The CHILDES Project

Tools for Analyzing Talk – Electronic Edition

Part 1: The CHAT Transcription Format

Brian MacWhinney
Carnegie Mellon University

June 19, 2015

Citation for last printed version:

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

1 Table of Contents

2 Introduction

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 (P. 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. For further ideas regarding this type of work, you may wish to connect to http://talkbank.org where there are various extensions of the CHILDES project.

2.6 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:

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

The CHILDES system 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 CHILDES programs to enter new data sets. Eventually, these new data sets will be available to other researchers as a part of the growing CHILDES database. In this way, CHAT, CLAN, and the database function as a coarticulated set of complementary tools.

There are manuals for each of the three CHILDES tools. The CHAT manual, which you are now reading, describes the conventions and principles of CHAT transcription. The CLAN manual describes the use of the CLAN computer programs that you can use to transcribe, annotate, and analyze language interactions. The third manual, which is actually a collection of over a dozen separate manuals retrievable from a single link on the web, describes the data files in the CHILDES database. Each of these database manuals describes the data sets in one major component of the database. In addition, there is a short manual that provides an overview for the entire database.

2.7 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 (Bernstein-Ratner, Rooney, & MacWhinney, 1996). The exercises in Chapter 7 of Part II 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. Ongoing work in TalkBank is documented on the web at http://talkbank.org.

2.8 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, Mike Forrester, and Chris Ramsden 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.

2.9 Constructing the Database

The primary reason for the success of the CHILDES database has been the generosity of over 100 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, 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.

2.10 Disseminating CHILDES

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.

2.11 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 to 2010, the 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 which seeks to improve the CHILDES tools and to use CHILDES as a model for other disciplines studying human communication. 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.

2.12 How to Use These Manuals

Each of the three parts of the CHILDES system is described in a separate 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 CHILDES 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. 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 the CHILDES archive, 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. At the end of the CLAN manual, there are also a series of exercises that help students further consolidate their knowledge of CHAT and CLAN.

The CHILDES 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 CHILDES 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.

There are many researchers in the fields that study language learning who will never need to use CHILDES. Indeed, we estimate that the three CHILDES tools will never be used by at least half of the researchers in the field of child language. There are three common reasons why individual researchers may not find CHILDES useful:

1. some researchers may have already committed themselves to use of another analytic system;
2. some researchers may have collected so much data that they can work for many years without needing to collect more data and without comparing their own data with other researchers' data; and
3. some researchers may not be interested in studying spontaneous speech data.

Of these three reasons for not needing to use the three CHILDES tools, the third is the most frequent. For example, researchers studying comprehension would only be interested in CHILDES data when they wish to compare findings arising from studies of comprehension with patterns occurring in spontaneous production.

2.13 Changes

The CHILDES 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 CHILDES 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 CHILDES Project is active.

3 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 CHILDES (Child Language Data Exchange System) Project. All of the transcripts in the CHILDES database 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 CHILDES system has had a major impact on the study of child language. At the time of the last monitoring in 2003, there were over 2000 published articles that had made use of the programs and database. In 2007, the size of the database had grown to over 44 million words, making it by far the largest database of conversational interactions available anywhere. The total number of researchers who have joined as CHILDES members across the length of the project is now over 4500. 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, approximately 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.

3.1 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.

3.2 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 hand-written transcript. Let us look at some of the dangers surrounding the enterprise of transcription.

3.2.1 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].

3.2.2 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.

3.2.3 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.

3.3 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.

3.4 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.

3.5 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, unlike a human language, a transcription system needs to address two different audiences. One audience is the human audience of transcribers, analysts, and readers. The other audience is the digital computer and its programs. In order to successfully deal with these two audiences, a system for computerized transcription needs to achieve the following goals:

1. **Clarity:** Every symbol used in the coding system should have some clear and definable real-world referent. The relation between the referent and the symbol should be consistent and reliable. Symbols that mark particular words should always be spelled in a consistent manner. Symbols that mark particular conversational patterns should refer to actual patterns consistently observable in the data. In practice, codes will always have to 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. Systematicity is a simple extension of clarity across transcripts or corpora. 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.
2. **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 CHILDES 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.
3. **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.

4 CHAT Outline

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 midCHAT.

4.1 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.

4.2 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 should end with an utterance terminator. The basic utterance terminators are the period, the exclamation mark, and the question mark.
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. Similarly,

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. 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
```

4.3 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.

4.4 midCHAT

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 the next levels, which are midCHAT and midCLAN. Learning midCHAT involves mastering the transcription of words and conversational features. In particular, the midCHAT learner should work through the chapters on words, utterances, and scoped symbols. Depending on the shape of the particular project, the transcriber 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 eventuallly read all of the current manual, although some sections that seem less relevant to the project can be skimmed.

4.5 File Naming

The CHILDES 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 byindividual children or groups of children. For longitudinal corpora, we recommend that file names use the age of the child and an index number. For example, the transcript from the fourth taping session when the child was 2;3;22 would be called 04-20322.cha. It is better to use ages for file names, rather than dates or other material.

4.6 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.

1. **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 CHILDES system in general.
2. **Restrictions.** If the data are being contributed to the CHILDES system, contributors can set particular restrictions on the use of their data. For example, re-

searchers 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.

3. **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.
4. **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.
5. **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?
6. **Codes.** If there are project-specific codes, these should be described.
7. **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.
8. **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.

The various readme files for the corpora that are now in the CHILDES database were all contributed in this form. To maintain consistency and promote an overview of the database, these files were then edited and reformatted and combined into the database files that can now be downloaded from the server.

4.7 Checking Syntactic Accuracy

Each CLAN command runs a very superficial check to see if a file conforms to minCHAT. 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.

5 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 movie
@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.

5.1 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 three hidden headers that appear before this header. These are the @Font header, the @UTF8 header, the @PID header, and the @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 Handle System (www.handle.net) that allows for persistent identification of the location of digital objects. These values can be entered into any system that resolves PIDs to locate the required resource, such as the server at http://128.2.71.222:8000.

@ColorWords

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

5.2 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:     sv, pt
```

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:

Table 1: ISO Language Codes

Language	Code	Language	Code	Language	Code
Afrikaans	afr	German	deu	Polish	pol
Arabic	ara	Greek	ell		
Basque	eus	Hebrew	heb	Portuguese	por
Cantonese	zho-yue	Hungarian	hun	Punjabi	pan
Catalan	cat	Icelandic	isl	Romanian	ron
Chinese	zho	Indonesian	ind	Russian	rus
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. 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.

The @s 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 $ 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.

Tone languages like Cantonese, Mandarin, and Thai are allowed to have word forms that include tones and numbers for polysemes.

@Participants:

This is the third visible header. Like the @Begin and @Participants headers, it is obligatory. It lists all of the actors 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:

1. **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, Ioni, and Ferfulice.

2. **Name.** The speaker's name can be omitted. 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.”

3. **Role.** After the ID and name, you type in the role of the speaker. There are a fixed set of roles specified in the depfile.cut file used by CHECK and we recommend trying to use these fixed roles whenever possible. Please consult that file for the full list. 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 roles, like Adult, Child, or Teenager. Then, the exact nature of the role can be put in the place of the name, as in these examples:

```
@Participants:  TBO Toll_Booth_Operator Adult,AIR Airport_Attendant
    Adult, SI1 First_Sibling Sibling, SI2 Second_Sibling Sibling, OFF
    MOT_to_INV OffScript, NON Computer_Talk Non_Human
```

@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.

1. CA. Use of this option suspends the usual requirement for utterance terminators.
2. 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.
3. Sign. Use of this option permits the use of all capitals in words for Sign Language notation.
4. IPA. Use of this option permits the use of IPA notation on the main line.
5. Line. Use of this option tells the web browser to expect time marking bullets on each line. By default, the browser expects a bullet at the end of each tier.
6. Multi. Use of this option tells the checkers 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.
7. Caps. This option turns off CLAN's restriction against having capital letters inside words.
8. Bullets. This option turns off the requirement that each time-marking bullet should begin after the previous one.

@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:

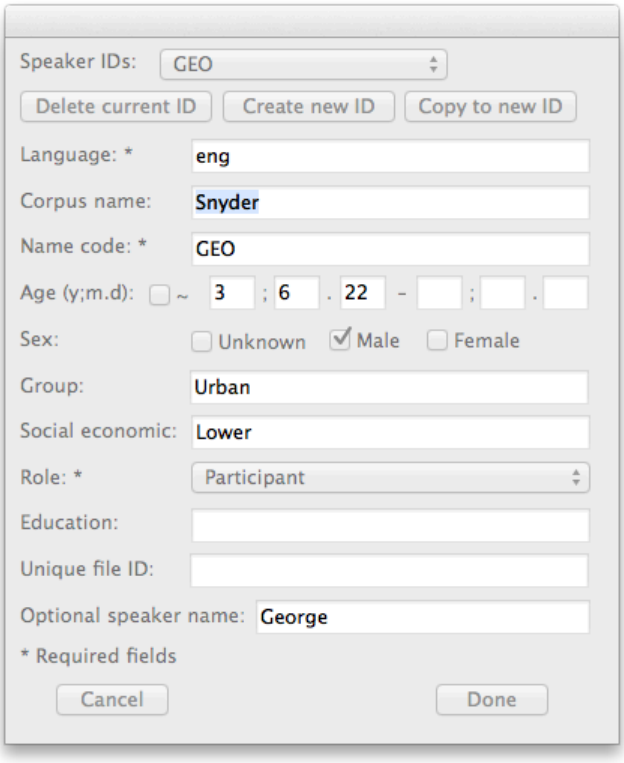
```
@ID: language|corpus|code|age|sex|group|SES|role|education|custom|
```

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 characterizations of the possible fields for the @ID header.

Language:	as in Table 1 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
SES:	Ethnicity (White, Black, Latino, Asian, Pacific), SES (WC, UC, MC, LI)
Role:	the role as given in the @Participants line
Education:	educational level of the speaker
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. If you do not know the child's age in days, you can simply use years and months, as in 6;4. 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 the terms "missing" or "unlinked" after the media type. So the line has this shape:

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

@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.

5.3 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 #:

5.4 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.

@Exceptions:

This allows for special word forms in certain corpora.

@Interaction Type:

The possible entries here include: constructed computer phonecall telechat meeting work medical classroom tutorial private family sports religious legal face_to_face

@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: two three four five more audience

@Recording Quality:

Possible entries here are: poor, fair, good, and excellent.

@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 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 is provided by coding on the %tim dependent tier in conjunction with the @Time Start header described next.

@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.

```
@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.

5.5 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.

@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.

@New Language:

This header is used to indicate the shift from the initially most frequent language listed in the @Languages header to a new most frequent language. This header should only be used when there is a marked break in a transcript from the use of one language to a fairly uniform use of another language.

@Page:

This header is used to indicate 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?

6 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:

1. special learner-form markers,
2. codes for unidentifiable material,
3. codes for incomplete words,
4. ways of treating formulaic use of words, and
5. 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.

6.1 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. Other non-letter characters such as the plus sign (+) or the at sign (@) can be used within words to express special meanings. 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.

6.2 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.

6.3 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.

Table 2: 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
@f	family-specific form	bunko@f	broken	fam
@g	general special form	gongga@g	-	-
@i	interjection, interaction	uhhuh@i	-	int
@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
@pm	protomorpheme	wi@pm	will?	pm
@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	small creature	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:

1. **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.
2. **Babbling** can be used to mark both low-level early babbling and high-level sound play in older children. These forms have no obvious meaning and are used just to have fun with sound.
3. **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.

4. **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.
5. **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 *under-toad* to refer to some mysterious being in the surf, although the word was simply *undertow* initially.
6. **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.
7. **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.
8. **Letters** can either be transcribed with the @l marker or simply as single-character words. Strings of letters are marked as @k.
9. **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.
10. **Nonvoiced forms** are produced typically by hearing-impaired children or their parents who are mouthing words without making their sounds.
11. **Onomatopoeias** include animal sounds and attempts to imitate natural sounds.
12. **Phonological consistent forms** (PCFs) are early forms that are phonologically consistent, but whose meaning is unclear to the transcriber. Usually these forms have some relation to small function words.
13. **Protomorphemes** are forms that will eventually become morphemes, including function words and affixes.
14. **Metalinguistic reference** can be used to either cite or "quote" single standard words or special child forms.
15. **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.
16. **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.
17. **Sign language** use can be indicated by the @sl.
18. **Sign and speech** use involves making a sign or informal sign in parallel with saying the word.

19. **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.
20. **Test words** are nonce forms generated by the investigators to test the productivity of the child's grammar.
21. **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.
22. **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.
23. **Unknown** forms can be marked with @x. However, usually unknown forms are transcribed using the xxx, yyy, and www markers.
24. **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.

Later in this chapter we present a set of standard spellings of English words that make use of @d, @fp, and @i largely unnecessary. However, in languages where such a list is not available, it may be necessary to use forms with @d or @i. 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, @u, or @w 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 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.

If transcribers find it difficult to distinguish between child-invented forms, onomatopoeia, and familial forms, they can use the @ symbol without any following letter. In this way, they can at least indicate the fact that the preceding word is not a standard item in the adult lexicon.

6.4 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.

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 Fragment &

The & symbol can be used at the beginning of a string to indicate that the following material is just a phonological fragment or piece of a word and that CLAN should not treat it as a word. It is important not to include any of the three utterance terminators – the exclamation mark, the question mark, or the period – because CLAN will treat these as utterance terminators. This form of notation is useful when the speaker stutters or breaks off before completing a recognizable word (false starts). The utterance "t- t- c-can't you go" is transcribed as follows:

```
*MAR:     &t &t &k can't you go?
```

The ampersand can also be used for nonce and nonsense forms:

```
*DAN:     &glnk &glnk.
%com:     weird noises
```

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 to have the material treated as a word, use the @u form of notation instead (see the previous section).

Unless you specifically attempt to search for strings with the ampersand, the CLAN commands will not see them at all. 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"&*".

6.5 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 an extremely 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. If what is important is not the actual word omitted, but its 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. This item would be counted for scoping conventions, but it would not be included in the MLU count. 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. 0art: Unless there is a missing plural, a common noun without an article is coded as 0art.
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.
5. 0pobj: Every preposition requires an object. However, often a preposition may be functioning as an adverb. The coder must look at the verb to decide whether a word is functioning as a preposition as in "John put on 0pobj" or an adverb as in "Mary jumped up."

In English, there seldom are solid grounds for assigning codes like 0adj, 0adv, 0obj, 0prep, or 0dat.

6.6 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. This section lists some of these words with their standard orthographic form.

The forms in these lists all have some conventional lexical status in standard American English. In this regard, they differ from the various nonstandard forms indicated by the special form markers @b, @c, @f, @l, @n, @o, @p, and @s. Because there is no clear limit to the number of possible babbling forms, onomatopoeic forms, or neologistic forms, there is no way to provide a list of such forms. In contrast, the words given in this section are fairly well known to most speakers of the language, and many can be found in unabridged dictionaries. The list given here is only a beginning; over time, we intend to continue to add new forms.

Some of the forms use parentheses to indicate optional material. For example, the exclamation *yeek* can also be said as *eek*. When a speaker uses the full form, the transcriber types in *yeek*, and when the speaker uses the reduced form the transcriber types *(y)eek*. When clan analyzes the transcripts, the parentheses can be ignored and both *yeek* and *eek* will be retrieved as instances of the same word. Parentheses can also be used to indicate missing fragments of suffixes. The majority of the words listed can be found in the form given in *Webster's Third New International Dictionary*. Those forms that cannot be found in *Webster's Third* are indicated with an asterisk. The asterisk should not be used in actual transcription.

6.6.1 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 cover better be coded as:

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

6.6.2 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.

Earlier, 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.

The other 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, appelations 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*.

Because the dash is used on the %mor line to indicate suffixation, it is important to avoid confusion between the standard use of the dash in compounds such as “blue-green” and the use of the dash in CHAT. To do this, use the compound marker to replace the dash or hyphen, as in *blue+green* instead of *blue-green*.

6.6.3 Capitalization

In general, capitals are only allowed at the beginnings of words. However, they can also occur later in a word in these cases:

1. if the @Options tier includes: "sign" or "CA".
2. If @u is used at the end of the word.
3. After the + symbol.
4. After the _ underscore symbol.
5. The word is listed on the @Exceptions tier.
6. The capital letter is preceded by prefix, like "Mac", which is specified in the depfile in [UPREFS Mac] code.

6.6.4 Acronyms

Acronyms should be transcribed by using the component letters as a part of a “linked” form. In compounds, the @l marking is not used, since it would make the acronym unreadable. Thus, USA is be written as *U_S_A*. In this case, the first letter is capitalized in order to mark it as a proper noun. Other examples include M_I_T, C_M_U, M_T_V, E_T, I_U, C_three_P_O, R_two_D_two, and K_Mart. 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.

6.6.5 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 ay m” 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.”

6.6.6 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*.

Table 2: 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		

6.6.7 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.

Table 3: 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.

6.6.8 Assimilations

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. For forms of this type, CHAT allows the transcriber to place the assimilated form on the main line followed by a fuller form in square brackets, as in the form:

```
gonna [: going to]
```

CLAN allows the user to either analyze the material preceding the brackets or the material following the brackets, as described in the section of the chapter on options that discusses the +r switch. An extremely incomplete list of assimilated forms is given below. None of these forms can be found in *Webster's Third New International Dictionary*.

Table 4: Assimilations

Nonstandard	Standard	Nonstandard	Standard
coulda(ve)	could have	mighta	might have
dunno	don't know	need(t)a	need to
dyou	do you	oughta	ought to
gimme	give me	posta	supposed to
gonna	going to	shoulda(ve)	should have

gotta	got to	sorta	sort of
hadta	had to	sorta	sort of
hasta	has to	wanna	want to
hafta	have to	wassup	what's up
kinda	kind of	whaddya	what did you
lemme	let me	whyntcha	why didn't you
lotsa	lots of		

If you transcribe these forms as single morphemes, they will be counted as single morphemes and programs like MOR will recognize them as wholes. If you do not believe that they are morphemic units, you can break them up into components in two ways. First, forms involving alterations of *you* can be represented by having *ya*, *chu*, and *cha* as alternative spellings for *you*. Second, you can analyze these forms using the replacement notation. Thus, transcribers can choose to either enter "could cha" or "couldcha [: could you]". If you have chosen to represent *you* as *ya*, you must remember to include *ya* in your search lists. Another 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." However, this method is not good, if you are convinced that these forms are monomorphemic.

6.6.9 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 great these all as communicators. Because their phonological shape varies so much, these forms often have an unclear lexical status. The following table 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 MOR program uses a standard set of forms in its co.cut, co-rhymes.cut, co-under.cut, and co-voc.cut files that you will need to consult.

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

6.6.10 Spelling Variants

There are a number of words that are misspelled so frequently that the misspellings seem as acceptable as the standard spellings. These include *altho* for standard *although*, *donut* for *doughnut*, *tho* for *though*, *thru* for *through*, and *abc's* for *abcs*. Transcribers should use the standard spellings for these words. 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*.

6.6.11 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.*

Table 7: 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

6.6.12 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. There are four ways to deal with this problem. The first is to add each variant form to the 0lexicon.cdc file, which also contains other nonstandard forms. Because these variant forms are, in nearly all cases, nonhomographic with other words, researchers analyzing the transcript will simply need to include the variant in their search lists. An example of an exception to this is *den* for *then*, because *den* is already the standard word for an animal's burrow. A second solution to this problem is to follow each variant form with the standard form, as given below using the [: replacement] notation. A third 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. The fourth 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*.

Table 8: 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		

6.6.13 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*.

Table 9: 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

6.6.14 Word separation in Japanese

Many analyses with CLAN relies on words as items. 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, which can be downloaded from the web at http://childes.psy.cmu.edu/morgrams/wakachi2002.zip 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).

6.6.15 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:

Table 11: 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.

7 Utterances

The basic units of CHAT transcription are the morpheme, the word, and the utterance. In addition, some transcribers may be interested in marking tone units. In the previous two chapters we examined principles for transcribing words and morphemes. In this chapter we examine ways of delimiting utterances and tone units.

7.1 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 fie 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. Consider the ways in which the five forms will lead to different results in the mlu command. The first three forms will all be counted as having one utterance with four morphemes for an MLU of 4.0. The fourth form 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.

Admittedly, 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 three 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, unless the transcriber explicitly declares that the data should not be used to compute syntactic and sentential measures.

The fourth form of transcription treats the successive productions of the word "milk" as repeated attempts to produce a single word. This form of transcription makes sense if there is clear evidence that the child was having trouble saying the word. If there is no evidence that the word is really a repetition, it would seem best to use the fifth form of transcription. Studies of early child syntax have emphasized the extent to which the child is subject to constraints on utterance length (L Bloom & Lahey, 1973; L. Bloom, Lightbown, & Hood, 1975; Gerken, 1991; Gerken, Landau, & Remez, 1990). However, if one decides to count all repetitions of single words as full productions, it would seem that one is overestimating the degree of syntactic integration being achieved by the child. On the other hand, some researchers have argued that treatment of words as separate utterances in the earliest stages of language acquisition tends to underestimate the level of syntactic control being achieved by the child (Branigan, 1979; Elbers & Wijnen, 1993).

CLAN provides a partial solution to this dilemma. In cases where the researcher wants to use separate utterances for each word, the commands will treat each utterance as having a single morpheme. If the fourth form of transcription with repetition marks is used, the commands will, by default, treat the utterance as having only one morpheme. However, there is an 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 measures.

The example we have been discussing involves a simple case of word repetition. In other cases, researchers may want to group together nonrepeated 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, 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.

A somewhat separate but related issue is the treatment of interactional markers and other "communicators" such as "yes," "sure," "well," and "now." In general, it seems best to group these markers together with the utterances to which they are most closely bound intonationally. However, it only makes sense to do this if the utterances are contiguous in discourse. Here are some examples:

```
*CHI:     no, Mommy no go.
*CHI:     no Mommy go.
*CHI:     no (.) Mommy go.
```

However, in other cases, it makes sense to transcribe "no" by itself:

```
*CHI:     no
*MOT:     why not?
*CHI:     Mommy go.
```

7.2 Discourse Repetition

In the previous section, 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.

7.3 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 line, 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?
```

7.4 Satellite Markers

There are two marks in CHAT for satellite material occurring at the beginning and ends of utterances that is somehow separate from the utterance. A special double comma Unicode symbol can be used to mark tag questions. This mark should be surrounded with spaces. It is entered using the F2+t combination.

```
*MOT:     you're coming home soon „ aren't you?
```

The other satellite marker can be used to separate off initial vocatives or topics in topic-comment languages. It is entered using the F2+v combination. Here is an example:

```
*MOT:     Mom ‡ you're coming home soon „ aren't you?
```

Both of these satellite markers should be surrounded by spaces, since they will be treated as separate word forms by MOR and GRASP.

7.5 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 **:**

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

Other

Transcribers should avoid using other separators, because most of them have special meanings in CHAT. For example, when attached directly to the end of a word, the dash

means incompletion. Therefore, it should not be used to mark appositives or other parenthetical material, as in standard written practice.

7.6 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 ↓↑ .

7.7 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 Chapter 10. 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, as in this example:

```
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 using to marked 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, Rooney, & MacWhinney, 1996). This form of word attack is marked by a caret or up arrow placed directly before the word.

7.8 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 words.

7.8.1 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, they can also take an object: imit, point, and move. 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	&=points:car
&=imit:plane	&=ges:squeeze	&=reads	&=points:nose
&=imit:lion	&=ges:come	&=walks:door	&=turns:page
&=imit:baby	&=shows:picture	&=runs:door	&=hits:table
&=ges:ignore	&=shows:scab	&=eats	&=pats:head
&=ges:unsure		&=drinks	

The object of the &=imit codes indicate 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 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.

7.8.2 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.

7.8.3 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 ?
```

7.8.4 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 Event **&{l=*** **intervening text** **&}l=***

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

Long Nonverbal 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 &{l=waving:hands and then continue until the end marked by &}l=waving:hands.

7.9 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 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 in this way 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 quote (“, Unicode 201C) and end quote (”, Unicode 201D) double-quote symbols can be used. These can be entered in the CLAN editor using F2-' and F2-" respectively. If there is a quote embedded inside a quote, then mark the internal quote using the single-quote begin (‘ Unicode 2018) and end (’ Unicode 2019) marks, which can be from the Special Characters window.

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 +".
```

7.10 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.
```

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.

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.
```

8 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.

8.1 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 digitized video file or audio file. If you use the escape-A command in the editor, all of this information is hidden and you see a single bullet. 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, 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 the interaction.

Text Bullet **·%txt: cat.jpg·**

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

8.2 Paralinguistic Scoping and Events

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.

8.3 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 morphemicized 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 “broth(er).”

In order for replacement to function properly, nothing should be placed between the replacing string and the string to be replaced. For example, 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 is 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].
```

Dependent Tier on Main Line [%xxx: text]

There are six dependent tiers that can be placed directly on the main line. They are %act, %add, %gpx, %int, %sit, and %spe. This placement of dependent tier information is useful when you wish to refer to a particular set of words, rather than the utterance as a whole, as in this example:

```
*RES:     would all of you <who have not had seconds>
   [%gpx: looks at Timmy] come up to the front of the line?
```

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> [?] ?
```

8.4 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."

Sometimes it is necessary to break up the standard flow of the interaction in order to code utterances on separate lines. For example, the following transcription is not legal in CHAT:

```
*SAR:     <I +/.
*EXP:     that's great.
*SAR:     +, bought a> [//] I just bought a helmet.
```

Instead, this passage should be transcribed as:

```
*SAR:     I [>] bought a [//] I just bought a helmet.
*EXP:     [<] <that's great>.
```

In the last analysis, researchers who want to capture overlaps in absolutely full detail should rely on the facilities of "sonic CHAT" that are described for the editor, rather than attempting to capture overlaps by using complex embeddings of pair delimiters.

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 retracing symbol, as in:

```
*HAR:     it's [/] (.) &um (.) it's [/] it's like (.) 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 retracing, as in this example:

```
*HAR:     <it's it's it's> [/] it's like (.) 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] like (.) 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 had 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 63 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.

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. 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 additonal user-defined single-letter 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.

Interposed Word &*

It is sometimes convenient to mark the interposition of a short comment word 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.
```

8.5 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.

8.6 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 two-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
```

9 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 lowercase 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 are 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.

9.1 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 [%act: kicks the ball].
```

And if one does not care about preserving the identification of the information as an action, the following form can be used:

```
*ADA: 0 [% kicks the 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. The first form preserves this best and the last form fails to preserve it at all. Actions also include gestures, such as nodding, pointing, waving, and shrugging.

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 do-'nt> [//] I do-'nt 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 there is a clan command that can enter it automatically by comparing the main line to the other coding lines. However, when dealing with very difficult speech such as that of Wernicke's aphasics (particularly in other languages), the transcriber may find it useful to first type in this line as a kind of notepad from which it is then possible to create the main line and the %err line.

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). For information on the GRASP parser for CHILDES data, see the materials at http://talkbank.org/grasp

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 describe phonological phenomena. 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. To do this easily, you need to use a keyboard entry system. Information on such systems is available from http://childes.psy.cmu.edu.

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 2039 and 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 23A8 ⎨ and the close by Unicode 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 for time stamp coding. It should not be confused with the millisecond accurate timing found in the bullets inserted by sonic CHAT. This 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.

```
*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 are the %mor line.

9.2 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?
*CHI:     sit in my lap.
%act:     $sc=4-7 $IMIT
```

10 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:

- 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.
- Utterances and inter-TCU pauses are numbered by the automatic line numbering function.
- 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.
- After the line number comes an asterisk and then the speaker ID code and a colon and a tab, as in CHAT format.
- Tabs are not used elsewhere.
- CA Overlaps, as marked with the special symbols ⌈ ⌉ ⌊ and⌋, are aligned automatically by the INDENT program, so hand indentation is not needed.
- To maintain proper alignment, CLAN uses a special fixed-width Unicode font.
- CHAT requires obligatory utterance terminators. However, CA uses terminal contours instead, as noted in the table below, and these are optional.
- 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.
- CHAT uses the following forms for marking disfluencies, as further discussed in the next chapter.
    - pairs of Unicode 21AB leftwards arrow with loop to mark initial segment repetition as in ↫b-b-b↫boy
    - pairs of Unicode 2260 not-equal-to sign to mark blocked segments as in ru≠b-b-b≠bber
    - the colon for marking drawls or extensions
    - the ^ symbol for marking a break inside a work

- forms such as &um for marking filled pauses
- silent pauses as marked by (.) or (0.6) etc.
- [/] string for word or phrase repetition
- string [//] string for retracing
- +… 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.
- %ges: dependent tiers such as %ges, %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 or laughter stretch symbol is ¥ which is inserted by F1 and then the letter l. After row 32, the items are inserted using F2, instead of F1. For details 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 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 ≋ is used to mark a technical break TCU continuation that is done for purposes of improving readability and overlap alignment. In this case the triple wavy is 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. The ≈ symbol is used in a parallel way to mark a no-break TCU continuation. 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.

In order to facilitate the translation of CHAT files back into CA using the CHAT2CA program, some additional symbols are added to the CHAT files. These include +=. for

the CA beginning latch mark and ++. for missing final delimiters. These codes are noted in the chapter on terminators.

11 Disfluency Transcription

As noted in the previous chapter, CHAT uses the following forms for marking disfluencies.

- pairs of Unicode 21AB leftwards arrow with loop to mark segment repetition as in ↫b-b-b↫boy or ba↫na-na↫nana.
- a single Unicode 2260 not-equal-to sign to mark blocking, as in ≠rubber
- the colon for marking drawls or extensions
- the ^ symbol for marking a break inside a work
- forms such as &-*um* for marking filled pauses
- silent pauses as marked by (.) or (0.6) etc.
- [/] string for word or phrase repetition
- string [//] string for retracing
- +… for trailing off

These disfluency types can be traced in FREQ and KWAL commands through these search strings:

&-*	fillers
↫*	initial repetition
≠	blocking
^	internal pausing
:	drawling, lengthening
[/]	word repetitions
[//]	retracings
</>	actual words repeated
<//>	actual words retraced

The file called fluency-s.cut in the /lib/fluency folder of the CLAN distribution includes strings for searching for these patterns. The command to use this file is

```
freq +s@fluency-s.cut *.cha
```

12 Arabic Transcription

In order to transcribe Arabic in Roman characters, the SemTalk group has devised the following coding system. This system and related materials on the acquisition of Arabic, Hebrew, and other Semitic languages can be found at semtalk.talkbank.org. The transcription conventions for Arabic were set by the following people: Hanan Asaad - Haifa University, Abbassia Bouhaous - University of Heidelberg, Amal Kadry – Bar-Ilan University, Lior Laks – Tel Aviv University, Bracha Nir-Sagiv – Tel Aviv University, Fatena Omar - Haifa University, Sigal Uziel-Karl – Haifa University

The following charts list the Arabic transcription symbols. To use the special symbols, the latest CHILDES Unicode version has to be installed. Once this is done, the symbols with the diacritics can be inserted as follows: To insert the diacritic representing pharyngeals, type any of the requested letter(s), and then press the F2 and comma keys to get the diacritic. To insert the superscript 'h' (to represent interdental fricative, the voiced velar fricative, or the interdental emphatic), hold down shift+F2 and type h. When transcribing geminates, e.g. shaddah – use double consonants.

Vowels

Symbol	Letter	Name
u	ُ	dame
i	ِ	kasra
a:	ا	alef
u:	و	waw
i:	ي	ya
e	ي	ya (ba'd**e**n)
o	و	waw (bantal**o**n)

Consonants

Symbol	Letter	Name
`	إ	hamza
b	ب	ba
t	ت	ta
t^h	ث	tha
j	ج	jim
ḥ	ح	ḥa
x	خ	xa
d	د	dal
d^h	ذ	dhal
r	ر	ra
z	ز	zen
sh	ش	shin
s	س	sin
ṣ	ص	sad
ḍ	ض	dad
$ḍ^h$	ظ	ẓa
ṭ	ط	ṭa
'	ع	'en
g^h	غ	gen
f	ف	fa
q	ق	qaf
k	ك	kaf
l	ل	lam
m	م	mim
n	ن	nun
h	ه	ha
w	و	waw
y	ي	ya

13 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.

13.1 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 provides several ways of dealing with code-switching. The selection of some or all of these methods of notation depends primarily on the user's needs for retrieval of codes during analysis.

1. The languages spoken by the various participants can be noted with the @Language of XXX header tier.
2. Individual words may be identified with the @s terminator to indicate their second language status. The exact identity of the second language can be coded as needed. For example, words in French could be noted as @f and words in German as @g. In the limiting case, it would be possible to mark every single word in a French-German bilingual transcript as either @f or @g. Of course, doing this would be tedious, but it would provide a complete key for eventual retrieval and study.
3. It is possible to use the six-letter code for the main tier as an easy way of indicating the matrix language being used for each utterance. For example, *CHIGG could indicate the child speaking German to a German speaker and *CHIGF could indicate the child speaking German to a French speaker. Retrieval during analysis would then rely on the use of the +t switch, as in +t*CHIG*, +t*CHUGG, and +t*CHI*.
4. The system of gem markers can also be used to indicate the beginnings and ends of segments of discourse in particular languages.
5. A large database may consist of files in certain well-specified interaction types. For example, conversations with the mother may be in German and those with the father in French. If this is the case, the careful selection of file names such as ger01.cha and fre01.cha can be used to facilitate analysis.

These techniques are all designed to facilitate the retrieval of material in one language separately from the other. The choice of one method over another will depend on the nature of the material being transcribed and the eventual goals of the analysis.

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. However, the other forms discussed above for noting code-switching can also be used for these purposes.

13.2 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 implest 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.

13.3 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.

13.4 Nested Files for Gesture Analysis

The analysis of gestural, postural, and proxemics communication requires careful attention to detail. Because gesture uses many regions of the body, and because multiple participants often produce gestures in parallel, there is an enormous amount of concurrent or parallel activity going on across the various gestural and verbal streams, particularly in conversations involving more than two people. One approach to dealing with this multiple parallelism uses the Partitur or musical notation display found in ELAN or EXMARaLDA. CHAT files can be converted to these formats for the analysis of gesture. However, there are good reasons to consider working instead inside CLAN, even for these more complex analyses. In this section, we will describe how micromultimodal analysis for gesture can be constructed inside CLAN.

The example we will use comes from the frokost.mov video recorded by Lone Laursen and contributed to TalkBank in 2006. Lone and Brian MacWhinney then worked out a method of analyzing these data in a set of nested files that we will explain here. This interaction involves the discussion in a Danish hospital lunchroom of an experiment at the hospital on reaction to pain. In this short narrative, Deedee, the principal storyteller, explains how some Japanese visitors were assigned to the "pain group" which involved the use of some chili pepper and possibly some boiling water. Apparently, the actual pain endured was not so much, but the ladies enjoy joking about the idea that they were administering a sort of minimal pain during *torturtid* or "torture time". During this interaction, Deedee is facing across the table to Nina with Else between the two on the backside of the table. It is possible to analyze out five largely independent sequences of co-constructed gestures that we labeled as: 1torturtid, 2smertegruppen, 3kogepladde, 4kanikkmenskal, and 5brutalt. For each of these five sequences, we produced microanalytic files using the CLAN editor. These files are not full CHAT files and they only need to follow a few formatting guidelines, which we will explain here.

These sample files can be downloaded from http://talkbank.org/CABank/frokost.zip. They include the basic CHAT transcripts called frokost.cha, the frokost.mov movie file, and the five microanalytic coding files. Before looking at how to create these files, let's see how they work. Open up the frokost.cha file and command-click (Mac) or control-click (PC) on the 14th line which begins with the @G code. After this, the "nested" file named 1torturtid.cut will start to open, but first you see a dialog saying that CLAN is loading 12 movie clips. These are pointers to various segments of the overall frokost.mov movie file. The file will then be opened in read-only mode and you can click on both the thumbnails that are inserted and the various other bullets in the file to play back the linked segments.

Now let us take a look at how these nested gesture-coding files are created and used. First, to insert references to the nested files within the main transcript file, you need to enter an @G line such as this one in which the bullets are expanded.

@G: torture time sequence •%txt:"1torturtid"•

To do this, you type the first part directly. Then, to enter the material between the two bullets, you select "Insert Bullet into Text" from the Mode menu, or you can type Command-I. A dialog then pops up asking you to "Please locate movie, sound, picture, or text file". In this case, you want a text file such as 1torturtid.cut. Repeat this for each of the five coding sequences you plan to elaborate.

Now let us take a look at what you may want to put inside each of the nested gesture sequence coding files. Here is the first part of the first one. This is the segment in which Deedee prepares herself to tell the story, saying *så er de snart torturtid* "so this is soon torture time." During this time Nina rests her chin on her elbow, signaling her readiness to listen to the story:

********** begin segment **********

@Media: frokost, audio

```
Sequences:D1-D2-D3-D4  torture time by Deedee
          N1-N2-N3-N4  oh my gosh by Nina

@T:       Segment              D1
Action          hands brushing table
Gaze            down to hands
Classification  prep, story, sync:så
Meaning         preamble to action

Segment         D2
Action          fingers touching lightly, thumbs up
Gaze            front to Nina
Classification  fixation, story, sync:torturtid
Meaning         action focus

Segment         N1
Action          rests chin on hand, elbow on table, right shoulder back
Gaze            front to Deedee
Classification  attention
Meaning         attention

*D:       ⌜så er det snart⌝⌜torturtid→⌝
%ges:     ⌊-------D1-----⌋⌊----D2---⌋
           ⌊----------N1----------->⌋
%com:     assimilating the pronounciation of a danish actor in a then tv show


%pic:     D1        D2

********************  end segment  ***************
```

At the beginning of this first segment, there is an @Media line that tells CLAN which media to play and whether it is video or audio. After that, there is a short summary of the two gesture sequences being coded. The D1-D2-D3-D4 sequence has four gestures from Deedee and the N1-N2-N3-N4 sequence has four gestures from Nina. After this summary comes a listing of the first three gestures that are produced in synchrony (D1, D2, and N1). Each gesture is described along four dimensions: action, gaze direction, classification, and meaning. The classification line also includes a statement about the synchrony between the gesture and one or more words. At this stage in our understanding of gesture coding, it makes no sense to restrict the vocabulary in these descriptions to some officially sanctioned set. Instead, it should be up to each investigator to find coding and explanation systems that work best for his/her project. However, if this vocabulary is carefully selected, then one can use CLAN's KWAL program to locate all instances of a given coding term across a collection of *.cut coding files. For example, if one uses the term "Iconic", then this command will locate all the iconic gestures, and the output lines of KWAL can be triple-clicked to go back to each source line.

```
kwal +sIconic *.cut
```

Finally, you may wish to enter a few lines that show the overall alignment of the gestures with the speech. The %pic line at the end of the sequence encodes the positioning of thumbnails that will be inserted in the thumbnail process to be described next. This line must have the %pic: header at the beginning of the line. After that the positions of the names and bullets for the thumbnails will be used to place them correctly on the screen. Each of these coding subfiles can be opened directly and edited. However, when you command-click on their @G lines in the main file, then CLAN opens them as read-only files with pictures inserted for each frame in the %pic line.

We have so far described only the beginning segment of the first of five coding files for this two-minute segment of a one-hour file. It is clear that gesture coding, while fascinating, is extremely time intensive. However, by looking more closely through this type of microanalytic multimodal microscope, we can learn a lot about the texture of interactions. To appreciate this more fully, please browse further through the frokost.cha sample.

13.5 Sign Language Transcription

The TalkBank database at http://talkbank.org includes a sample file called “bonbon.cha” that illustrates the ways in which a short passage in American Sign Language (ASL) can be transcribed in CHAT. This transcript is linked through bullets to segments of a video that can also be download from the TalkBank website. Here is a sample sentence encoded in this format with the links to the video marked with asterisks. In the actual CHAT transcript, these are bullets that can be expanded to show the linked time values.

*BON: the waves reached as high as five feet. *
%xss1: water * CL: rising and lowering of a swell * CL: rising of the
swell at its apex * five * F-E-E-T * gesture = woo;wow
;serious;"heavy" *
%xss3: one handed * two handed CL; 5 hand shape * one hand * one handed *
%com: her left hand is making the CL for water * Expressed with emphasis *
%xss2: eyes are looking up ; mouth slightly open * pursed lips *

13.6 Sign and Speech

CHAT can also be used to analyze interactions that combine signed and spoken language. For example, the 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.

14 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.

14.1 Interchange Types

Table 44: 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

14.2 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.

15 Error Coding

Errors are marked by placing the [*] symbol after the error. If there is a replacement string, such as [: because], that should come before the error code. 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. CHAT codes errors by symbols placed after the asterisk in the [*] field. This system provides error codes at the word and utterance level.

15.1 Word level error codes summary

[* p] phonological error

p:w	word
p:n	non-word
p:m	methathesis

[* s] semantic error

s:r	related word, target known
s:ur	unrelated word, target known
s:uk	word, unknown target
s:per	perseveration

[* n] neologism

n:k	neologism, known target
n:uk	neologism, unknown target
n:k:s	neologism, known target, stereotypy
n:uk:s	neologism, unknown target, stereotypy

[* d] dysfluency

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

[* m:a] morphological agreement error

m:a	agreement error, as in *have* for *has*
m:a:0es	missing 3rd person singular -s suffix (agreement error)
m:a:+es	superfluous 3rd person singular agreement error, as in *we goes*
m:a:0s	missing plural on noun, as in *two sister*
m:a:+s	superfluous plural on noun, as in *one dogs*

Agreement errors should only be marked on the verb or the head noun, not on both agreeing items.

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

-ing	progressive
-s	plural
-es	3rd person singular
-ed	past
-en	perfective
-‘s	possessive

[* m] other morphological errors

m:c	case error
m:0s	missing plural –s suffix, including *child* for *children*
m:0's	missing possessive -s suffix
m:0ing	missing progressive –ing suffix
m:0ed	missing past -ed suffix, including *come* from *came*
m:=ed	overregularized -ed, as in *seed* for *saw*
m:=s	overregularized –s, as in *childs* for *children*
m:+s	superfluous plural, as in *feets* for *feet*, *gowns* for *gown*, or *children* for *child*
m:+ed	superfluous past, as in *ranned* for *ran*, *walked* for *walk*, or came for come
m:+ing	superfluous progressive, as *coming* for *come*
m:+es	superfluous 3rd person singular –s suffix
m:++s	double –s, as in *kniveses*
m:m	morphophonological error as in *knifes* for *knives*

[* f] formal lexical device

f:a:0:d	article should be zero, used “the”
f:a:0:i	article should be zero, used “a”
f:a:d	article, should be definite, used indefinite
f:a:i	article, should be indefinite, used definite
f:p	part of speech wrong, as in *mine* for *my*, also part of speech errors involving derivations, such as *assess* > *assessment*

0 missing word or part of speech **– 0art, 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].

15.2 Word level coding – details

- When the error is a real word, even though it is the wrong real word, the target should be marked with the special replacement form using a double colon, as in:

  it was singing [:: ringing] [* s:r] [* p:w] in my ears.

Doing this 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.

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

  it was singing [:: ringing] [* s:r] [* p:w] in my ears.

- Also, if the error is repeated, add "-rep" to the error code; if the error is revised (to another error or the correct word), add "-ret" to the error code, as in this example:

  he [: she] [* s:r-ret] [//] she said it was fine.

[* p] – Phonological errors

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

- 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.
- 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: Errors that do not meet these criteria (e.g., only 1 of 3 elements match) will be coded as [* s:ur] if they are real words and [* n:k] if they are non-words, so if you prefer to use less strict criteria for phonological errors, you should check these categories as well.

[* p:w] – the error is a real word (note the use of the double colon)

heat [:: eat]	gun [:: done]
breeding [:: bleeding]	boater [:: butter]
say [:: see]	cable [:: table]
we [:: three]	bag [:: bad]

[* p:n] – the error is a non-word – transcribe using IPA and attach @u to the error

pɜˑsɪtʃ@u [: person]	kɛlɪŋ@u [: telling]

peɪbl̩@u [: table] lɛθ@u [: left]

[* p:m] – the error involves metathesis – transcribe using IPA and attach @u to the error

stɪsə˞z@u [: sisters]
mɪdwɛts@u [: midwest]

[* s] – Semantic errors

[* s:r] – the error is a recognizable English word that is semantically related to the target word. Note the use of the double colon to indicate that the error is a real word.

fork [:: knife] anybody [:: anything]
prince [:: princess] he [:: she]

[* s:ur] – the error is a real word for which the target is known but it is not semantically related to the target, and does not meet the criteria for phonological errors as stated above

fares [:: scared]
hi [:: time]
fry [:: drive]

[* s:uk] – the error is a real word for which the target is unknown. Here, the words marked as errors made no sense given the context in which they were used. Therefore, no target form is given.

PAR: I'm going to get somebody bough [* s:uk] .
PAR: and I go wolf [* s:uk] .

[* s:per] – repetition of a word when it is no longer appropriate (Brookshire, 1997)

PAR: The boy kicked the ball through the ball [:: window] [* s:per] .

[* n] – Neologistic errors – transcribe using IPA and attach @u to the error

[* n:k] – the error is a non-word for which the target is known but does not meet criteria for semantic or phonemic errors as stated above

PAR: she had all her gɹæstɪdʒɪz@u [: groceries] [* n:k].

[* n:uk] – the error is a non-word for which the target is not known, **add [: x@n] as the target word after the error**

Examples:
PAR: I've only gone to two ɻɛsɪz@u [: x@n] [n:uk] .
PAR: if I could fɹeɪv@u [: x@n][n:uk] it I guess I can bɹæm@u [: x@n] [* n:uk] it.

[* n:k:s] – the error is a recurring non-word for which the target is known

[* n:uk:s] – the error is a recurring non-word for which the target is not known, **add [: x@n] as the target word after the error**

0word – Missing word errors

0art, 0aux, etc. – missing word or part of speech

CHAT transcript examples:
*PAR: and that 0cop what 0art peanut butter sandwich is.
*PAR: the boy 0aux falling.

15.3 Utterance level error coding (post-codes)

[+ 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

CHAT transcript examples:
*PAR: one two bread. [+ gram]
*PAR: whatever I'm think up. [+ gram]
*PAR: I don't think that they're not supposed to be doing that. [+ gram]
*PAR: what'd he put it to me? [+ gram]

*PAR: they were knowing them and they didn't know what was going on
either too. [+ gram]
*PAR: is getting want to be wasn't. [+ gram]
*PAR: when everything that we going out now. [+ gram]

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 but be syntactically correct; 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]
PAR: this guy is &cr &h hæpɪŋ@u [: helping] [p:n] with other people having problems . [+ cir]

16 Morphosyntactic Coding

Many students of child language are interested in examining the role of universals in language acquisition. To test for the impact of universals, researchers need to examine the development of grammatical marking and syntax in corpora from different languages. If such research is to be conducted efficiently, it must be made available to computational analysis. This requires use of a standard representation of morphosyntactic codings. This chapter presents a system for constructing such a representation, using the %mor tiers. This system is intended as a full replacement for the system of main line morpheme segmentation that was used in CHAT from 1984 to 1996. It is now possible to automatically generate a %mor tier from a main tier by using the MOR command. At present, MOR grammars exist for English, French, German, Hebrew, Italian, Japanese, Cantonese, Mandarin, and Spanish. The grammars for French, and German are based on full listings of forms, rather than morphological analysis by parts, using the minMOR approach. The other grammars us the analytic method outlined in the chapter on MOR in the CLAN manual. Both types of grammars insert codes using the format described in this chapter.

16.1 One-to-one correspondence

MOR creates a %mor tier with a one-to-one correspondence between words on the main line and words on the %mor tier. In order to achieve this one-to-one correspondence, the following rules are observed:

1. Each word group (see below) on the %mor line is surrounded by spaces or an initial tab to correspond to the corresponding space-delimited word group on the main line. The correspondence matches each %mor word (morphological word) to a main line word in a left-to-right linear order in the utterance.
2. Utterance delimiters are preserved on the %mor line to facilitate readability and analysis. These delimiters should be the same as the ones used on the main line.
3. Along with utterance delimiters, the satellite markers of ‡ for the vocative and „ for tag questions or dislocations are also included on the %mor line in a one-to-one alignment format.
4. Retracings and repetitions are excluded from this one-to-one mapping, as are nonwords such as xxx or strings beginning with &. When word repetitions are marked in the form word [x 3], the material in parentheses is stripped off and the word is considered as a single form.
5. When a replacing form is indicated on the main line with the form [: text], the material on the %mor line corresponds to the replacing material in the square brackets, not the material that is being replaced. For example, if the main line has **gonna [: going to],** the %mor line will code **going to.**
6. The [*] symbol that is used on the main line to indicate errors is not duplicated on the %mor line.

16.2 Tag Groups and Word Groups

On the %mor line, alternative taggings of a given word are clustered together in *tag groups*. These groups include the alternative taggings of a word that are produced by the MOR program. Alternatives are separated by the ^ character. Here is an example of a tag group for one of the most ambiguous words in English:

```
adv|back^adj|back^n|back^v|back
```

After you run the POST program on your files, all of these alternatives will be disambiguated and each word will have only one alternative. You can also use the hand disambiguation method built into the CLAN editor to disambiguate each tag group case by case.

The next level of organization for the MOR line is the word group. Word groups are combinations marked by the preclitic delimiter $, the postclitic delimiter ~ or the compound delimiter +. For example, the Spanish word *dámelo* can be represented as

```
vimpsh|da-2S&IMP~pro:clit|1S~pro:clit|OBJ&MASC=give
```

This word group is a series of three words (verb~postclitic~postclitic) combined by the ~ marker. Clitics may be either preclitics or postclitics. Separable prefixes of the type found in German or Hungarian and other discontinuous morphemes can be represented as word groups using the preclitic delimiter $, as in this example for *ausgegangen* (“gone”):

```
prep|aus$PART#v|geh&PAST:PART=go
```

Note the difference between the coding of the preclitic “aus” and the prefix “ge” in this example.

Compounds are also represented as combinations, as in this analysis of angel+fish.

```
n|+n|angel+n|fish
```

Here, the first characters (n) represent the part of speech of the whole compound and the subsequent tags, after each plus sign, are for the parts of speech of the components of the compound. Proper nouns are not treated as compounds. Therefore, they take forms with underlines instead of pluses, such as Luke_Skywalker or New_York_City.

16.3 Words

Beneath the level of the word group is the level of the word. The structure of each individual word is:

```
prefix#
part-of-speech|
stem
&fusionalsuffix
-suffix
=english (optional, underscore joins words)
```

There can be any number of prefixes, fusional suffixes, and suffixes, but there should be only one stem. Prefixes and suffixes should be given in the order in which they occur in the word. Since fusional suffixes are fused parts of the stem, their order is indeterminate. The English translation of the stem is not a part of the morphology, but is included for convenience for non-native speakers. If the English translation requires two words, these words should be joined by an underscore as in "lose_flowers" for French *defleurir*.

Now let us look in greater detail at the nature of each of these types of coding. Throughout this discussion, bear in mind that all coding is done on a word-by-word basis, where words are considered to be strings separated by spaces.

16.4 Part of Speech Codes

The morphological codes on the %mor line begin with a part-of-speech code. The basic scheme for the part-of-speech code is:

category:subcategory:subcategory

Additional fields can be added, using the colon character as the field separator. The subcategory fields contain information about syntactic features of the word that are not marked overtly. For example, you may wish to code the fact that Italian "andare" is an intransitive verb even though there is no single morpheme that signals intransitivity. You can do this by using the part-of-speech code **v:intrans**, rather than by inserting a separate morpheme.

In order to avoid redundancy, information that is marked by a prefix or suffix is not incorporated into the part-of-speech code, as this information will be found to the right of the | delimiter. These codes can be given in either uppercase, as in **ADJ**, or lowercase, as in **adj**. In general, CHAT codes are not case-sensitive.

The particular codes given below are the ones that MOR uses for automatic morphological tagging of English. Individual researchers will need to define a system of part-of-speech codes that correctly reflects their own research interests and theoretical commitments. Languages that are typologically quite different from English may have to use very different part-of-speech categories. Quirk, Greenbaum, Leech, and Svartvik (1985) explain some of the intricacies of part-of-speech coding. Their analysis should be taken as definitive for all part-of-speech coding for English. However, for many purposes, a more coarse-grained coding can be used.

The following set of top-level part-of-speech codes is the one used by the MOR program. Additional refinements to this system can be found by studying the organization of the lexicon files for that program For example, in MOR, numbers are coded as types of determiners, because this is their typical usage. The word "back" is coded as either a noun, verb, preposition, or adjective. Further distinctions can be found by looking at the MOR lexicon.

Table 45: Major Parts of Speech

Category	Code
Adjective	ADJ
Adverb	ADV
Communicator	CO
Conjunction	CONJ
Determiner	DET
Filler	FIL
Infinitive marker *to*	INF
Noun	N
Proper Noun	N:PROP
Number	DET:NUM
Particle	PTL
Preposition	PREP
Pronoun	PRO
Quantifier	QN
Verb	V
Auxiliary verb, including modals	V:AUX
WH words	WH

16.5 Stems

Every word on the %mor tier must include a "lemma" or stem as part of the morpheme analysis. The stem is found on the right hand side of the | delimiter, following any pre-clitics or prefixes. If the transcript is in English, this can be simply the canonical form of the word. For nouns, this is the singular. For verbs, it is the infinitive. If the transcript is in another language, it can be the English translation. A single form should be selected for each stem. Thus, the English indefinite article is coded as **det|a** with the lemma "a" whether or not the actual form of the article is "a" or "an."

When English is not the main language of the transcript, the transcriber must decide whether to use English stems. Using English stems has the advantage that it makes the corpus more available to English-reading researchers. To show how this is done, take the German phrase "wir essen":

```
*FRI:     wir essen.
%mor:     pro|wir=we v|ess-INF=eat .
```

Some projects may have reasons to avoid using English stems, even as translations. In this example, "essen" would be simply **v|ess-INF**. Other projects may wish to use only English stems and no target-language stems. Sometimes there are multiple possible trans-

lations into English. For example, German "Sie"/sie" could be either "you," "she," or "they." Choosing a single English meaning for the stem helps fix the German form.

16.6 Affixes

Affixes and clitics are coded in the position in which they occur with relation to the stem. The morphological status of the affix should be identified by the following markers or delimiters: - for a suffix, # for a prefix, and **&** for fusional or infixed morphology.

The **&** is used to mark affixes that are not realized in a clearly isolable phonological shape. For example, the form "men" cannot be broken down into a part corresponding to the stem "man" and a part corresponding to the plural marker, because one cannot say that the vowel "e" marks the plural. For this reason, the word is coded as **n|man&PL**. The past forms of irregular verbs may undergo similar ablaut processes, as in "came," which is coded **v|come&PAST**, or they may undergo no phonological change at all, as in "hit", which is coded **v|hit&PAST.** Sometimes there may be several codes indicated with the **&** after the stem. For example, the form "was" is coded **v|be&PAST&13s**. Affix and clitic codes are based either on Latin forms for grammatical function or English words corresponding to particular closed-class items. MOR uses the following set of affix codes for automatic morphological tagging of English.

Table 46: Inflectional Affixes for English

Function	Code
adjective suffix *er*, *r*	CP
adjective suffix *est*, *st*	SP
noun suffix *ie*	DIM
noun suffix *s*, *es*	PL
noun suffix *'s*, *'*	POSS
verb suffix *s*, *es*	3S
verb suffix *ed*, *d*	PAST
verb suffix *ing*	PRESP
verb suffix *en*	PASTP

Table 47: Derivational Affixes for English

Function	Code
adjective and verb prefix *un*	UN
adverbializer *ly*	LY
nominalizer *er*	ER
noun prefix *ex*	EX
verb prefix *dis*	DIS
verb prefix *mis*	MIS
verb prefix *out*	OUT
verb prefix *over*	OVER

verb prefix *pre*	PRE
verb prefix *pro*	PRO
verb prefix *re*	RE

16.7 Clitics

Clitics are marked by a tilde, as in **v|parl&IMP:2S=speak~pro|DAT:MASC:SG** for Italian "parlagli" and **pro|it~v|be&3s** for English "it's." Note that part of speech coding with the | symbol is repeated for clitics after the tilde. Both clitics and contracted elements are coded with the tilde. The use of the tilde for contracted elements extends to forms like "sul" in Italian, "ins" in German, or "rajta" in Hungarian in which prepositions are merged with articles or pronouns.

Table 48: Clitic Codes for English

Clitic	Code
noun phrase post-clitic *'d*	v:aux|would, v|have&PAST
noun phrase post-clitic *'ll*	v:aux|will
noun phrase post-clitic *'m*	v|be&1S, v:aux|be&1S
noun phrase post-clitic *'re*	v|be&PRES, v:aux|be&PRES
noun phrase post-clitic *'s*	v|be&3S, v:aux|be&3S
verbal post-clitic *n't*	neg|not

16.8 Compounds

Here are some words that we might want to treat as compounds: *sweat+shirt*, *tennis+court*, *bathing+suit*, *high+school*, *play+ground*, *choo+choo+train*, *rock+'n+roll*, and *sit+in*. There are also many idiomatic phrases that could be best analyzed as linkages. Here are some examples: *a_lot_of*, *all_of_a_sudden*, *at_last*, *for_sure*, *kind_of*, *of_course*, *once_and_for_all*, *once_upon_a_time*, *so_far*, and *lots_of*.

On the %mor tier it is necessary to assign a part-of-speech label to each segment of the compound. For example, the word *blackboard* or *black+board* is coded on the %mor tier as **n|+adj|black+n|board**. Although the part of speech of the compound as a whole is usually given by the part-of-speech of the final segment, forms such as *make+believe* which is coded as adj|+v|make+v|believe show that this is not always true.

In order to preserve the one-to-one correspondance between words on the main line and words on the %mor tier, words that are not marked as compounds on the main line should not be coded as compounds on the %mor tier. For example, if the words "come here" are used as a rote form, then they should be written as "come_here" on the main tier. On the %mor tier this will be coded as **v|come_here**. It makes no sense to code this as **v|come+adv|here**, because that analysis would contradict the claim that this pair functions as a single unit. It is sometimes difficult to assign a part-of-speech code to a

morpheme. In the usual case, the part-of-speech code should be chosen from the same set of codes used to label single words of the language. For example, some of the these idiomatic phrases can be coded as compounds on the %mor line.

Table 49: Phrases Coded as Linkages

Phrase	Phrase
qn|a_lot_of	adv|all_of_a_sudden
co|for_sure	adv:int|kind_of
adv|once_and_for_all	adv|once_upon_a_time
adv|so_far	qn|lots_of.

16.9 Punctuation Marks

MOR can be configured to recognize certain punctuation marks as whole word characters. In particular, the file punct.cut contains these entries:

„ {[scat end]} "end"
‡ {[scat beg]} "beg"
, {[scat cm]} "cm"
“ {[scat bq]} "bq"
” {[scat eq]} "eq"
‘ {[scat bq]} “bq2”
’ {[scat eq]} “eq2”

When the punctuation marks on the left occur in text, they are treated as separate lexical items and are mapped to forms such as beg|beg on the %mor tier. The “end” marker is used to mark postposed forms such as tags and sentence final particles. The “beg” marker is used to mark preposed forms such as vocatives and communicators. The “bq” marks the beginning of a quote and the “eq” marks the end of a quote. These special characters are important for correctly structuring the dependency relations for the GRASP program.

16.10 Sample Morphological Tagging for English

The following table describes and illustrates a more detailed set of word class codings for English. The %mor tier examples correspond to the labellings MOR produces for the words in question. It is possible to augment or simplify this set, either by creating additional word categories, or by adding additional fields to the part-of-speech label, as discussed previously. The entries in this table and elsewhere in this manual can always be double-checked against the current version of the MOR grammar by typing “mor +xi” to bring up interactive MOR and then entering the word to be analyzed.

Table 50: Word Classes for English

Class	Examples	Coding of Examples

adjective	big	adj|big
adjective, comparative	bigger, better	adj|big-CP, adj|good&CP
adjective, superlative	biggest, best	adj|big-SP, adj|good&SP
adverb	well	adv|well
adverb, ending in ly	quickly	adv:adj|quick-LY
adverb, intensifying	very, rather	adv:int|very, adv:int|rather
adverb, post-qualifying	enough, indeed	adv|enough, adv|indeed
adverb, locative	here, then	adv:loc|here, adv:tem|then
communicator	aha	co|aha
conjunction, coordinating	and, or	conj:coo|and, conj:coo|or
conjunction, subord.	if, although	conj:sub|if, conj:sub|although
determiner, singular	a, the, this	det|a, det|this
determiner, plural	these, those	det|these, det|those
determiner, possessive	my, your, her	det:poss|my
infinitive marker	to	inf|to
noun, common	cat, coffee	n|cat, n|coffee
noun, plural	cats	n|cat-PL
noun, possessive	cat's	n|cat~poss|s
noun, plural possessive	cats'	n|cat-PL~poss|s
noun, proper	Mary	n:prop|Mary
noun, proper, plural	Mary-s	n:prop|Mary-PL
noun, proper, possessive	Mary's	n:prop|Mary~poss|s
noun, proper, pl. poss.	Marys'	n:prop|Mary-PL~poss|s
noun, adverbial	home, west	n|home, adv:loc |home
number, cardinal	two	det:num|two
number, ordinal	second	adj|second
postquantifier	all, both	post|all, post|both
preposition	in	prep|in, adv:loc|in
pronoun, personal	I, me, we, us, he	pro|I, pro|me, pro|we, pro|us
pronoun, reflexive	myself, ourselves	pro:refl|myself
pronoun, possessive	mine, yours, his	pro:poss|mine, pro:poss:det|his
pronoun, demonstrative	that, this, these	pro:dem|that
pronoun, indefinite	everybody, nothing	pro:indef|everybody
pronoun, indefinite, poss.	everybody's	pro:indef|everybody~poss|s
quantifier	half, all	qn|half, qn|all
verb, base form	walk, run	v|walk, v|run
verb, 3rd singular present	walks, runs	v|walk-3S, v|run-3S
verb, past tense	walked, ran	v|walk-PAST, v|run&PAST
verb, present participle	walking, running	part|walk-PRESP, part|run-PRESP
verb, past participle	walked, run	part|walk-PASTP, part|run&PASTP
verb, modal auxiliary	can, could, must	aux|can, aux|could, aux|must

Since it is sometimes difficult to decide what part of speech a word belongs to, we offer the following overview of the different part-of-speech labels used in the standard English grammar.

ADJectives modify nouns, either prenominally, or predicatively. Unitary compound modifiers such as **good-looking** should be labeled as adjectives.

ADVerbs cover a heterogenous class of words including: manner adverbs, which generally end in **-ly**; locative adverbs, which include expressions of time and place; intensifiers that modify adjectives; and post-head modifiers, such as **indeed** and **enough.**

COmmunicators are used for interactive and communicative forms which fulfill a variety of functions in speech and conversation. Also included in this category are words used to express emotion, as well as imitative and onomatopeic forms, such as **ah**, **aw**, **boom**, **boom-boom**, **icky**, **wow**, **yuck**, and **yummy.**

CONJunctions conjoin two or more words, phrases, or sentences. Coordinating conjunctions include: **and**, **but**, **or**, and **yet**. Subordinating conjunctions include: **although**, **because**, **if**, **unless,** and **until.**

COORDinators include **and, or,** and **as well as**. These can combine clauses, phrases, or words.

DETerminers include articles, and definite and indefinite determiners. Possessive determiners such as **my** and **your** are tagged **det:poss.**

INFinitive is the word "to" which is tagged **inf|to.**

INTerjections are similar to communicators, but they typically can stand alone as complete utterances or fragments, rather than being integrated as parts of the utterances. They include forms such as **wow, hello, good-morning, good-bye, please, thank-you**.

Nouns are tagged with **n** for common nouns, and n**:prop** for proper nouns (names of people, places, fictional characters, brand-name products).

NEGative is the word "not" which is tagged **neg|not.**

NUMbers are labelled **num** for cardinal numbers. The ordinal numbers are adjectives.

Onomatopoeia are words that imitate the sounds of nature, animals, and other noises.

Particles are words that are often also prepositions, but are serving as verbal particles.

PREPositions are the heads of prepositional phrases. When a preposition is not a part of a phrase, it should be coded as a particle or an adverb.

PROnouns include a variety of structures, such as reflexives, possessives, personal pronouns, deictic pronouns, etc.

QUANTifiers include **each, every, all, some,** and similar items.

Verbs can be either main verbs, copulas, or auxililaries.

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* (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: Cambridge University Press.

Bernstein-Ratner, N., Rooney, B., & MacWhinney, B. (1996). Analysis of stuttering using CHILDES and CLAN. *Clinical Linguistics and Phonetics, 10*, 169-187.

Bloom, L., & Lahey, M. (1973). *Language development and language disorders*. New York: John Wiley & Sons.

Bloom, L., Lightbown, P., & Hood, L. (1975). Structure and variation in child language. *Monographs of the Society for Research in Child Development, 40*, whole no. 2.

Bloom, P. (2000). *How children learn the meanings of words*. Cambridge, MA: MIT Press.

Branigan, G. (1979). Some reasons why successive single word utterances are not. *Journal of Child Language, 6*, 411-421.

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: Cambridge University Press.

Darwin, C. (1877). A biographical sketch of an infant. *Mind, 2*, 292-294.

De Houwer, A. (1990). *The acquisition of two languages from birth: A case study*. New York: Cambridge University Press.

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.

Elbers, L., & Wijnen, F. (1993). Effort, production skill, and language learning. In C. Ferguson, L. Menn & C. Stoel-Gammon (Eds.), *Phonological development* (pp. 337-368). Timonium, MD: York.

Fletcher, P. (1985). *A child's learning of English*. Oxford: Blackwell.

Gerken, L. (1991). The metrical basis for children's subjectless sentences. *Journal of Memory and Language, 30*, 431-451.

Gerken, L., Landau, B., & Remez, R. E. (1990). Function morphemes in young children's speech perception and production. *Developmental Psychology, 26*(2), 204-216.

Goldman-Eisler, F. (1968). *Psycholinguistics: Experiments in spontaneous speech*. New York: 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*, 556-569.

MacWhinney, B. (1989). Competition and lexical categorization. In R. Corrigan, F. Eckman & M. Noonan (Eds.), *Linguistic categorization* (pp. 195-242). Philadelphia: Benjamins.

MacWhinney, B., & Osser, H. (1977). Verbal planning functions in children's speech. *Child Development, 48*, 978-985.

Malvern, D. D., Richards, B. J., Chipere, N., & Purán, P. (2004). *Lexical diversity and language development*. New York: 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: Academic.

Ochs, E. A., Schegloff, M., & Thompson, S. A. (1996). *Interaction and grammar*. Cambridge: Cambridge University Press.

Oshima-Takane, Y., & MacWhinney, B. (1995). *Japanese CHAT Manual*. Tokyo: Tokyo 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, E., 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*. 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: ACL.

Sagae, K., MacWhinney, B., & Lavie, A. (2004). Automatic parsing of parent-child interactions. *Behavior Research Methods, Instruments, and Computers, 36*, 113-126.

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: Academic Press.

Sokolov, J. L., & Snow, C. (Eds.). (1994). *Handbook of Research in Language Development using CHILDES*. Hillsdale, NJ: Erlbaum.

Sperberg-McQueen, C. M., & Burnard, L. (Eds.). (1992). *Guidelines for electronic text encoding and interchange*. Waterloo: University of Waterloo.

Stemberger, J. (1985). *The lexicon in a model of language production*. New York: 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.

Selting, M. (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: Academic Press.

Slobin, D. I. (1993). Coding child language data for crosslinguistic analysis. In J. A. Edwards & M. D. Lampert (Eds.), *Talking data: Transcription and coding in discourse research* (pp. 207- 219). Hillsdale, NJ: Lawrence Erlbaum Associates.

Sokolov, J. L., & Snow, C. (Eds.). (1994). *Handbook of research in language development using CHILDES*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Sperberg-McQueen, C. M., & Burnard, L. (Eds.). (1992). *Guidelines for electronic text encoding and interchange*. Waterloo: University of Waterloo.

Stemberger, J. (1985). *The lexicon in a model of language production*. New York: Garland.

Stern, C., & Stern, W. (1907). *Die Kindersprache*. [Child language.] Leipzig: Barth.

Svartvik, J., & Quirk, R. (Eds.). (1980). *A corpus of English conversation*. Lund: Gleerup/ Liber.

Szuman, S. (1955). Rozwój tresci slownika u dziece. *Studia Pedagogicane*, 2.

Talmy, L. (1985). Lexicalization patterns: Semantic structure in lexical forms. In T. Shopen (Ed.), *Language typology and semantic description: Vol. 3. Grammatical categories and the lexicon* (pp. 36-149). Cambridge, UK: Cambridge University Press.

Trager, G. (1958). Paralanguage: A first approximation. *Studies in Linguistics, 13,* 1- 12.

Wernicke, C. (1874). *Die Aphasische Symptomenkomplex*. [The aphasic symptom complex.] Breslau: Cohn & Weigart.