are also available:

Primary Stress ˈ

The Unicode symbol (U02C8) can be used to mark primary stress. It is placed right before the stressed syllable, as in this example:

```
MOT:      baby want baˈna:nas ?
```

Secondary Stress

The Unicode symbol (U02CC) can be used to mark secondary stress. It is placed right before the stressed syllable, as in this example:

```
MOT:      baby want ˌbaˈna:nas ?
```

Lengthened Syllable :

A colon within a word indicates the lengthening or drawling of a syllable. This mark should be attached to a vowel or continuant, because it is difficult to drawl an obstruent:

```
MOT:      baby want bana:nas?
```

Pause Between Syllables ^

A pause between syllables may be indicated as in this example:

```
MOT:      is that a rhi^noceros ?
```

There is no special CHAT symbol for a filled pause. Instead, &-ah, &-eh, &-er, &-ew, &-hm, &-mm, &-uh, &-uhm, and &-um are used to mark the various forms of filled pauses.

Blocking ^

Speakers with marked language disfluencies often engage in a form of word attack known as "blocking" (Bernstein Ratner et al., 1996). This form of word attack is marked by a caret or up arrow placed directly before the word.

9.10 Local Events

We tend to think of the basic form of a transcript as involving a series of words, along with occasional commentary about these words. We can think of these words as a chain of events in which our convention of writing from left to right represents the temporal sequence of the events. During this sequence of words, we can also distinguish a variety of local events that do not map onto words. There are five types of these local events: simple events, complex events, pauses, long events, and interposed remarks.

9.10.1 Simple Events

In addition to the formalized exclamations given in the chapter on words, speakers produce a wide variety of sounds such as cries, sneezes, and coughs. These are indicated in CHAT with the prefix &=, in order to produce forms such as &=sneezes and &=yells. In order to retrieve these forms consistently, we have set up the following standardized spellings. Note that verbs are given in the third person present form. Other languages can either use this set or create their own translations of these terms. Perhaps the most common of these is &=laughs, which can be used to represent all types of laughs, chuckles, and giggles.

&=belches	&=hisses	&=grunts	&=whines
&=coughs	&=hums	&=roars	&=whistles
&=cries	&=laughs	&=sneezes	&=whimpers
&=gasps	&=moans	&=sighs	&=yawns
&=groans	&=mumbles	&=sings	&=yells
&=growls	&=pants	&=squeals	&=vocalizes

It is important to remember that these codes must fully characterize complete local events. If your intention is to mark that a stretch of words has been mumbled, then you should use the scoped codes discussed in the next chapter. However, if you only wish to code that some mumbling or singing occurs at a particular point, then you can use this simpler form.

Simple event forms can also be used to mark actions such as running and reading. When these actions are transitive, as in imit: (imitation), point: and move: they can also

take an object. For example, a very common vocalizer is &=imit:motor for an imitation of the sound of a motor. The table below illustrates this use of compound simple codes.

&=imit:motor	&=ges:frustration	&=writes:dog	&=points:car
&=imit:plane	&=ges:squeeze	&=reads:sign	&=points:nose
&=imit:lion	&=ges:come	&=walks:door	&=turns:page
&=imit:baby	&=shows:picture	&=runs:door	&=hits:table
&=ges:ignore	&=shows:scab	&=eats:cookie	&=pats:head
&=ges:unsure	&=moves:doll	&=drinks:milk	

The object of the &=imit codes indicates the noise source being imitated vocally. The objects of the &=ges codes indicate the meaning of the gestures being used. The objects of activities such as &=walk and &=run indicate the direction or goal of the walking or running. For actions such as &=slurp and &=eat used by themselves, the code represents the auditory results of the slurping or eating.

Finally, you can compose codes using parts of the body as in &=head:yes to indicate nodding "yes" with the head. Some codes of this type include: &=head:yes, &=head:no, &=head:shake, &=hands:no, &=hands:hello, &=eyes:open, &=mouth:open, and &=mouth:close.

This form of coding is compact and can be easily searched. Moreover, it is easy to locate at a point within an ongoing utterance without breaking up the readability of the utterance. Whenever possible, try to use this form of coding as a substitute for writing longer comments on the comment line or inserting complex local events on the main line.

9.10.2 Interposed Word &*

It is sometimes convenient to mark the interposition or insertion of a short comment word in a back channel, such as "yeah" or "mhm", within a longer discourse from the speaker who has the floor without breaking up the utterance of the main speaker. This is marked using &* followed by the speaker's 3-letter ID, a colon, and then the interposed word. Here is an example of how this can be used:

```
CHI:  when I was over at my friend's house &*MOT:mhm the dog tried to
    lick me all over.
```

9.10.3 Complex Local Events

In addition to the restricted set of simple events discussed above, it is possible to use an open form to simply insert any sort of description of an event on the main line.

Complex Local Event **[^ text]**

Like the simple local events, these complex local events are assumed to occur exactly at the position marked in the text and not to extend over some other events. If the material is intended as a comment over a longer scope of events, use the form of the scoped comments given in the next section. This form of coding can also be used at the very beginning of utterances to replace the earlier "precodes" that marked things like the specific addressee, events just before the utterance, or the background to the utterance.

9.10.4 Pauses

The third type of local event is the unfilled pause, which takes up a specified duration of time at the point marked by the code. Pauses that are marked only by silence are coded on the main line with the symbol (.). Longer pauses between words can be represented as (..) and a very long pause as (…) This example illustrates these forms:

```
*SAR:     I don't (..) know .
*SAR:     (...) what do you (...) think ?
```

If you want to be exact, you can code the exact length of the pauses in seconds, as in these examples.

```
*SAR:     I don't (0.15) know .
*SAR:     (13.4) what do you (2.) think ?
```

If you need to add minutes, then you can use a colon for one minute, and 5.15 seconds:

```
*SAR:     I don't (1:05.15) know .
```

9.10.5 Long Events

It is possible to mark the beginning and ending of some extralinguistic event with the long feature convention. For this marking, there is a beginning code at the beginning of the event and a termination code for the ending.

Long Vocal Event **&{l=* intervening text &}l=***

Here the asterisk marks some description of the long event. For example, a speaker could begin laughing at the point marked by &{l=laughs and then continue until the end marked by &}l=laughs.

Long Nonvocal Event **&{n=* intervening text &}n=***

Here the asterisk marks some description of a long nonverbal event. For example, a speaker could begin waving their hands at the point marked by &{n=waving:hands and then continue until the end marked by &}n=waving:hands.

9.11 Special Utterance Terminators

In addition to the three basic utterance terminators, CHAT provides a series of more complex utterance terminators to mark various special functions. These special terminators all begin with the + symbol and end with one of the three basic utterance terminators.

Trailing Off **+…**

The trailing off or incompletion marker (plus sign followed by three periods) is the terminator for an incomplete, but not interrupted, utterance. Trailing off occurs when speakers shift attention away from what they are saying, sometimes even forgetting what

they were going to say. Usually the trailing off is followed by a pause in the conversation. After this lull, the speaker may continue with another utterance or a new speaker may produce the next utterance. Here is an example of an uncompleted utterance:

```
*SAR:     smells good enough for +...
*SAR:     what is that?
```

If the speaker does not really get a chance to trail off before being interrupted by another speaker, then use the interruption marker +/. rather than the incompletion symbol. Do not use the incompletion marker to indicate either simple pausing (.), repetition [/], or retracing [//]. Note that utterance fragments coded with +… will be counted as complete utterances for analyses such as MLU, MLT, and CHAINS. If your intention is to avoid treating these fragments as complete utterances, then you should use the symbol [/-] discussed later.

Trailing Off of a Question **+..?**

If the utterance that is being trailed off has the shape of a question, then this symbol should be used.

Question With Exclamation **+!?**

When a question is produced with great amazement or puzzlement, it can be coded using this symbol. The utterance is understood to constitute a question syntactically and pragmatically, but an exclamation intonationally.

Interruption **+/.**

This symbol is used for an utterance that is incomplete because one speaker is interrupted by another speaker. Here is an example of an interruption:

```
*MOT:     what did you +/.
*SAR:     Mommy.
*MOT:     +, with your spoon.
```

Some researchers may wish to distinguish between an invited interruption and an uninvited interruption. An invited interruption may occur when one speaker is prompting his addressee to complete the utterance. This should be marked by the ++ symbol for other-completion, which is given later. Uninvited interruptions should be coded with the symbol +/. at the end of the utterance. An advantage of using +/. instead of +... is that programs like MLU are able to piece together the two segments and treat it as a single utterance when a segment with +/. is followed by +, on the next utterance.

Interruption of a Question **+/?**

If the utterance that is being interrupted has the shape of a question, then this symbol should be used.

Self-Interruption **+//.**

Some researchers wish to be able to distinguish between incompletions involving a trailing off and incompletions involving an actual self-interruption. When an incompletion is not followed by further material from the same speaker, the +... symbol should always be selected. However, when the speaker breaks off an utterance and starts up another, the +//. symbol can be used, as in this example:

```
*SAR:     smells good enough for +//.
*SAR:     what is that?
```

There is no hard and fast way of distinguishing cases of trailing off from self-interruption. For this reason, some researchers prefer to avoid making the distinction. Researchers who wish to avoid making the distinction should use only the +... symbol.

Self-Interrupted Question **+//?**

If the utterance being self-interrupted is a question, you can use the +//? symbol.

Transcription Break **+.**

It is often convenient to break utterances at phrasal boundaries in order to mark overlaps. When this is done, the first segment is ended with the +. terminator, as in this example:

```
*SAR:     smells good enough for me +.
*MOT:     but +.
*SAR:     if I could have some.
*MOT:     why would you want it?
```

Quotation **“ and ”**

For marking short quotation stretches inside an utterance, the begin double-quote (“, Unicode 201C) and end double-quote (”, Unicode 201D) symbols can be used. These can be entered in the CLAN editor using F2-' and F2-" respectively.

Quotation Follows **+"/.**

During story reading and similar activities, a great deal of talk may involve direct quotation. In order to mark off this material as quoted, a special symbol can be used, as in the following example:

```
*CHI:     and then the little bear said +"/.
*CHI:     +" please give me all of your honey.
*CHI:     +" if you do, I'll carry you on my back.
```

The use of the +"/. symbol is linked to the use of the +" symbol. Breaking up quoted material in this way allows us to maintain the rule that each separate utterance should be on a separate line. This form of notation is only used when the material being quoted is a complete clause or sentence. It is not needed when a few words are being quoted in noncomplement position. In those cases, use the standard single and double quotation marks described just above.

Quotation Precedes **+".**

This symbol is used when the material being directly quoted precedes the main clause, as in the following example:

```
*CHI:     +" please give me all of your honey.
*CHI:     the little bear said +".
```

9.12 Utterance Linkers

There is another set of symbols that can be used to mark other aspects of the ways in which utterances link together into turns and discourse. These symbols are not utterance terminators, but utterance initiators, or rather "linkers." They indicate various ways in which an utterance fits in with an earlier utterance. Each of these symbols begins with the + sign.

Quoted Utterance **+"**

This symbol is used in conjunction with the +"/. and +". symbols discussed earlier. It is placed at the beginning of an utterance that is being directly quoted.

Quick Uptake **+^**

Sometimes an utterance of one speaker follows quickly on the heels of the last utterance of the preceding speaker without the customary short pause between utterances. An example of this is:

```
*MOT:     why did you go?
*SAR:     +^ I really didn't.
```

Self Completion **+,**

The symbol +, can be used at the beginning of a main tier line to mark the completion of an utterance after an interruption. In the following example, it marks the completion of an utterance by CHI after interruption by EXP. Note that the incompleted utterance must be terminated with the incompletion marker.

```
*CHI:     so after the tower +/.
*EXP:     yeah.
*CHI:     +, I go straight ahead.
```

Other Completion **++**

A variant form of the +, symbol is the ++ symbol which marks “latching” or the completion of another speaker's utterance, as in the following example:

```
*HEL:     if Bill had known +...
*WIN:     ++ he would have come.
```

10 Scoped Symbols

Up to this point, the symbols we have discussed are inserted at single points in the transcript. They refer to events occurring at particular points during the dialogue. There is another major class of symbols that refers not to particular points in the transcript, but to stretches of speech. These marker symbols are enclosed in square brackets and the material to which they relate can be enclosed in angle brackets. The material in the square brackets functions as a descriptor of the material in angle brackets. If a scoped symbol applies only to the single word preceding it, the angle brackets need not be marked, because CLAN considers that the material in square brackets refers to a single preceding word when there are no angle brackets. There should be no other material entered between the square brackets and the material to which it refers. Depending on the nature of the material in the square brackets, the material in the angle brackets may be automatically excluded from certain types of analysis, such as MLU counts and so forth. Scoped symbols are useful for marking a wide variety of relations, including paralinguistics, explanations, and retracings.

10.1 Audio and Video Time Marks

In order to link segments of the transcript to stretches of digitized audio and video, CHAT uses the following notation:

Time Alignment **·0_1073·**

This marker provides the begin and end time in milliseconds for a segment in a digitized video file or audio file. Usually, this information is hidden. However, if you use the escape-A command in the editor, the bullet will expand and you will see the time values. Each set of time alignment information has an implicit scope that includes all of the material to the left up to the next set of bullets. These time marks allow for single utterance playback or continuous playback. If you insert a dash before the time, as in

·-5567_9888·

this indicates that continuous playback should not actually wait through long periods of silence between the bullets. By default, these bullets should occur at the end of speaker lines, after the final terminator and after any postcodes. However, if the option "multiple" is selected in the @Options field, then bullets may also occur within utterances.

Pic Bullet **·%pic: cat.jpg·**

This marker is used to insert a bullet that can be clicked to display a picture. This field is also used in the gesture coding system discussed in the CLAN manual. The format of these files is not fixed by CHAT, but many of the same conventions are used. One additional code used there is the @T: header which marks the place of the insertion of a video picture taken from a movie as a thumbnail representation of what is happening at a particular moment in the interaction.

Text Bullet **·%txt: cat.txt·**

This marker is used to insert a bullet that can be clicked to display a text file.

10.2 Paralinguistic and Duration Scoping

Paralinguistic Material **[=! text]**

Paralinguistic events, such as "coughing," "laughing," or "yelling" can be marked by using square brackets, the =! symbol, a space, and then text describing the event.

```
*CHI:     that's mine [=! cries].
```

This means that the child cries while saying the word "mine." If the child cries throughout, the transcription would be:

```
*CHI:     <that's mine> [=! cries].
```

In order to indicate crying with no particular vocalization, you should use the &=cries "simple form" notation discussed earlier, as in

```
*CHI:     &=cries .
```

This same format of [=! text] can also be used to describe prosodic characteristics such as "glissando" or "shouting" that are best characterized with full English words. Paralinguistic effects such as soft speech, yelling, singing, laughing, crying, whispering, whimpering, and whining can also be noted in this way. For a full set of these terms and details on their usage, see Crystal (1969) or Trager (1958). Here is another example:

```
*NAO:     watch out [=! laughing].
```

Stressing **[!]**

This symbol can be used without accompanying angle brackets to indicate that the preceding word is stressed. The angle brackets can also mark the stressing of a string of words, as in this example:

```
*MOT:     Billy, would you please <take your shoes off> [!].
```

Contrastive Stressing **[!!]**

This symbol can be used without accompanying angle brackets to indicate that the preceding word is contrastively stressed. If a whole string of words is contrastively stressed, they should be enclosed in angle brackets.

Duration **[# time]**

This symbol indicates the duration in seconds of the preceding material that has been marked with angle brackets as in:

```
*MOT:     I could use <all of them> [# 2.2] for the party.
```

10.3 Explanations and Alternatives

Explanation **[= text]**

This symbol is used for brief explanations on the text tier. This symbol is helpful for specifying the deictic identity of objects and people.

```
*MOT:    don't look in there [= closet]!
```

Explanations can be more elaborate as in this example:

```
*ROS:    you don't scare me anymore [= the command "don't scare me
   anymore!"].
```

An alternative form for transcribing this is:

```
*ROS:    you don't scare me any more.
%exp:    means to issue the imperative "Don't scare me anymore!"
```

Replacement **[: text]**

Earlier we discussed the use of a variety of nonstandard forms such as "gonna" and "hafta.". In order for MOR to morphemicize such words, the transcriber can use a replacement symbol that allows clan to substitute a target language form for the form actually produced. Here is an example:

```
*BEA:    when ya gonna [: going to] stop doin(g) that?
*CHA:    whyncha [: why don't you] just be quiet!
```

In this example, "gonna" is followed by its standard form in brackets. The colon that follows the first bracket tells CLAN that the material in brackets should replace the preceding word. The replacing string can include any number of words, but the thing being replaced can only be a single word, not a series of words. There must be a space following the colon, in order to keep this symbol separate from other symbols that use letters after the colon. This example also illustrates two other ways in which CHAT and clan deal with nonstandard forms. The lexical item "ya" is treated as a lexical item distinct from "you." However, the semantic equivalence between "ya" and "you" is maintained by the formalization of a list of dialectal spelling variations. The string "doin(g)" is treated by CLAN as if it were "doing." This is done by simply having the programs ignore the parentheses, unless they are given instructions to pay attention to them, as discussed in in the CLAN manual. From the viewpoint of CLAN, a form like "doin(g)" is just another incomplete form, such as "bro(ther)."

In order for replacement to function properly, nothing should be placed between the replacing string and the string to be replaced. For example, to mark replacement and error using the [*] code, one should use the form:

```
goed [: went] [*]
```

rather than:

```
goed [*] [: went]
```

Replacement of Real Word [:: text]

When the error involves the incorrect use of a real word, the double colon form of the replacement string may be used, as in:

```
piece [:: peach] [*]
```

For further details on this usage, please see the chapter on Error Coding.

Alternative Transcription [=? text]

Sometimes it is difficult to choose between two possible transcriptions for a word or group of words. In that case an alternative transcription can be indicated in this way:

```
*CHI:     we want <one or two> [=? one too].
```

Comment on Main Line [% text]

Instead of placing comment material on a separate %com line, it is possible to place comments or any type of code directly on the main line using the % symbol in brackets. Here is an example of this usage:

```
*CHI:     I really wish you wouldn't [% said with strong raising of
    eyebrows] do that.
```

You should be careful with using comments on the main line. Overuse of this particular notational form can make a transcript difficult to read and analyze. Because placing a comment directly onto the main line tends to highlight it, this form should be used only for material that is crucial to the understanding of the main line.

Best Guess [?]

Often audiotapes are hard to hear because of interference from room noise, recorder malfunction, vocal qualities, and so forth. Nonetheless, transcribers may think that, through the noise, they can recognize what is being said. There is some residual uncertainty about this "best guess." This symbol marks this in relation to the single preceding word or the previous group of words enclosed in angle brackets.

```
*SAR:     I want a frog [?]
```

In this example, the word that is unclear is "frog." In general, when there is a symbol in square brackets that takes scoping and there are no preceding angle brackets, then the single preceding word is the scope. When more than one word is unclear, you can surround the unclear portion in angle brackets as in the following example:

```
*SAR:     <going away with my mommy> [?] ?
```

10.4 Retracing, Overlap, and Clauses

Overlap Follows [>]

During the course of a conversation, speakers often talk at the same time. Transcribing these interactions can be trying. This and the following two symbols are designed to help sort out this difficult transcription task. The "overlap follows" symbol indicates that the text enclosed in angle brackets is being said at the same time as the following speaker's bracketed speech. They are talking at the same time. This code must be used in combination with the "overlap precedes" symbol, as in this example:

```
*MOT:     no (.) Sarah (.) you have to <stop doing that> [>] !
*SAR:     <Mommy I don't like this> [<].
*SAR:     it is nasty.
```

Using these overlap indicators does not preclude making a visual indication of overlap in the following way:

```
*MOT:     no (.) Sarah (.) you have to <stop doing that> [>] !
*SAR:                      <Mommy I don't like this> [<].
*SAR:     it is nasty.
```

CLAN ignores the series of spaces, treating them as if they were a single space.

Overlap Precedes [<]

The "overlap precedes" symbol indicates that the text enclosed in angle brackets is being said at the same time as the preceding speaker's bracketed speech. This code must be used in combination with the "overlap follows" symbol. Sometimes several overlaps occur in a single sentence. It is then necessary to use numbers to identify these overlaps, as in this example:

```
*SAR:     and the <doggy was> [>1] really cute and
    it <had to go> [>2] into bed.
*MOT:     <why don't you> [<1] ?
*MOT:     <maybe we could> [<2].
```

If this sort of intense overlapping continues, it may be necessary to continue to increment the numbers as long as needed to keep everything straight. However, once one whole turn passes with no overlaps, the number counters can be reinitialized to "1."

Lazy Overlap +<

If you don't want to mark the exact beginning and end of overlaps between speakers and only want to indicate the fact that two turns overlap, you can use this code at the beginning of the utterance that overlaps a previous utterance, as in this example:

```
*CHI:     we were taking them home.
*MOT:     +< they had to go in here.
```

This marking simply indicate that the mother's utterance overlaps the previous child utterance. It does not indicate how much of the two utterances overlap. If you need to combine this mark with other utterance linker marks, place this one first, followed by a space. It would not make sense to combine this mark with the +^ mark.

Repetition [/]

Often speakers repeat words or even whole phrases (Goldman-Eisler, 1968; MacWhinney & Osser, 1977). The [/] symbol is used in those cases when a speaker begins to say something, stops and then repeats the earlier material without change. The material being retraced is enclosed in angle brackets. If there are no angle brackets, CLAN assumes that only the preceding word is being repeated. In a retracing without correction, it is necessarily the case that the material in angle brackets is the same as the material immediately following the [/] symbol. Here is an example of this:

```
*BET:     <I wanted> [/] I wanted to invite Margie.
```

If there are pauses and fillers between the initial material and the retracing, they should be placed after the repetition symbol, as in:

```
*HAR:     it's [/] (.) &-um (.) it's [/] it's (.) a &-um (.) dog.
```

When a word or group of words is repeated several times with no fillers, all of the repetitions except for the last are placed into a single group, as in this example:

```
*HAR:     <it's it's it's> [/] it's (.) a &-um (.) dog.
```

By default, all of the clan commands except mlu, mlt, and modrep include repeated material. This default can be changed by using the +r6 switch.

Multiple Repetition [x N]

An alternative way of indicating several repetitions of a single word uses this form:

```
*HAR:     it's [x 4] (.) a &-um (.) dog.
```

This form indicates the fact that a word has been repeated four times. If this form is used, it is not possible to get a count of the repetitions to be added to MLU. However, because this is not usually desirable anyway, there are good reasons to use this more compact form when single words are repeated. For some illustrations of the use of this type of coding for the study of disfluencies such as stuttering, consult Bernstein Ratner, Rooney, and MacWhinney (1996).

Retracing [//]

This symbol is used when a speaker starts to say something, stops, repeats the basic phrase, changes the syntax but maintains the same idea. Usually, the correction moves closer to the standard form, but sometimes it moves away from it. The material being retraced is enclosed in angle brackets. If there are no angle brackets, CLAN assumes that only the preceding word is being retraced. In retracing with correction, it is necessarily true that the material in the angle brackets is different from what follows the retracing symbol. Here is an example of this:

```
*BET:     <I wanted> [//] &-uh I thought I wanted to invite Margie.
```

Retracing with correction can combine with retracing without correction, as in this example:

```
*CHI:     <the fish is> [//] the [/] the fish are swimming.
```

Sometimes retracings can become quite complex and lengthy. This is particularly true in speakers with language disorders. It is important not to underestimate the extent to which retracing goes on in such transcripts. By default, all of the clan commands except mlu, mlt, and modrep include retraced material. This default can be changed by using the +r6 switch.

Reformulation [///]

Sometimes retracings involve full and complete reformulations of the message without any specific corrections. Here is an example of this type:

```
*BET:    <all of my friends had> [///] uh we all decided to go home
   for lunch.
```

When none of the material being corrected is included in the retracing, it is better to use the [///] marker than the [//] marker.

False Start Without Retracing [/-]

In some projects that place special emphasis on counts of particular disfluency types, it may be more convenient to code retracings through a quite different method. For example, the symbols [/] and [//] are used when a false start is followed by a complete repetition or by a partial repetition with correction. If the speaker terminates an incomplete utterance and starts off on a totally new tangent, this can be coded by using the [/-] symbol:

```
*BET:    <I wanted> [/-] uh when is Margie coming?
```

If the material is coded in this way, CLAN will count only one utterance. If the coder wishes to treat the fragment as a separate utterance, the +... and +//. symbols that were discussed on page 67 should be used instead. By default, all of the CLAN programs except MLU, MLT, and MODREP include repeated material. This default can be changed by using the +r6 switch.

Unclear Retracing Type [/?]

This symbol is used primarily when reformatting SALT files to CHAT files, using the SALTIN command. SALT does not distinguish between filled pauses such as “uh”, repetitions ([/]), and retracings ([//]); all three phenomena and possible others are treated as “mazes.” Because of this, SALTIN uses the [/?] symbol to translate SALT mazes into chat hesitation markings.

Excluded Material [e]

Certain types of analysis focus on the speaker’s ability to produce task-relevant material. For example, in a picture description task, it may be useful to exclude material that is not relevant to the actual description of the picture. To do this, the material to be excluded can be marked with [e], as in this example:

```
*BET:     <I think that maybe> [e] the cat is up the tree.
```

Material marked in this way will automatically be excluded by analysis on the %mor line and from the other programs such as DSS, IPSyn, VOCD, GRASP etc that operate on that line.

Clause Delimiter [^c]

If you wish to conduct analyses such as MLU and MLT based on clauses rather than utterances as the basic unit of analysis, you should mark the end of each clause with this symbol. You should not use this code to treat complete sentences as if they were clauses. Instead, each sentence should be transcribe on its own main line. This mark should only be used to demarcate the clauses within complex sentences.

It is not necessary to mark the scope of this symbol, since it is assumed to apply to all the material before it up to the beginning of the utterance or up to the preceding [^c] marker. It is possible to create additional user-defined codes using the format of [^c *], such as [^c err] which could be defined as a marker of a clause that includes an error, or [^c 0s] for a clause with no subject, etc. Then, inside the MLU and MLT programs, you need to add the +c switch to specify exactly which codes of this type should be recognized.

10.5 Error Marking

Errors are marked by placing the [*] symbol after the error. Usually, the [*] marker occurs right after the error. However, if there is a replacement string, such as [: because], that should come first. In repetitions and retracing with errors in the initial part of the retracing, the [*] symbol is placed before the [/] mark. If the error is in the second part of the retracing, the [*] symbol goes after the [/]. In error coding, the form actually produced is placed on the main line and the target form is given on the %err line. The full system for error coding is presented in a separate chapter.

10.6 Initial and Final Codes

The symbols we have discussed so far in this chapter usually refer to words or groups of words. CHAT also allows for codes that refer to entire utterances. These codes are placed into square brackets either at the beginning of the utterance or after the final utterance delimiter. They always begin with a + sign.

Postcodes [+ text]

Postcodes are symbols placed into square brackets at the end of the utterance. They should include the plus sign and a space after the left bracket. There is no predefined set of postcodes. Instead, postcodes can be designed to fit the needs of your particular project. Unlike scoped codes, postcodes must apply to the whole utterance, as in this example:

```
*CHI: not this one. [+ neg] [+ req] [+ inc]
```

Postcodes are helpful in including or excluding utterances from analyses of turn length or utterance length by MLT and MLU. The postcodes, [+ bch] and [+ trn], when combined with the -s and +s+ switch, can be used for this purpose. When the SALTIN command translates codes from SALT format to CHAT format, it treats them as postcodes, because the scope of codes is not usually defined in SALT.

Language Precodes [- text]

Language precodes are used to mark the switch to a different language in multilingual interactions. The text in these codes should come from the three-letter ISO codes used in the @Languages header.

Excluded Utterance [+ bch]

Sometimes we want to have a way of marking utterances that are not really a part of the main interaction, but are in some "back channel." For example, during an interaction that focuses on a child, the mother may make a remark to the investigator. We might want to exclude remarks of this type from analysis by MLT and MLU, as in this interaction:

```
*CHI:     here one.
*MOT:     no, here.
%sit:     the doorbell rings.
*MOT:     just a moment. [+ bch]
*MOT:     I'll get it. [+ bch]
```

In order to exclude the utterances marked with [+ bch], the -s"[+ bch]" switch must be used with mlt and mlu.

Included Utterance [+ trn]

The [+ trn] postcode can force the MLT command to treat an utterance as a turn when it would normally not be treated as a turn. For example, utterances containing only "0" are usually not treated as turns. However, if one believes that the accompanying nonverbal gesture constitutes a turn, one can note this using [+ trn], as in this example:

```
*MOT:     where is it?
*CHI:     0. [+ trn]
%act:     points at wall.
```

Later, when counting utterances with MLT, one can use the +s+"[+ trn]" switch to force counting of actions as turns, as in this command:

```
mlt +s+"[+ trn]" sample.cha
```

11 Dependent Tiers

In the previous chapters, we have examined how CHAT can be used to create file headers and to code the actual words of the interaction on the main line. The third major component of a CHAT transcript is the ancillary information given on the dependent tiers. Dependent tiers are lines typed below the main line that contain codes, comments, events, and descriptions of interest to the researcher. It is important to have this material on separate lines, because the extensive use of complex codes in the main line would make it unreadable. There are many codes that refer to the utterance as a whole. Using a separate line to mark these avoids having to indicate their scope or cluttering up the end of an utterance with codes.

It is important to emphasize that no one expects any researcher to code all tiers for all files. CHAT is designed to provide options for coding, not requirements for coding. These options constitute a common set of coding conventions that will allow the investigator to represent those aspects of the data that are most important. It is often possible to transcribe the main line without making much use at all of dependent tiers. However, for some projects, dependent tiers are crucial.

All dependent tiers should begin with the percent symbol (%) and should be in lower-case letters. As in the main line, dependent tiers consist of a tier code and a tier line. The dependent tier code is the percent symbol, followed by a three-letter code ID and a colon. The dependent tier line is the text entered after the colon that describes fully the elements of interest in the main tier. Except for the %mor and %gra tiers, these lines do not require ending punctuation. Here is an example of a main line with two dependent tiers:

```
*MOT:     well go get it!
%spa:     $IMP $REF $INS
%mor:     ADV|well V|go&PRES V|get&PRES PRO|it!
```

The first dependent tier indicates certain speech act codes and the second indicates a morphemic analysis with certain part of speech coding. Coding systems have been developed for some dependent tiers. Often, these codes begin with the symbol $. If there is more than one code, they can be put in strings with only spaces separating them, as in:

```
%spa:     $IMP $REF $INS
```

Multiple dependent tiers may be added in reference to a single main line, giving you as much richness in descriptive capability as is needed.

11.1 Standard Dependent Tiers

When possible, dependent tiers should be selected from the standard list of 3-letter tiers given here. However, if this list is inadequate, users can create extension tiers using three letters preceded by "x" as in %xtob for a tier that marks ToBI prosodic features. Here we list all of the dependent tier types that are used for child language data. It is unlikely that a given corpus would ever be transcribed in all of these ways. The listing that follows is alphabetical.

Action Tier %act:

This tier describes the actions of the speaker or the listener. Here is an example of text accompanied by the speaker's actions:

```
*ROS: I do it!
%act: runs to toy box
```

The %act tier can also be used in conjunction with the 0 symbol when actions are performed in place of speaking:

```
*ADA: 0.
%act: kicks the ball
```

This could also be coded as:

```
*ADA: 0 [% kicks the ball].
```

In this case the 0 on the main line is used to indicate that there is an action but no speech.

Or one can use the &= form, as in :

```
*ADA: &=kicks:ball .
```

The choice among these three forms depends on the extent to which the coder wants to keep track of a particular type of dependent tier information.

Addressee Tier %add:

This tier describes who talks to whom. Use the three-letter identifier given in the participants header to identify the addressees.

```
*MOT: be quiet.
%add: ALI, BEA
```

In this example, Mother is telling Alice and Beatrice to “be quiet.”

Alternate transcription tier %alt:

This tier is used to provide an alternative possible transcription. If the transcription is intended to provide an alternative for only one word, it may be better to use the main line form of this coding tier in the form [=? text].

CONNL Tier %cnl:

This tier is used for representing morphological categories in CONNL format to allow grammatical relations tagging using a CONNL tagger.

Coding Tier %cod:

This is the general purpose coding tier. It can be used for mixing codes into a single tier for economy or ease of entry. Here is an example.

```
*MOT: you want Mommy to do it?
```

```
%cod: $MLU=6 $NMV=2 $RDE $EXP
```

Cohesion Tier **%coh:**

This tier is used to code text cohesion devices.

Comment Tier **%com:**

This is the general purpose comment tier. One of its many uses is to note occurrence of a particular construction type, as in this example:

```
*EVE: that's nasty (.) is it?
%com: note tag question
```

Notations on this line should usually be in common English words, rather than codes. If special symbols and codes are included, they should be placed in quotation marks, so that CHECK does not flag them as errors.

Definitions Tier **%def:**

This tier is needed only for files that are reformatted from the SALT system by the SALTIN command.

English Rendition Tier **%eng:**

This line provides a fluent, nonmorphemicized English translation for non-English data.

```
*MAR: yo no tengo nada.
%eng: I don't have anything.
```

Error coding Tier **%err:**

This tier codes additional information about errors that cannot be fully expressed on the main line.

Explanation Tier **%exp:**

This tier is useful for specifying the deictic identity of objects or individuals. Brief explanations can also appear on the main line, enclosed in square brackets and preceded by the = sign and followed by a space.

Facial Gesture Tier **%fac:**

This tier codes facial actions. Ekman & Friesen (1969, 1978) have developed a complete and explicit system for the coding of facial actions. This system takes about 100 hours to learn to use and provides extremely detailed coding of the motions of particular muscles in terms of facial action units. Kearney and McKenzie (1993) have developed

computational tools for the automatic interpretation of emotions using the system of Ekman and Friesen.

Flow Tier %flo:

This tier codes a "flowing" version of the transcript that is as free as possible of transcription conventions and that reflects a minimal number of transcription decisions. Here is an example of a %flo line:

```
*CHI:     <I don't> [//] I don't wanna [: want to] look
    in a [* the] badroom [* bedroom] or Bill's room.
%flo:     I don't I don't wanna look in a badroom or Bill's room.
```

Most researchers would agree that the %flo line is easier to read than the *CHI line. However, it gains readability by sacrificing precision and utility for computational analyses. The %flo line has no records of retracings; words are simply repeated. There is no regularization to standard morphemes. Standard English orthography is used to give a general impression of the nature of phonological errors. There is no need to enter this line by hand, because FLO command can enter it automatically.

Gloss Tier %gls:

This tier can be used to provide a "translation" of the child's utterance into the adult language. Unlike the %eng tier, this tier does not have to be in English. It should use an explanation in the target language. This tier differs from the %flo tier in that it is being used not to simplify the form of the utterance but to explain what might otherwise be unclear. Finally, this tier differs from the %exp tier in that it is not used to clarify deictic reference or the general situation, but to provide a target language gloss of immature learner forms.

Gestural- Proxemic Tier %gpx:

This tier codes gestural and proxemic material. Some transcribers find it helpful to distinguish between general activity that can be coded on the %act line and more specifically gestural and proxemic activity, such as nodding or reaching, which can be coded on the %gpx line.

Grammatical Relations Tier %gra:

This tier is used to code dependency structures with tagged grammatical relations (Sagae, Davis, Lavie, MacWhinney, & Wintner, 2007; Sagae, Lavie, & MacWhinney, 2005; Sagae, MacWhinney, & Lavie, 2004).

Grammatical Relations Training %grt:

This tier is used for training of the MEGRASP grammatical relations tagger. It has the same form at the %gra tier.

Intonational Tier %int:

This tier codes intonations, using standard language descriptions.

Model Tier %mod:

This tier is used in conjunction with the %pho tier to code the phonological form of the adult target or model for each of the learner's phonological forms.

Morphological Tier %mor:

This tier codes morphemic segments by type and part of speech. Here is an example of the %mor tier:

```
*MAR: I wanted a toy.
%mor: PRO|I&1S V|want-PAST DET|a&INDEF N|toy.
```

Orthography Tier %ort:

This tier is used for languages with a non-Roman script. When Roman script is inserted on the main line, this line can be used for the local script. Or it can be used in the other way with local script on the main line and Roman on the %ort line. There should be a one-to-one correspondence between the items on the two lines.

Paralinguistics Tier %par:

This tier codes paralinguistic behaviors such as coughing and crying.

Phonology Tier %pho:

This dependent tier is used to provide a phonological transcription. When the researcher is attempting to describe phonological errors, the %err line should be used instead. The %pho line is to be used when the entire utterance is being coded in IPA. Here is an example of the %pho tier in use.

```
*SAR: I got a boo+boo.
%pho: ai gæt ə bubu
```

Transcription on the %pho line should be done using the IPA symbols in Unicode.

Words on the main tier should align in a one-to-one fashion with forms on the phonological tier. This alignment takes all forms produced into account and does not exclude retraces or non-word forms. On the %pho line it is sometimes important to describe several words as forming a single phonological group in order to describe liaison and other assimilation effects within the group. To mark this, the Unicode characters for U+2039 and U+203A, which appear as ‹ and ›, should be entered on both the main and %pho lines using F2+< and F2+>.

Signing Tier %sin:

Parents of deaf children often sign or gesture along with speech, as do the children themselves. To transcribe this, researchers often place the spoken material on the main line and the signed material on the %sin line. Words on the %sin tier can consist of any alphanumberic characters and colons, as in forms such as g:point:toy. Like the %pho and %mor tiers, the words on the %sin tier must be placed into one-to-one correspondence with words on the main tier. To do this, it may be necessary to enter many "0" forms on the %sin tier when a word is not matched by a sign or gesture. At other times, several words on the main tier may align with a single gesture. To mark this grouping, you can group the forms on the main line with two Unicode bracketing symbols. The beginning of the group is marked by Unicode U+23A8 ⎨ and the close by Unicode U+23AC ⎬.

Situation Tier %sit:

This tier describes situational information relevant only to the utterance. There is also an @Situation header. Situational comments that relate more broadly to the file as a whole or to a major section of the file should be placed in a @Situation header.

```
*EVE: what that?
*EVE: woof@o woof@o.
%sit: dog is barking
```

Speech Act Tier %spa:

This tier is for speech act coding. Many researchers wish to transcribe their data with reference to speech acts. Speech act codes describe the function of sentences in discourse. Often researchers express a preference for the method of coding for speech acts. Many systems for coding speech acts have been developed. A set of speech act codes adapted from a more general system devised by Ninio and Wheeler is provided in the chapter on speech act coding.

Timing Tier %tim:

This tier is used for older data for which there is no possible linking of the transcript to the media. It should not be confused with the millisecond accurate timing found in the bullets inserted by sonic CHAT. The %tim tier is used just to mark large periods of time during the course of taping. These readings are given relative to the time of the first utterance in the file. The time of that utterance is taken to be time 00:00:00. Its absolute time value can be given by the @Time Start header. Elapsed time from the beginning of the file is given in hours:minutes:seconds. Thus, a %tim entry of 01:20:55 indicates the passage of 1 hour, 20 minutes, and 55 seconds from time zero. If you only want to track time in minutes and seconds, you can use the form minutes:seconds, as in 09:22 for 9 minutes and 22 seconds. None of the CLAN programs use the information encoded in the %tim tier. It is just included for hand analyses.

```
*MOT:     where are you?
%tim:     00:00:00
```

```
... (40 pages of transcript follow and then)
*EVE:     that one.
%tim:     01:20:55
```

If there is a break in the interaction, it may be necessary to establish a new time zero. This is done by inserting a new @Time Start header. You can also use this tier to mark the beginning and end of a time period by using a form such as:

```
*MOT:     where are you?
%tim:     04:20:23-04:21:01
```

Training Tier **%trn:**

This is the training tier for the POST tagger. It has the same form as the %mor line.

11.2 Synchrony Relations

For dependent tiers whose codes refer to the entire utterance, it is often important to distinguish whether events occur before, during, or after the utterance.

Occurrence Before **< bef >**

If the comment refers to something that occurred immediately before the utterance in the main line, you may use the symbol <bef>, as in this example:

```
*MOT:     it is her turn.
%act:     <bef> moves to the door
```

Occurrence After **< aft >**

If a comment refers to something that occurred immediately after the utterance, you may use the form <aft>. In this example, Mother opened the door after she spoke:

```
*MOT:     it is her turn.
*MOT:     go ahead.
%act:     <aft> opens the door
```

If neither < bef > or < aft > are coded, it is assumed that the material in the coding tier occurs during the whole utterance or that the exact point of its occurrence during the utterance is not important.

Although CHAT provides transcribers with the option of indicating the point of events using the %com tier and <bef> and <aft> scoping, it may often be best to use the @Comment header tier instead. The advantage of using the @Comment header is that it indicates in a clearer manner the point at which an activity actually occurs. For example, instead of the form:

```
*MOT:     it is her turn.
%act:     <bef> moves to the door
```

one could use the form:

```
@Comment: Mot moves to the door.
```

```
*MOT:     it is her turn.
```

The third option provided by CHAT is to code comments in square brackets right on the main line, as in this form:

```
*MOT:     [^ Mot moves to the door] it is her turn.
```

Of these alternative forms, the second seems to be the best in this case.

Scope on Main Tier $sc=n

When you want a particular dependent tier to refer to a particular word on the main tier, you can use this additional code to mark the scope. For example, here the code marks the fact that the mother's words 4 through 7 are imitated by the child.

```
*MOT:     want to come sit in my lap?
%act:     $sc=4-7 $IMIT
*CHI:     sit in my lap.
```

12 CHAT-CA Transcription

CHAT also allows transcription that is more closely in accord with the requirements of CA (Conversational Analysis) transcription. CA is a system devised by Sacks, Schegloff, and Jefferson (Sacks, Schegloff, & Jefferson, 1974) for the purpose of understanding the construction of conversational turns and sequencing. It is now used by hundreds of researchers internationally to study conversational behavior. Recent applications and formulations of this approach can be found in Ochs, Schegloff, and Thompson (1996), as well as the related "GAT" formulation of Selting (1998). Workers in this tradition find CA notation easier to use than CHAT, because the conventions of this system provide a clearer mapping of features of conversational sequencing. On the other hand, CA transcription has limits in terms of its ability to represent conventional morphemes, orthography, and syntactic patterns. By supplementing CHAT transcription on the word level with additional utterance level codes for CA, the strengths of both systems can be maintained. To achieve this merger, some of the forms of both CHAT and CA must be modified. To implement CA format, CHAT-CA uses these functions:

1. The fact that a transcript is using CA notation is indicated by inserting the term CA in the @Options header. Older corpora can be maintained in their original non-CHAT format by entering the word "heritage" on the @Options header tier before the @ID tiers.
2. Utterances and inter-TCU pauses are numbered by the automatic line numbering function.
3. Line numbers can be turned on and off for viewing and printing by using CLAN options. Line numbers are not stored by themselves in CHAT, although they are encoded in the XML version of CHAT.
4. After the line number comes an asterisk and then the speaker ID code and a colon and a tab, as in CHAT format.
5. Tabs are not used elsewhere.
6. CA Overlaps, as marked with the special symbols ⌈ ⌉ ⌊ and⌋, are aligned automatically by the INDENT program, so hand indentation is not needed.
7. To maintain proper alignment, CLAN uses a special fixed-width Unicode font.
8. CHAT requires obligatory utterance terminators. However, CA uses terminal contours instead, as noted in the table below, and these are optional.
9. Instead of marking comments in double parentheses, CHAT uses the [% com] notation. However, common sounds, gestures, and activities occurring at a point in an utterance are marked using the &=gesture form.
10. CHAT uses the following forms for marking disfluencies, as further discussed in the next chapter.
11. pairs of Unicode 21AB leftwards arrow with loop to mark initial segment repetition as in ↫b-b-b↫boy
12. pairs of Unicode 2260 not-equal-to sign to mark blocked segments as in ru≠b-b-b≠bber
13. the colon for marking drawls or extensions
14. the ^ symbol for marking a break inside a work
15. forms such as &-um for marking filled pauses

16. silent pauses as marked by (.) or (0.6) etc.
17. [/] string for word or phrase repetition
18. string [//] string for retracing
19. +… for trailing off

In addition to these basic utterance-level CA forms, CHAT-CA requires the standard CHAT headers such as these:

@Begin and @End. Using these guarantees that the file is complete.
@Comment: This is a useful general purpose field
@Bg, @Eg: These mark "gems" for later retrieval
@Participants: This field identifies the speakers.
%gpx: dependent tiers such as %gpx, %spa can be added as needed.

Gail Jefferson continually elaborate the coding of CA features through special marks during her career. Her creation of new marks was limited, for many years, by what was available on the typewriter. With the advent of Unicode, we are able to capture all of the marks she had proposed along with others that she occasionally used. The following table summarizes these marks of CHAT-CA.

	Character Name	Char	Function	F1 +	Unicode
1	up-arrow	↑	shift to high pitch	up arrow	2191
2	down-arrow	↓	shift to low pitch	down arrow	2193
3	double arrow tilted up	⇗	rising to high	1	21D7
4	single arrow tilted up	↗	rising to mid	2	2197
5	level arrow	→	level	3	2192
6	single arrow tilted down	↘	falling to mid	4	2198
7	double arrow down	⇘	falling to low	5	21D8
8	infinity mark	∞	unmarked ending	6	221E
9	double wavy equals	≈	+≈ no break continuation	=	2248
10	triple wavy equals	≋	+≋ technical continuation	+	224B
11	triple equal	≡	≡uptake (internal)	u	2261
12	raised period	·	inhalation	.	2219
13	open bracket top	⌈	top begin overlap	[	2308
14	close bracket top	⌉	top end overlap	]	2309
15	open bracket bottom	⌊	bottom begin overlap	shift [	230A
16	closed bracket bottom	⌋	bottom end overlap	shift]	230B
17	up triangle	∆	∆faster∆	right arrow	2206
18	down triangle	∇	∇slower∇	left arrow	2207
19	low asterisk	⁑	⁑creaky⁑	*	204E
20	double question mark	⁇	⁇unsure⁇	/	2047
21	degree sign	°	°softer°	zero	00B0
22	fisheye	◉	◉louder◉	)	25C9
23	low bar	▁	▁low pitch▁	d	2581
24	high bar	▔	▔high pitch▔	h	2594
25	smiley	☺	☺smile voice☺	l	263A
26	double integral	∬	∬whisper∬	w	222C

27	upsilon with dialytika	Ÿ	Ÿ yawn Ÿ	y	03AB
28	clockwise integral	∮	∮ singing ∮	s	222E
29	section marker	§	§ precise§	p	00A7
30	tilde	∾	constriction∾	n	223E
31	half circle	↻	↻pitch reset	r	21BB
32	capital H with dasia	Ἡ	laugh in a word	c	1F29
33	lower quote	„	tag or final particle	t	201E
34	double dagger	‡	vocative or summons	v	2021
35	dot	ạ	Arabic dot	,	0323
36	raised h	ʰ	Arabic aspiration	H	02B0
37	macron	ā	stressed syllable	-	0304
38	glottal	ʔ	glottal stop	q	0294
39	reverse glottal	ʕ	Hebrew glottal	Q	0295
40	caron	š	caron	;	030C

The column marked F1 in the previous table gives methods for inserting the various non-ASCII Unicode characters. For example the smile voice symbol is ☺ is inserted by F1 and then the letter l. It must be used both before and after the stretch of material with the smile or laughing voice. After row 32, the items are inserted using F2, instead of F1. For the most recent version of this symbol set, please consult the current list on the web. Of these various symbols, there are four that must be placed either at the beginning of words or inside words. These include the arrows for pitch rise and fall, the inverted question mark for inhalation, and the ≡ symbol for quick TCU internal uptake. The paired symbols for intonational stretches such as louder, faster, and slower can be placed anywhere, except inside comments. They must be used in pairs to mark the beginning and end of the feature in question.

The triple wavy symbol with a plus (+≋) is used to mark a break in a TCU caused by interruption from another speaker. Use of this symbol can improve readability and overlap alignment. In this case the triple wavy without a plus can be placed at the end of the last word of the first segment and then at the beginning of the continuation, where it is joined with a plus sign and followed by a space, as in +≋ . The ≈ symbol is used in a parallel way to mark a TCU continuation that is not forced by an interruption from another speaker. It occurs at the end of the last word of the first segment and in the form +≈ with a following space at the beginning of the following line. CA transcribers can also use underlining to represent emphasis on a word or a part of a word. However, if text is taken from a CHAT file to Word the underlining will be lost.

In general, CA marks must occur either inside words or at the beginnings or ends of words. In most cases, they should not occur by themselves surrounded by spaces. The exception to this is the utterance continuator mark +≋ which should be preceded by the tab mark and followed by a space.

In addition to these features that are basic to CA, our implementation requires transcribers to begin their transcript with an @Begin line and to end it with an @End line. Comments can be added using the @Comment format, and transcribers should use the @Participants header in this form:

```
@Participants:   geo, mom, tim
```

This line uses only three-letter codes for participant names. By adding this line, it is possible to have quicker entry of speaker codes inside the editor.

13 Disfluency Transcription

CHAT uses the following forms for marking disfluencies.

Stuttering-like disfluencies	**Code**	**Example**	**Notes**
prolongation	:	s:paghetti	Place after prolonged segment
broken word	^	spa^ghetti	Pause within word
blocking	≠	≠butter	A block before word onset
repeated segment	↫	↫r-r-r↫rabbit OR like↫ike↫	The curly left arrow brackets the repetition; iterations are marked with hyphens
lengthened repeated segment	↫ and doubling	↫rr-rr-r↫rabbit	The doubling of "r" indicates lengthening of the "r" segment
phonological fragment	&+	&+sn dog	Changes from “snake” to “dog”
other non-word strings	&	&gara	Word play etc.
Typical Disfluencies			
whole word repetition	follow word with [/]	butter [/] butter	Repeated word counts once
multiple whole word repetition	indicate number of repetitions in brackets	butter [x 7]	Indicates that the word ‘butter’ was repeated seven times
phrase repetition	<> [/]	<that is a> [/] that is a dog.	< > is used to mark repeated material
word revision	[//]	a dog [//] beast	Revision counts once
phrase revision	<> [//]	<what did you> [//] how can you see it ?	Revision counts once
pause	(.) or (..) or (…)	(.)	Counts the number of short, medium, long pauses
pause duration	(2.4)	(2.4)	Adds up the time values, if marked
filled pause	&-	&-um &-you_know	Fillers with underscore count as one word

The ≠ character to mark blocking is entered by typing F2 and =
The ↫ character to mark segment repetition is entered by typing F2 and /
Blocking of filled pauses is indicated in this way: &-≠you_know

These disfluency types can be traced in FREQ and KWAL commands through the search strings given in the files called fluency-sep.cut and fluency-comb.cut in the

/lib/fluency folder in CLAN. They can be counted automatically using the FLUCALC program.

14 Transcribing Aphasic Language

Here are some tips for transcribing typical features of aphasic language. These conventions are all discussed elsewhere in this manual, but are repeated here for the convenience of researchers and clinicians working specifically with speech from persons with aphasia.

Commas can be used as needed to mark phrasal junctions, but they are not used by the programs and have no tight prosodic definition.

Fragments (phonological) get entered with the ampersand-plus symbol attached at the beginning. So, for all incomplete words, use &+ followed by the graphemes that capture the sounds produced.

```
*PAR:     so now I can &+sp speak a little bit.
*PAR:     and then &+sh &+s &+w we came home.
```

If you want to mark disfluencies more precisely, you should use the codes in the preceding section on Disfluency coding.

Gestures can be captured in several ways. You can compose codes using parts of the body to indicate head nods and shakes, for example, using the ampersand, the equal sign, the body part, colon, and then the movement or its meaning. You can use up to two colons for each gesture code and you can use more than one word after the colon if you connect the words with an underscore symbol.

```
*PAR:     &=head:no .
*PAR:     &=hand:hello .
*PAR:     see you later &=ges:wave .
*PAR:     the woman &=ges:fishing fishing pole water &=casts:pole .
```

You can also use the %fac and %gpx codes for facial or bodily gestures that extend throughout longer periods, including the whole sentence.

```
*PAR:     she was fish [/] fish.
%gpx:     raising her arm up and down
```

The various ways of marking **Incomplete utterances** are described in section 8.11 on special utterance terminators.

Interjections, **Exclamations, and Interactional Markers** are all called communicators or "co" in the MOR grammar. The complete list of all word forms recognized by MOR is given in the 0allwords.cex file at the top of the ENG-MOR grammar. To see just the list of communicator forms, like in the files in the /lex folder that begin with "co".

Fillers are listed in the file co-fil.txt in that same folder. There are just a few of these. They are all entered in this format &-uh or &-um.

```
*INV:  how do you think your language is these days?
*PAR:  well &-uh &-uh pretty good.
```

If a speaker **laughs** or **sighs**, for example, and you want to capture that, you can transcribe it with the ampersand and equal sign.

```
*PAR:    well &=laughs tell you the truth, I can't say what I said.
```

You can put the laugh or sigh on its own line if it serves as the speaker's turn.

```
*PAR:    &=laughs .
```

A list of these *Simple Events* appears in the CHAT manual and includes cough, groan, sneeze, etc.

Neologisms can be marked by putting the @n symbols next to the neologism.

```
*PAR:    oh yes, this is a little sakov@n that's all.
```

Overlapping speakers can be handled in several ways. The easiest is to use a lazy overlap marking +< at the beginning of the utterance that overlapped the previous utterance. This indicates that the second utterance overlapped the previous one, but it doesn't indicate exactly which words were overlapped. There are other ways to handle overlaps that you can learn about from the manual as you become more familiar with the program.

```
*PAR:    that's about.
*INV:    +< what about that?
```

Paraphasias can be marked as errors with an asterisk inside square brackets. If you know the intended target word, you can indicate it in square brackets next to the error. If the error is a non- word, you can transcribe it in IPA symbols or you can transcribe it orthographically. If IPA symbols are used, add @u to the end of the error. The CLAN program will use these replacement words when creating the morphological tier. If 2 colons are used, the morphological tier will use the error word and not the replacement word.

```
*PAR:    no dubs [: dogs] [*] allowed in the cemetery.
*PAR:    the pɪnts@u [: prince] [*] wants to know who the slipper
    fits.
```

Pauses can be captured in the transcription by using a period inside parentheses -- (.) indicates an unfilled pause, (..) indicates a longer pause, and (…) indicates a very long pause. The CHAT manual explains how to code the exact length of a pause in a section entitled *Pauses*. Also, if you want to distinguish fluent from disfluent pauses, you can use (.)d for the latter.

```
*PAR:    I don't (.) know.
*PAR:    (…) what do you (…) think?
```

If you want to be exact, you can code the length of the pauses and enter the minutes, seconds, and parts of seconds within the parentheses. Minutes precede the colon, seconds follow the colon, and parts of seconds are given after the period symbol. The following examples code pauses lasting .5 seconds, 1 minute and 13.41 seconds, and 2 seconds,

respectively. If you are not coding minutes, you do not need the colon at all. Most likely, the final example of 2 seconds illustrates the type of pause coding that would be most relevant.

```
*PAR:     I don't (0.5) know.
*PAR:     (1:13.41) what do you (2.) think?
```

Quoted material is likely to occur during story telling and similar activities. To mark material as quoted, special symbols are used. The +"/. symbols are used at the end of the sentence that precedes the quoted material. The +" symbols are used to begin the next line, which contains a complete clause or sentence of quoted material.

```
*PAR: and so the prince found the slipper.
*PAR: and he said +"/.
*PAR: +" my_gosh I can find the lady who is fitting this slipper.
```

If the quoted material continues for more lines, use +" at the beginning of each quoted line.

If the quote precedes the main clause, use +" at the beginning of the quote and then use +"/. at the end of the main clause.

```
*PAR:     +" my_gosh I can find the lady who is fitting this slipper.
*PAR:     the prince said +".
```

Repetitions are called *Retracing Without Correction*. The material that is repeated is enclosed in angle brackets (< >) and followed immediately by the square brackets ([/]) with one slash mark enclosed. If only one word has been repeated once, angle brackets are not needed and CLAN will assume that the one word before the square brackets with the slash was repeated. You do not need to use angle brackets or square brackets with the slash mark when fillers (e.g., uh, um) are repeated.

```
*PAR:     <it was> [/] it was so bad.
*PAR:     and the [/] the window was open.
*PAR:     and she &+s spilled <the the the> [/] &-uh &-uh the water
    on    the floor.
```

You can indicate several repetitions of a single word by using the square brackets and inserting an *x*, a space, and the number of times the word was repeated.

```
*PAR:     it's [x 4] &-um a dog.
```

Revisions are called Retracing With Correction and occur when the speaker changes something (usually the syntax) of an utterance but maintains the same idea. The material being retraced is enclosed in angle brackets, followed immediately by the square brackets with 2 slash marks enclosed. If only one word has been changed, angle brackets are not needed and CLAN will assume that the one word before the square brackets with the slash was revised. A change, or correction, should be something clearly identifiable that changes the syntax but maintains the same idea of the phras

```
*PAR:     well <Cinderella was a> [//] &-uh Cinderella is a nice
          girl.
*PAR:     and then sometimes we [//] I was scared about the traffic.
```

Self-interruptions occur when a speaker breaks off an utterance and starts up another. These are coded using +//. (or +//? for a question).

```
*PAR:  well then the [//] &-uh you_know <the the> [/] the airplane
       that [/] that his +//.
*PAR:  no the airplane that &-uh landed +//.
*PAR:  no [/] no that's not right.
```

Invited interruptions occur when one speaker prompts the other speaker to complete an utterance. These are coded using the +… symbols for trailing off and the ++ symbols for the other speaker's completion. This may be intentional (cuing) or unintentional.

```
*INV: how about &+ra +…
*PAR: ++ a radio.

*HEL: if Bill had known +…
*WIN: ++ he would have come.
```

Shortenings occur when a speaker drops sounds out of words. For example, a speaker may leave the final "g" off of "running", saying "runnin" instead. In CHAT, this shortened form should appear as runnin(g). Other examples that demonstrate sound omissions are (be)cause, prob(ab)ly, (a)bout, (re)member, (ex)cept.

Assimilations include words such as “gonna” and “kinda”. Most of these will be recognized by CLAN so no replacements (e.g., [: going to]) are needed. Tables with lists of shortenings and assimilations appear in the CHAT manual in sections 6.6.7 and 6.6.8, respectively. The most updated records, however, are always in the MOR lexicon.

Unintelligible segments of utterances should be transcribed as xxx.

Untranscribed material can be indicated with the letters www. This symbol is used on a main line to indicate material that a transcriber does not want to transcribe because it is not relevant to the interaction of interest. This symbol must be followed by the %exp line, explaining what was transpiring.

```
*PAR:     www.
%exp:     talking to spouse

*PAR:     www.
%exp:     looking through pictures
```

Utterance segmentation decisions can be challenging. See the guidelines in the first section of the chapter on Utterances in this manual.

15 Arabic and Hebrew Transcription

In order to transcribe Arabic and Hebrew in Roman characters, we make use of five special characters that can be entered in the CHAT editor in this way:

1. For the superscript h, type F2 and then h.
2. For the subscript dot, type F2 and then comma. This is also used to mark schwa.
3. For the macron on a Hebrew stressed vowel type F2 and dash (-).
4. For the Hebrew glottal type F2 and Q.
5. For the basic glottal stop symbol, type F2 and q.
6. For long vowels, type F2 and then : (colon) to insert the triangular Unicode colon 02D0, rather than the standard colon which is Unicode 003A.

Also, to allow for marking of Hebrew and Arabic prefixes, the # sign is allowed at the end of the prefix, which is then separated from the stem by a space, as in we# tiqfōc.

When transcribing geminates use double consonants or double vowels. This system is expressed in the following two charts:

Vowels

IPA	Arabic	Name	CHAT
iː	◌ِي	ya	ii
ɪ, i	◌ِ	kasra	i
eː	◌ِي	ya (ba'den)	ee
e		-	e
aː	ا	alef madda emphatic	aa
a, ɑ	ا	short a	a
æ	◌َ	fatHa	ae
æː		alef madda non-emphatic	æː
uː	و	waw, long	uu
u	و	waw, short	
ʊ	◌ُ	dame	u
oː	و	waw (bantaloːn)	oo
o, ɔ	و	short	o
ə		not in Arabic	ẹ

Consonants

IPA	Arabic	Name	CHAT
ʔ	إ	hamza	ʔ
b	ب	ba	b
p			p
t	ت	ta	t
θ	ث	tha	tʰ
ʒ	.چ	jim	j
ħ	ح	ḥa	ḥ
x	خ	xa	kʰ
χ			qʰ
d	د	dal	d
ð	ذ	dhal	dʰ
r	ر	ra	r
z	ز	zen	z
s	س	sin	s
ʃ	ش	shin	sʰ
sˤ	ص	sad	ṣ
dˤ	ض	dad	ḍ
tˤ	ط	ṭa	ṭ
zˤ	ظ	ẓa	ẓ
ʕ	ع	‘ayn	ʕ
ɣ	غ	ghayn	gʰ
f	ف	fa	f
q	ق	qaf	q
g	ج	gim	g
k	ك	kaf	k
l	ل	lam	l

m	م	mim	m
n	ن	nun	n
h	ه	ha	h
w	و	waw	w
j	ي	ya	y
ʧ			ts^h
ʤ			dj
v			v

16 Specific Applications

The basic CHAT codes can be adapted to work with a variety of more specific applications. In this chapter, we refer four such applications to illustrate the adaptation of the general codes to specific uses. A separate document, available from this server, describes the BTS (Berkeley Transcription System) for sign language.

When codes cannot be adapted for specific projects, it may be necessary to modify the underlying XML schema for CHAT. When this becomes necessary, please send email to macw@cmu.edu.

16.1 Code-Switching

Transcription is easiest when speakers avoid overlaps, speak in full utterances, and use a single standard language throughout. However, the real world of conversational interactions is seldom so simple and uniform. One particularly challenging type of interaction involves code-switching between two or even three different languages. In some cases, it may be possible to identify a default language and to mark a few words as intrusions into the default language. In other cases, mixing and switching are more intense.

CHAT relies on a system of interlaced marking for identifying the languages being used in code-switched interactions.

1. The languages spoken by the various participants must be noted with the @Languages header tier. See section 7.2 for the relevant ISO-639 codes. The first language on this line is considered to be the default language until a switch is marked.
2. Utterances that represent a switch to the second language are marked with precodes, as in [- eng] for a switch to English. Here is an example:

```
*MOT:     can you see?
*CHI:     [-spa] no puedo.
```

3. Individual words that switch away from the default language to the second language are marked with the @s terminator. If the @Languages header has "spa, eng", then the @s marked indicates a swith to English. If the @Languages header has "eng, spa" then the @s indicates a switch to Spanish. If the switch is to a language not included in the @Languages header, then the full form must be used as in word@s:por for switch to a Portuguese word.
4. When the default language of the interaction changes, the change can be marked with @New Language.

The @s special form marker code may also be used to explicitly mark the use of a particular language, even if it is not included in the @Languages header. For example, the code schlep@s:yid can be used to mark the inclusion of the Yiddish word "schlep" in any text. The @s code can also be further elaborated to mark code-blended words. The form well@s:eng&cym indicates that the word "well" could be either an English or a Welsh word. The combination of a stem from one language with an inflection from another can be marked using the plus sign as in swallowni@s:eng+hun for an English

stem with a Hungarian infinitival marking. All of these codes can be followed by a code with the $ sign to explicitly mark the parts of speech. Thus, the form recordar@s$inf indicates that this Spanish word is an infinitive. The marking of part of speech with the $ sign can also be used without the @s.

These techniques are all designed to facilitate the retrieval of material in one language separately from the other without having to tag each and every word. However, if one wants to see tags on every word, a transcript created using the above rules can be reformatted using this command, in which the –l switch adds language tags to every word:

```
kwal +d +t* +t@ +t% -l filename.cha
```

Relying further on the –l switch, it is possible to locate code-switches on the utterance level in a transcript by using a COMBO command of this type for switches from English to French:

```
combo +b2 -l +s"\**:^*s:eng^*^\**:^*s:fra" *.cha
```

Problems similar to those involved in code-switching occur in studies of narratives where a speaker may assume a variety of roles or voices. For example, a child may be speaking either as the dragon in a story or as the narrator of the story or as herself. These different roles are most easily coded by marking the six-character main line code with forms such as *CHIDRG, *CHINAR, and *CHISEL for child-as-dragon, child-as-narrator, and child-as-self.

16.2 Elicited Narratives and Picture Descriptions

Often researchers use a set of structured materials to elicit narratives and descriptions. These may be a series of pictures in a story book, a set of photos, a film, or a series of actions involving objects. The transcripts that are collected during this process can be studied most easily by using gem notation. The simplest form of this system, a set of numbers are used for each picture or page of the book. Here is an example from the beginning of an Italian file from the Bologna frog story corpus:

```
@G: 1
*AND:     questo e' un bimbo poi c' e' il cane e la rana.
*AND:     questa e' la casa.
@G: 2
*AND:     il bimbo dorme.
```

The first @G marker indicates the first page of the book with the boy, the dog, and the frog. The second @G marker indicates the second page of the book with the boy sleeping. When using this lazy gem type of marking, it is assumed that the beginning of each new gem is the end of the previous gem. Programs such as GEM and GEMLIST can then be used to facilitate retrieval of information linked to particular pictures or stimuli.

16.3 Written Language

CHAT can also be adapted to provide computerized records of written discourse. Typically, researchers are interested in transcribing two types of written discourse: (1) written productions produced by school students, and (2) printed texts such as books and newspapers. This format is particularly useful for coding written productions by school

children. In order to use CHAT effectively for this purpose, the following adaptations or extensions can be used.

The basic structure of a CHAT file should be maintained. The @Begin and @End fields should be kept. However, the @Participant line should look like this:

```
@Participants:  TEX Writer's_Name Text
```

Each written sentence should be transcribed on a separate line with the *TEX: field at the beginning. Additional @Comment and @Situation fields can be added to add descriptive details about the writing assignment and other relevant information.

For research projects that do not demand a high degree of accurate rendition of the actual form of the written words, it is sufficient to transcribe the words on the main line in normalized standard-language orthographic form. However, if the researcher wants to track the development of punctuation and orthography, the normalized main line should be supplemented with a %spe line. Here are some examples:

```
*TEX:     Each of us wanted to get going home before the Steeler's
    game let out .
%spe:     etch of /us wanted too git goin home *,
    be/fore the Stillers game let out 0.
```

In this example, the student had written “ofus” without a space and had incorrectly placed a space in the middle of “before”. The slash at the beginning of a word marks an omission and the internal slash marks an extra space. These two marks are used to achieve one-to-one alignment between the main line and the %spe line. This alignment can be used to facilitate the use of MODREP in the analysis of orthographic errors. It will also be used in the future by programs that perform automatic comparisons between the main line and the %spe line to diagnose error types.

The only purpose of the %spe line is to code word-level spelling errors, not to code any higher level grammatical errors or word omissions. Also, the words on the main line are all given in their standard target-language orthographic form. For clarity, final punctuation on the main line is preceded by a space. If a punctuation mark is omitted, it is coded with a zero. Forms that appear on the %spe line that have no role in the main line, such as extraneous punctuation, are marked with an asterisk.

These conventions focus on the writing of individual words. However, it may also be necessary to note larger features of composition. When the student crosses off a series of words and rewrites them, you can use the standard CHAT conventions for retracing with scoping marked by angle brackets and the [//] symbol. If you want to mark page breaks, you can use a header such as @Stim: Page 3. If you wish to mark a shift in ink, or orthographic style, you can use a general @Comment field.

16.4 Sign and Speech

CHAT can also be used to analyze interactions that combine signed and spoken language. For example, they may occur in the input of hearing parents to deaf children or with hearing children interacting with deaf parents. This system of transcription uses the %sin line to represent signs and gestures. Gestures are tagged with g: (e.g., "g:cat") which means a gesture for "cat." This code can be further elaborated in a form like "g:cat:dpoint" to indicate that the gesture was a deictic point (or g:cat:icon for an iconic

gesture, g:cat:dreq for a deictic request, etc.). For sign, a form like "s:cat" to indicate that the child signed "cat." There should be a one-to-one correspondence between the main line and the %sin line. This can be done by using 0 to mark cases where only one form is used:

*CHI: 0 .
%sin: g:baby:dpoint
The child gestured without any speech.

*CHI: baby .
%sin: g:baby:dpoint
The child pointed at the baby at the same time she said *baby*.

*CHI: baby 0 .
%sin: 0 g:baby:dpoint
The child said *baby* and then pointed at the baby.

*CHI: baby 0 .
%sin: g:baby:dpoint s:baby
The child said *baby* while pointing at the baby and then signed baby.

17 Speech Act Codes

One way of coding speech acts is to separate the component of illocutionary force from those aspects that deal with interchange types. One can also distinguish a set of codes that relate to the modality or means of expression. Codes of these three types can be placed together on the %spa tier. One form of coding precedes each code type with an identifier, such as "x" for interchange type and "i" for illocutionary type. Here is an example of the combined use of these various codes:

```
*MOT:     are you okay?
%spa:     $x:dhs $i:yq
```

Alternatively, one can combine the codes in a hierarchical system, so that the previous example would have only the code $dhs:yq. Choice of different forms for codes depends on the goals of the analysis, the structure of the coding system, and the way the codes interface with clan.

Users will often need to construct their own coding schemes. However, one scheme that has received extensive attention is one proposed by Ninio & Wheeler (1986). Ninio, Snow, Pan, & Rollins (1994) provided a simplified version of this system called INCA-A, or Inventory of Communicative Acts - Abridged. The next two sections give the categories of interchange types and illocutionary forces in the proposed INCA-A system.

17.1 Interchange Types

Interchange Type Codes

Code	Function	Explanation
CMO	comforting	to comfort and express sympathy for misfortune
DCA	discussing clarification of action	to discuss clarification of hearer's nonverbal communicative acts
DCC	discussing clarification of communication	to discuss clarification of hearer's ambiguous verbal communication or a confirmation of the speaker's understanding of it
DFW	discussing the fantasy world	to hold a conversation within fantasy play
DHA	directing hearer's attention	to achieve joint focus of attention by directing hearer's attention to objects, persons, and events
DHS	discussing hearer's sentiments	to hold a conversation about hearer's nonobservable thoughts and feelings
DJF	discussing a joint focus of attention	to hold a conversation about something that both participants are attending to, e.g., objects, persons, ongoing actions of hearer and speaker, ongoing events
DNP	discussing the nonpresent	to hold a conversation about topics that are not observable in the environment, e.g., past and future events and actions, distant objects and persons, abstract matters (excluding inner states)

DRE	discussing a recent event	to hold a conversation about immediately past actions and events
DRP	discussing the related-to-present	to discuss nonobservable attributes of objects or persons present in the environment or to discuss past or future events related to those referents
DSS	discussing speaker's sentiments	to hold a conversation about speaker's nonobservable thoughts and feelings
MRK	marking	to express socially expected sentiments on specific occasions such as thanking, apologizing, or to mark some event
NCS	negotiate copresence and separation	to manage the transition
NFA	negotiating an activity in the future	to negotiate actions and activities in the far future
NIA	negotiating the immediate activity	to negotiate the initiation, continuation, ending and stopping of activities and acts; to direct hearer's and speaker's acts; to allocate roles, moves, and turns in joint activities
NIN	noninteractive speech	to engage in private speech or produces utterances not addressed to present hearer
NMA	negotiate mutual attention	to establish mutual attentiveness and proximity or withdrawal
PRO	performing verbal moves	to perform moves in a game or other activity by uttering the appropriate verbal forms
PSS	negotiating possession of objects	to discuss who is the possessor of an object
SAT	showing attentiveness	to demonstrate that speaker is paying attention to the hearer
TXT	reading written text	to read or recite written text aloud
OOO	unintelligible	to mark unintelligible utterances
YYY	uninterpretable	to mark uninterpretable utterances

17.2 Illocutionary Force Codes

Directives

AC Answer calls; show attentiveness to communications.
AD Agree to carry out an act requested or proposed by other.
AL Agree to do something for the last time.
CL Call attention to hearer by name or by substitute exclamations.
CS Counter-suggestion; an indirect refusal.
DR Dare or challenge hearer to perform an action.
GI Give in; accept other's insistence or refusal.
GR Give reason; justify a request for an action, refusal, or prohibition.
RD Refuse to carry out an act requested or proposed by other.
RP Request, propose, or suggest an action for hearer, or for hearer and speaker.

RQ Yes/no question or suggestion about hearer's wishes and intentions
SS Signal to start performing an act, such as running or rolling a ball.
WD Warn of danger.

Speech Elicitations

CX Complete text, if so demanded.
EA Elicit onomatopoeic or animal sounds.
EI Elicit imitation of word or sentence by modelling or by explicit command.
EC Elicit completion of word or sentence.
EX Elicit completion of rote-learned text.
RT Repeat or imitate other's utterance.
SC Complete statement or other utterance in compliance with request.

Commitments

FP Ask for permission to carry out act.
PA Permit hearer to perform act.
PD Promise.
PF Prohibit/forbid/protest hearer's performance of an act.
SI State intent to carry out act by speaker.
TD Threaten to do.

Declarations

DC Create a new state of affairs by declaration.
DP Declare make-believe reality.
ND Disagree with a declaration.
YD Agree to a declaration.

Markings

CM Commiserate, express sympathy for hearer's distress.
EM Exclaim in distress, pain.
EN Express positive emotion.
ES Express surprise.
MK Mark occurrence of event (thank, greet, apologize, congratulate, etc.).
TO Mark transfer of object to hearer.
XA Exhibit attentiveness to hearer.

Statements

AP Agree with proposition or proposal expressed by previous speaker.
CN Count.
DW Disagree with proposition expressed by previous speaker.
ST Make a declarative statement.
WS Express a wish.

Questions

AQ Aggravated question, expression of disapproval by restating a question.

AA Answer in the affirmative to yes/no question.
AN Answer in the negative to yes/no question.
EQ Eliciting question (e.g., hmm?).
NA Intentionally nonsatisfying answer to question.
QA Answer a question with a wh-question.
QN Ask a product-question (wh-question).
RA Refuse to answer.
SA Answer a wh-question with a statement.
TA Answer a limited-alternative question.
TQ Ask a limited-alternative yes/no question.
YQ Ask a yes/no question.
YA Answer a question with a yes/no question.

Performances

PR Perform verbal move in game.
TX Read or recite written text aloud.

Evaluations

AB Approve of appropriate behavior.
CR Criticize or point out error in nonverbal act.
DS Disapprove, scold, protest disruptive behavior.
ED Exclaim in disapproval.
ET Express enthusiasm for hearer's performance.
PM Praise for motor acts, i.e. for nonverbal behavior.

Demands for clarification

RR Request to repeat utterance.

Text editing

CT Correct, provide correct verbal form in place of erroneous one.

Vocalizations

YY Make a word-like utterance without clear function.
OO Unintelligible vocalization.

Certain other speech act codes that have been widely used in child language research can be encountered in the CHILDES database. These general codes should not be combined with the more detailed INCA-A codes. They include ELAB (Elaboration), EVAL (Evaluation), IMIT (Imitation), NR (No Response), Q (Question), REP (Repetition), N (Negation), and YN (Yes/No Question.

18 Error Coding

18.1 Word level error codes

Errors at the word level are marked by placing the [*] symbol after the erroneous word. If there is a replacement string, such as [: because], that should come before the error code. When an error occurs in the initial part of a retracing, the [*] symbol is placed after the error, but before the [/] mark.

18.1.1 Phonological errors [* p]

p:w	word, as in *boater* for *butter*
p:n	non-word, as in *buther* for *butter*
p:m	metathesis, as in *stisserz* for *sisters*

To be considered a phonological error, the error must meet these criteria:

1. For one-syllable words, consisting of an onset (initial phoneme or phonemes) plus vowel nucleus plus coda (final phoneme or phonemes), the error must match on 2 out of 3 of those elements (e.g., onset plus vowel nucleus OR vowel nucleus plus coda OR onset plus coda). The part of the syllable that is in error may be a substitution, addition, or omission. For one-syllable words with no onset (e.g., eat) or no coda (e.g., pay), the absence of the onset or coda in the error would also count as a match.
2. For multi-syllabic words, the error must have complete syllable matches on all but one syllable, and the syllable with the error must meet the one-syllable word match criteria stated above.

Note: If using other criteria for phonological error coding (e.g., overlap of ≥ 50% of phonemes between error production and target word), some of the n:k and s:ur errors may qualify.

18.1.2 Semantic errors [* s]

s:r	related word, target known, as in *mother* for *father*
s:ur	unrelated word, target known, as in *comb* for *umbrella*
s:uk	word, unknown target, as in "I go *wolf*"
s:per	perseveration, as in "he kicked the ball through the *ball*"

For errors with related words for known targets, one can add these additional distinctions:

s:r:prep	wrong preposition, as in *on* for *in* or *off* for *out*
s:r:seg	word that is a partial segment of the target, as in *fire* for *fireman*
s:r:der	derivational error using a real word, as in *assess* for *assessment* or *humbleness* for *humility*

Errors involving grammatical categories, such as number, case, definiteness, or gender are coded as [* s:r:gc]. These can be further coded using the relevant part of speech, such as "art" for article or "pro" for pronoun, and "der" for derivation, as in these examples:

s:r:gc:art	definite for indefinite, indefinite for definite, definite for zero

s:r:gc:pro	*his* for *her*, *your* for *yours*, *my* for *mine*

18.1.3 Neologisms [* n]

n:k	neologism, known target, does not meet phonological error criteria
n:uk	neologism, unknown target
n:k:s	neologism, known target, stereotypy (recurring non-word)
n:uk:s	neologism, unknown target, stereotypy (recurring non-word)
n:k:der	neologism, known target, as in *integrativity* for *integration,* or *foundament* for *foundation*

18.1.4 Morphological errors [* m:a]

For the coding of morphological errors, these nine abbreviations can be used:

-ing	progressive
-3s	3rd person singular
-ed	past
-en	perfective
-s	noun plural
-'s	possessive
-s'	possessive plural
-er	comparative
-est	superlative

Missing regular forms, in which the base lemma appears with no suffix, are coded with m:0, as in

m:0ing	missing progressive suffix
m:03s	*missing 3rd person singular suffix
m:0ed	missing regular past suffix
m:0s	*missing regular plural suffix
m:0's	missing possessive suffix
m:0s'	missing possessive plural suffix

However, the two codes marked above with * should usually be coded instead as agreement errors, as noted below.

Substitutions of the base form for irregulars, with omission of the expected marking, are coded with m:base:*, as in

m:base:s	*child* for *children*, *ox* for *oxen*
m:base:ed	*come* for *came*, *bring* for *brought*
m:base:en	*take* for *taken* or *freeze* for *frozen*
m:base:er	*badder* for *worse*
m:base:est	*baddest* for *worst*

Substitutions of an irregular for the base form are coded with m:irr:* in this way:

m:irr:s	*children* for *child*
m:irr:ed	*found* for *find*
m:irr:en	*taken* for *take*

Substitutions between past and perfective irregulars are coded with m:sub:* in this way:

m:sub:ed	*frozen* for *froze, seen* for *saw*
m:sub:en	*froze* for *frozen, saw* for *seen*

Overregularizations of irregulars are coded in this way:

m:=ed	overregularized -ed past, as in *seed* for *saw*
m:=en	overregularized -en perfective, as in *taked* for *taken*
m:=s	overregularized –s plural, as in *childs* for *children*

Superfluous markings of regulars are coded in this way:

m:+ing	superfluous progressive, as in *running* for *run*
m:+3s	*superfluous 3rd person singular –s suffix, as in *goes* for *go*
m:+ed	superfluous regular past, as in *walked* for *walk*
m:+en	superfluous perfective, as in *taken* for *take*
m:+s	*superfluous plural, as in *gowns* for *gown*
m:+'s	superfluous possessive or plural possessive, as in *John's* for *John.*

However, the two codes marked above with * should usually be coded as agreement errors, as noted below.

Double markings of regulars and irregulars are coded in this way:

m:++ing	*runninging*
m:++3s	*wantses*
m:++ed	*talkeded*
m:++en	*changeded*
m:++s	*sevenses*
m:++'s	*boys's's*

Double markings of irregulars are marked using the above codes plus a final :i

m:++ed:i	*tooked*
m:++en:i	*brokened, takenen*
m:++s:i	*feets*

Agreement errors for irregulars are marked in this way:

m:vsg:a	verb 3rd singular for unmarked:
	has for *have, is* for *are, was* for *were*
m:vun:a	verb unmarked for 3rd singular
	have for *has, are* for *is*, *were* for *was*

When agreement errors involve regulars, the :a should be added to the basic code, as in

m:03s:a	*he want* for *he wants*
m:+3s:a	*we wants* for *we want*
m:0s:a	noun singular for plural, as in *two dog* for *two dogs*
m:+s:a	noun plural for singular, as in *this dogs* for *this dog*

Allomorphy errors in the stem or base are coded in this way:

m:allo	*knifes* for *knives, an* for *a*

18.1.5 Dysfluencies [* d]

d:sw dysfluency within word, as in *insuhside* for *inside*

18.1.6 Missing Words

Missing words or parts of speech are coded with an initial "0", as in **0det, 0aux, 0does, etc.** For agrammatic and jargon aphasic speech, it is often best to avoid trying to guess at what is missing and to just mark incomplete utterances with the postcode [+ gram] (explained below).

18.1.7 General Considerations

- If the error is a non-word and the target is known, the target is marked with the square brackets and a single colon. Doing this allows the MOR program to use the real word target for parsing in its analysis. When the error is a real word, even though it is the wrong real word, the target can be marked with a single colon or a double colon. The double colon allows the MOR program to use the actual word produced rather than the target in its analysis, but also allows programs such as FREQ to use either form when needed.

  PAR: they cut the &+l lock off the door and tall [: call] [p:w] the paramedics .
  OR
  PAR: they cut the &+l lock off the door and tall [:: call] [p:w] the paramedics .

- Multiple codes may be used if an error is, for example, both a semantic and phonemic paraphasia, as in this example:

  PAR: it was singing [: ringing] [s:r] [* p:w] in my ears .

- Also, if the error is repeated, you can add "-rep" to the error code; if the error is revised (to another error or the correct word), add "-ret" (retraced) to the error code, as in this example:

  PAR: it's a little dog [: cat] [s:r-ret] [//] cat .

18.2 Utterance level error coding (post-codes)

In addition to providing methods for coding errors at the word level, CHAT includes a series of codes for errors involving larger segments of an utterance or the whole utterance. Mutliple codes can be used for a given utterance. The codes are:

[+ gram]	grammatical error	**[+ per]**	perseveration
[+ jar]	jargon	**[+ cir]**	circumlocution
[+ es]	empty speech		

Grammatical error – [+ gram] – includes agrammatic and paragrammatic utterances:

- telegraphic speech
- speech in which content words (mainly nouns, verbs, and adjectives) are relatively preserved but many function words (articles, prepositions, conjunctions) are missing (adapted from Brookshire, 1997)
- utterances with frank grammatical errors (without requiring that each utterance be a complete sentence with a subject and predicate)
- utterances with errors in word order, syntactic structure, or grammatical morphology (Butterworth and Howard, 1987)
- utterance level grammatical errors as opposed to word level agreement errors or missing parts of speech

*PAR: one two bread . [+ gram]
*PAR: whatever I'm think up . [+ gram]
*PAR: is getting want to be wasn't . [+ gram] [+ jar]
*PAR: when everything that we going out now . [+ gram] [+ jar]

Jargon – [+ jar] – mostly fluent and prosodically correct but largely meaningless speech, containing paraphasias, neologisms, or unintelligible strings; resembles English syntax and inflection (adapted from Kertesz, 2007)

PAR: go and &+ha hack [s:uk] the gets [* s:uk] be able gable [* s:uk] get &+su sɪm@u [: x@n] [* n:uk] . [+ jar]
PAR: get this care [s:uk-ret] [//] kɛɪf@u [: x@n] [* n:uk] to eat here . [+ jar]
*PAR: and xxx . [+ jar]

Empty speech – [+ es] – speech that is syntactically correct but conveys little or no overall meaning, often a result of substituting general words (e.g., thing, stuff) for more specific words (Brookshire, 1997). Differentiating among "empty speech", "jargon", and "grammatical error" codes may be challenging. In truth, all these sentences may be meaningless in the conversational context. Briefly, empty speech utterances should contain general, vague, unspecific referents; jargon utterances should contain paraphasias and/or neologisms; and paragrammatic utterances (in the grammatical error category) should have inappropriate juxtapositions of grammatical elements.

*PAR: we got little things over here . [+ es]
*PAR: there was nothing in that one there . [+ es]

Perseveration – [+ per] – repetition of an utterance when it is no longer appropriate (Brookshire, 1997)

Circumlocution – [+ cir] – talking around words/concepts

*PAR: and through the help of <the whatever fairy or whoever the [x 3] what> [//] the lady that is helping Cinderella &-um she has <the chance to check the> [//] the prince check the &+s &-uh shoe . [+ cir] [+ gram]

References

Allen, G. D. (1988). The PHONASCII system. *Journal of the International Phonetic Association, 18*, 9-25.

Ament, W. (1899). *Die Entwicklung von Sprechen und Denken beim Kinder*. Leipzig: Ernst Wunderlich.

Augustine, S. (1952). *The Confessions, original 397 A. D.* (Vol. Volume 18). Chicago: Encyclopedia Britannica.

Bates, E., & MacWhinney, B. (1982). Functionalist approaches to grammar. In E. Wanner & L. Gleitman (Eds.), *Language acquisition: The state of the art* (pp. 173-218). New York, NY: Cambridge University Press.

Bernstein Ratner, N., Rooney, B., & MacWhinney, B. (1996). Analysis of stuttering using CHILDES and CLAN. *Clinical Linguistics and Phonetics, 10*(3), 169-188.

Bloom, P. (2000). *How children learn the meanings of words*. Cambridge, MA: MIT Press.

Brown, R. (1973). *A first language: The early stages*. Cambridge, MA: Harvard.

Chafe, W. (Ed.) (1980). *The Pear stories: Cognitive, cultural, and linguistic aspects of narrative production*. Norwood, NJ: Ablex.

Clark, E. (1987). The Principle of Contrast: A constraint on language acquisition. In B. MacWhinney (Ed.), *Mechanisms of Language Acquisition* (pp. 1-34). Hillsdale, NJ: Lawrence Erlbaum Associates.

Crystal, D. (1969). *Prosodic systems and intonation in English*. Cambridge: Cambridge University Press.

Crystal, D. (1979). Prosodic development. In P. Fletcher & M. Garman (Eds.), *Language acquisition: Studies in first language development*. New York, NY: Cambridge University Press.

Darwin, C. (1877). A biographical sketch of an infant. *Mind, 2*, 292-294.

Edwards, J. (1992). Computer methods in child language research: four principles for the use of archived data. *Journal of Child Language, 19*, 435-458.

Ekman, P., & Friesen, W. (1969). The repertoire of nonverbal behavior: Categories, origins, usage, and coding. *Semiotica, 1*, 47-98.

Ekman, P., & Friesen, W. (1978). *Facial action coding system: Investigator's guide*. Palo Alto, CA: Consulting Psychologists Press.

Fletcher, P. (1985). *A child's learning of English*. Oxford: Blackwell.

Goldman-Eisler, F. (1968). *Psycholinguistics: Experiments in spontaneous speech*. New York, NY: Academic Press.

Gvozdev, A. N. (1949). *Formirovaniye u rebenka grammaticheskogo stroya*. Moscow: Akademija Pedagogika Nauk RSFSR.

Halliday, M. (1966). Notes on transitivity and theme in English: Part 1. *Journal of linguistics, 2*, 37-71.

Halliday, M. (1967). Notes on transitivity and theme in English: Part 2. *Journal of linguistics, 3*, 177-274.

Halliday, M. (1968). Notes on transitivity and theme in English: Part 3. *Journal of linguistics, 4*, 153-308.

Jefferson, G. (1984). Transcript notation. In J. Atkinson & J. Heritage (Eds.), *Structures of social interaction: Studies in conversation analysis* (pp. 134-162). Cambridge: Cambridge University Press.

Kearney, G., & McKenzie, S. (1993). Machine interpretation of emotion: Design of memory-based expert system for interpreting facial expressions in terms of signaled emotions. *Cognitive Science, 17*, 589-622.

Kenyeres, E. (1926). *A gyermek elsö szavai es a szófajók föllépése*. Budapest: Kisdednevelés.

Kenyeres, E. (1938). Comment une petite hongroise de sept ans apprend le français. *Archives de Psychologie, 26*, 521-566.

Leopold, W. (1939). *Speech development of a bilingual child: a linguist's record: Vol. 1. Vocabulary growth in the first two years* (Vol. 1). Evanston, IL: Northwestern University Press.

Leopold, W. (1947). *Speech development of a bilingual child: a linguist's record: Vol. 2. Sound-learning in the first two years*. Evanston, IL: Northwestern University Press.

Leopold, W. (1949a). *Speech development of a bilingual child: a linguist's record: Vol. 3. Grammar and general problems in the first two years*. Evanston, IL: Northwestern University Press.

Leopold, W. (1949b). *Speech development of a bilingual child: a linguist's record: Vol. 4. Diary from age 2*. Evanston, IL: Northwestern University Press.

LIPPS. (2000). The LIDES manual: A document for preparing and analysing language interaction data. *International Journal of Bilingualism, 4*, 1-64.

Low, A. A. (1931). A case of agrammatism in the English language. *Archives of Neurology and Psychiatry, 25*, 556-597.

MacWhinney, B. (1989). Competition and lexical categorization. In R. Corrigan, F. Eckman, & M. Noonan (Eds.), *Linguistic categorization* (pp. 195-242). Philadelphia, PA: Benjamins.

MacWhinney, B., & Osser, H. (1977). Verbal planning functions in children's speech. *Child Development, 48*, 978-985.

Malvern, D., Richards, B., Chipere, N., & Purán, P. (2004). *Lexical diversity and language development*. New York, NY: Palgrave Macmillan.

Miller, J., & Chapman, R. (1983). *SALT: Systematic Analysis of Language Transcripts, User's Manual*. Madison, WI: University of Wisconsin Press.

Moerk, E. (1983). *The mother of Eve as a first language teacher*. Norwood, N.J.: ABLEX.

Ninio, A., Snow, C. E., Pan, B., & Rollins, P. (1994). Classifying communicative acts in children's interactions. *Journal of Communication Disorders, 27*, 157-188.

Ninio, A., & Wheeler, P. (1986). A manual for classifying verbal communicative acts in mother-infant interaction. *Transcript Analysis, 3*, 1-83.

Ochs, E. (1979). Transcription as theory. In E. Ochs & B. Schieffelin (Eds.), *Developmental pragmatics* (pp. 43-72). New York, NY: Academic.

Ochs, E. A., Schegloff, M., & Thompson, S. A. (1996). *Interaction and grammar*. Cambridge: Cambridge University Press.

Parisse, C., & Le Normand, M.-T. (2000). Automatic disambiguation of the morphosyntax in spoken language corpora. *Behavior Research Methods, Instruments, and Computers, 32*, 468-481.

Parrish, M. (1996). Alan Lomax: Documenting folk music of the world. *Sing Out!: The Folk Song Magazine, 40*, 30-39.

Pick, A. (1913). *Die agrammatischer Sprachstörungen*. Berlin: Springer-Verlag.

Preyer, W. (1882). *Die Seele des Kindes*. Leipzig: Grieben's.

Sacks, H., Schegloff, E., & Jefferson, G. (1974). A simplest systematics for the organization of turn-taking for conversation. *Language, 50*, 696-735.

Sagae, K., Davis, E., Lavie, A., MacWhinney, B., & Wintner, S. (2007). High-accuracy annotation and parsing of CHILDES transcripts. In *Proceedings of the 45th Meeting of the Association for Computational Linguistics* (pp. 1044-1050). Prague: ACL.

Sagae, K., Lavie, A., & MacWhinney, B. (2005). Automatic measurement of syntactic development in child language. In *Proceedings of the 43rd Meeting of the Association for Computational Linguistics* (pp. 197-204). Ann Arbor, MI: ACL.

Sagae, K., MacWhinney, B., & Lavie, A. (2004). Adding syntactic annotations to transcripts of parent-child dialogs. In *LREC 2004* (pp. 1815-1818). Lisbon: LREC.

Selting, M., & al., e. (1998). Gesprächsanalytisches Transkriptionssystem (GAT). *Linguistische Berichte, 173*, 91-122.

Slobin, D. (1977). Language change in childhood and in history. In J. Macnamara (Ed.), *Language learning and thought* (pp. 185-214). New York, NY: Academic Press.

Sokolov, J. L., & Snow, C. (Eds.). (1994). *Handbook of Research in Language Development using CHILDES*. Hillsdale, NJ: Erlbaum.

Stemberger, J. (1985). *The lexicon in a model of language production*. New York, NY: Garland.

Stern, C., & Stern, W. (1907). *Die Kindersprache*. Leipzig: Barth.

Trager, G. (1958). Paralanguage: A first approximation. *Studies in Linguistics, 13*, 1-12.

Wernicke, C. (1874). *Die Aphasische Symptomenkomplex*. Breslau: Cohn & Weigart.