EXMARaLDA and CHAT/CLAN

Last updated: 26-11-2010

This document explains interoperability between EXMARaLDA and CHAT/CLAN.

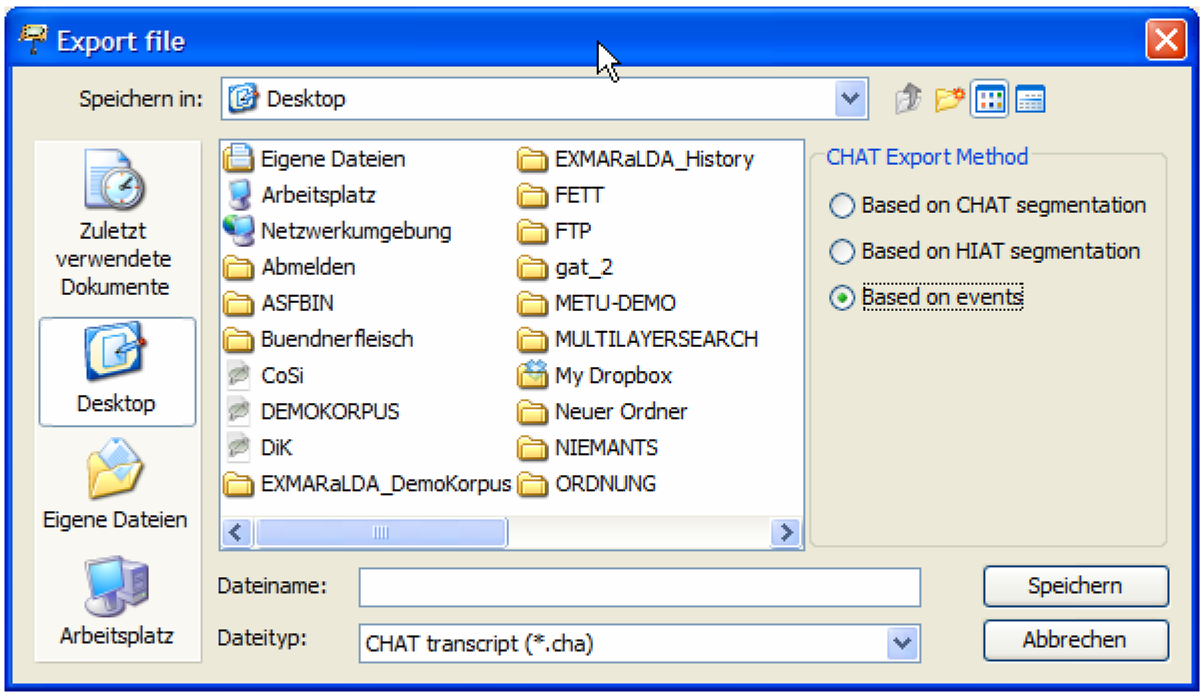
Contents

1. Introduction

This document explains how to import CHAT data into and export CHAT data from EXMARaLDA. CHAT is the data format read and written by the CLAN tool (see http://childes.psy.cmu.edu/). CHAT/CLAN and EXMARaLDA are both systems for creating and analyzing spoken language corpora. While they may differ in their general approaches and in the means chosen for their implementation, they also have a lot in common. We know from experience that there are many situations in which researchers either want to transform existing EXMARaLDA data into CHAT data or the other way around. This is possible in general. However, the transformation should be handled with caution. There is interoperability between the two systems, but it is not perfect. This means that by going from one system to the other, you may lose some information on the way, or the target data set may differ in certain respects from what you expect. The methods described here have been tested with a lot of “real” EXMARaLDA transcription data and with a lot of data from CHILDES and Talkbank. If they do not work as desired for your own data, we are grateful for any suggestions for improvement. Note that this document only describes the import and export routines built into EXMARaLDA. CLAN has its own import and export routines, described in the CHILDES manuals.

2. Exporting CHAT data from EXMARaLDA

CHAT export in EXMARaLDA is done through **File > Export…**. This will bring up a file dialog in which you can choose **CHAT transcript (*.cha)** as one of several export options.



Speichern in: Desktop

CHAT Export Method

○ Based on CHAT segmentation

○ Based on HIAT segmentation

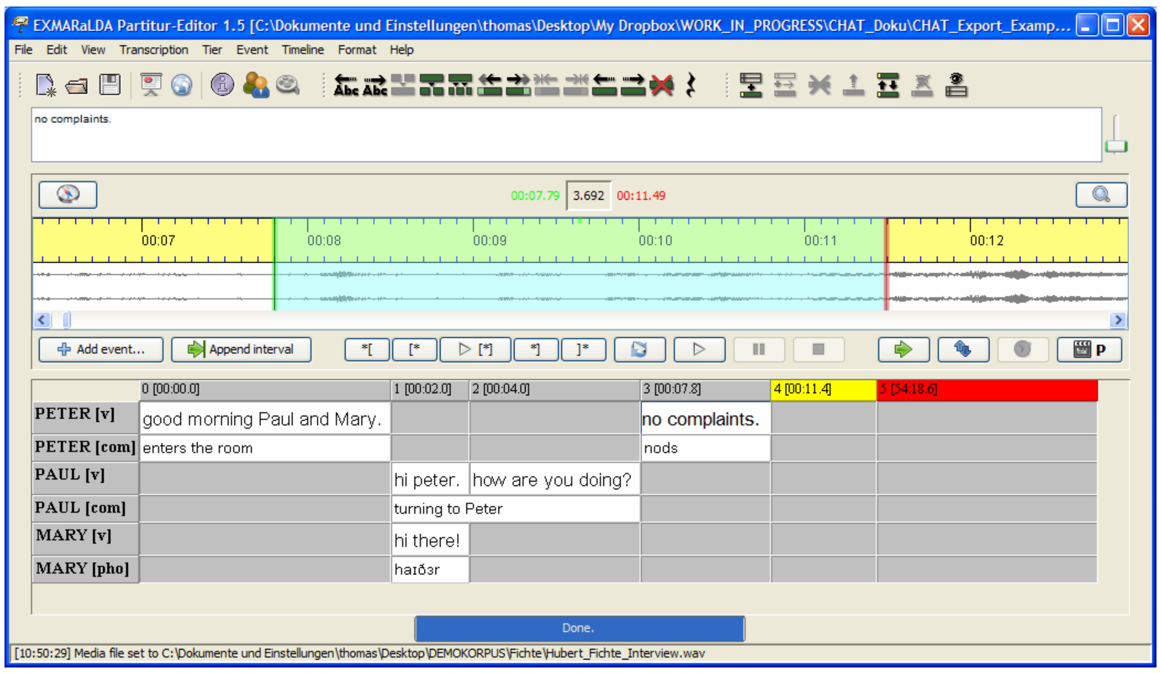
● Based on events

Dateiname:

Dateityp: CHAT transcript (*.cha)

Figure 1: CHAT export dialog

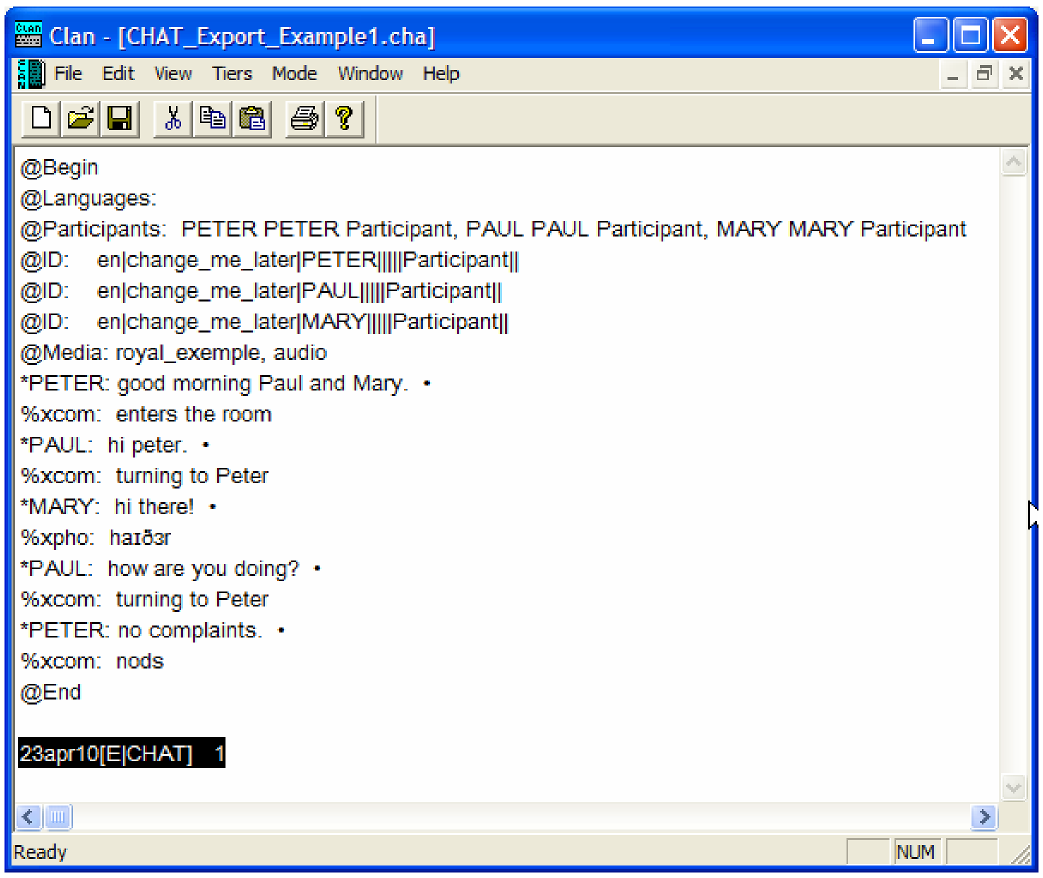
On the right side of the dialog (or, on some systems, on the bottom side), EXMARaLDA offers you three options for the export method. The simplest is to export CHAT **Based on events**. This means that each event in a tier of type ‘T(ranscription)’ in the EXMARaLDA transcription will become one utterance in the exported CHAT transcript. Events on additional tiers by the same speaker which coincide with the event will be exported to dependent tiers for that utterance. The following example illustrates this:



	0 [00:00.0]	1 [00:02.0]	2 [00:04.0]	3 [00:07.8]	4 [00:11.4]	5 [54:18.6]
PETER [v]	good morning Paul and Mary.			no complaints.		
PETER [com]	enters the room			nods		
PAUL [v]		hi peter.	how are you doing?			
PAUL [com]		turning to Peter				
MARY [v]		hi there!				
MARY [pho]		haɪðɜr				

[10:50:29] Media file set to C:\Dokumente und Einstellungen\thomas\Desktop\DEMOKORPUS\Fichte\Hubert_Fichte_Interview.wav

Figure 2: Source transcription in the Partitur-Editor...


Clan - [CHAT_Export_Example1.cha]
File Edit View Tiers Mode Window Help
Ready
NUM

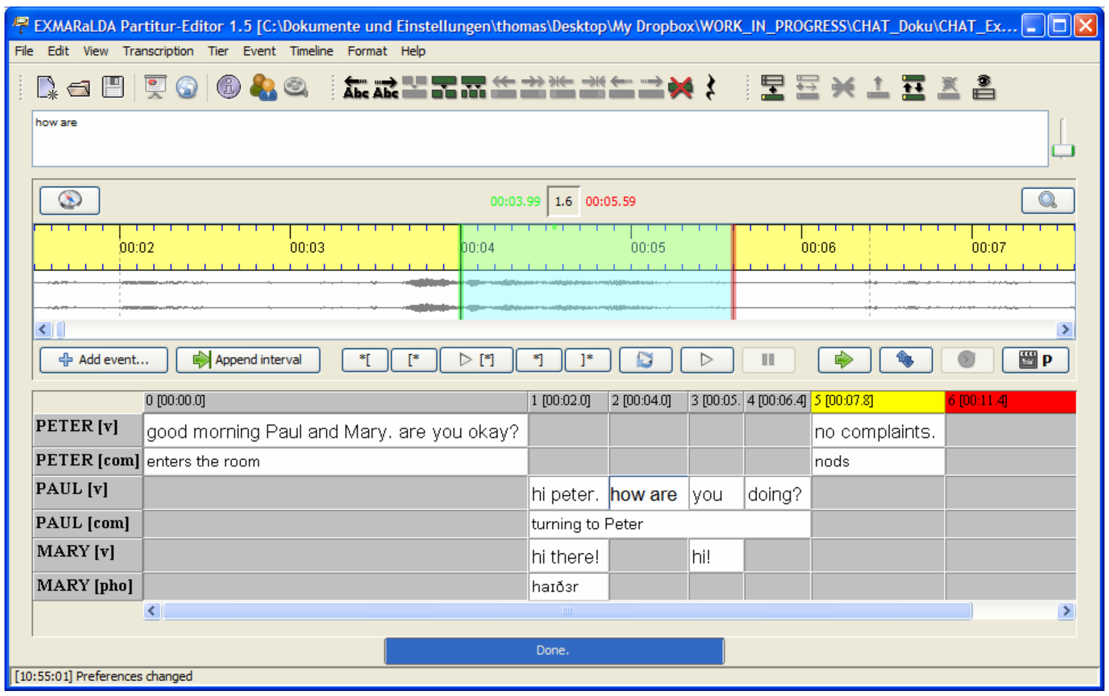
```
@Begin
@Languages:
@Participants:  PETER PETER Participant, PAUL PAUL Participant, MARY MARY Participant
@ID:    en|change_me_later|PETER|||||Participant||
@ID:    en|change_me_later|PAUL|||||Participant||
@ID:    en|change_me_later|MARY|||||Participant||
@Media: royal_exemple, audio
*PETER: good morning Paul and Mary.  •
%xcom:  enters the room
*PAUL:  hi peter.  •
%xcom:  turning to Peter
*MARY:  hi there!  •
%xpho:  haɪðɜr
*PAUL:  how are you doing?  •
%xcom:  turning to Peter
*PETER: no complaints.  •
%xcom:  nods
@End

23apr10[E|CHAT]   1
```

Figure 3: … and corresponding exported CHAT transcript in CLAN (export based on events)

Note that, in the exported CHAT file:

- Each event from the v-tiers in the source file has become one utterance in the target file. This is because the v-tiers have been assigned the type ‘T(ranscription)’
- Each event from the other tiers in the source file ends up in a dependent tier. This is because all other tiers have been assigned the type “A(nnotation)”
- The comment ‘turning to Peter’ occurs only once in the source, but twice in the target file. This is because it coincides with two events in a tier of type ‘T’ and hence with two utterances in the target file.
- Overlaps (as between Paul’s first and Mary’s utterance) are not marked explicitly, but represented through identical bullets for the respective utterances.

For many purposes, the export based on events is the best – and in any case the simplest – option. However, note that EXMARaLDA does not require you to transcribe one utterance per event. You can be both less precise (by transcribing several utterances in one event) and more precise (by distributing one utterance across several events). The latter is especially important if you want to mark the extension of speaker overlap in a very detailed manner. For cases where the rule “One utterance per event” does not hold, you may want to try the CHAT export **Based on CHAT segmentation**. This method will attempt to determine utterance boundaries according to the CHAT segmentation algorithm (see **How to use segmentation** for further details), i.e. it will look for symbols defined by CHAT as utterance terminators (such as the period or the question mark) and thus decide which parts of the source transcription to transform into an utterance in the target transcription. The following example illustrates this:



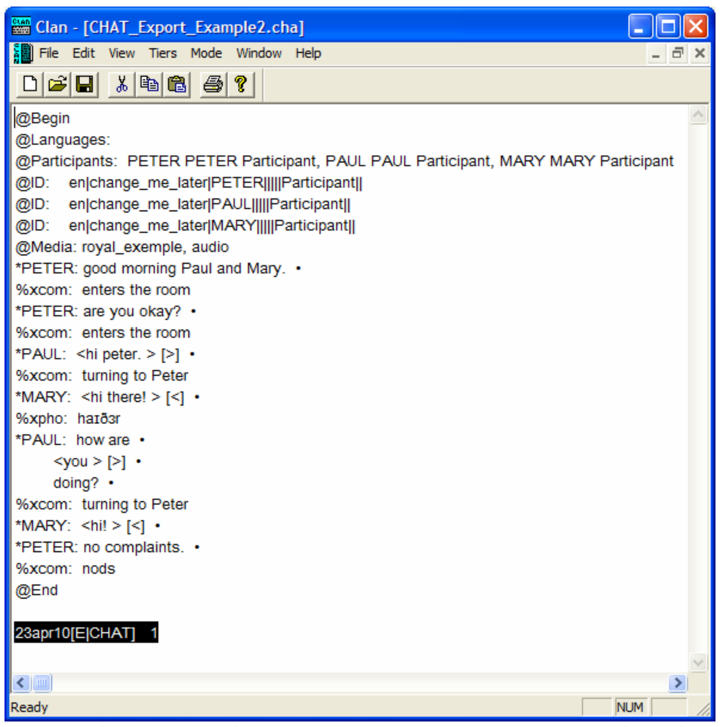
EXMARaLDA Partitur-Editor 1.5 [C:\Dokumente und Einstellungen\thomas\Desktop\My Dropbox\WORK_IN_PROGRESS\CHAT_Doku\CHAT_Ex…

how are

	0 [00:00.0]	1 [00:02.0]	2 [00:04.0]	3 [00:05.	4 [00:06.4]	5 [00:07.8]	6 [00:11.4]
PETER [v]	good morning Paul and Mary. are you okay?					no complaints.	
PETER [com]	enters the room					nods	
PAUL [v]		hi peter.	how are	you	doing?		
PAUL [com]		turning to Peter					
MARY [v]		hi there!		hi!			
MARY [pho]		haɪðɜr					

[10:55:01] Preferences changed

Figure 4: Source transcription in the Partitur-Editor (slightly modified version of the above)…



Clan - [CHAT_Export_Example2.cha]

```
@Begin
@Languages:
@Participants:  PETER PETER Participant, PAUL PAUL Participant, MARY MARY Participant
@ID:    en|change_me_later|PETER|||||Participant||
@ID:    en|change_me_later|PAUL|||||Participant||
@ID:    en|change_me_later|MARY|||||Participant||
@Media: royal_exemple, audio
*PETER: good morning Paul and Mary.  •
%xcom:  enters the room
*PETER: are you okay?  •
%xcom:  enters the room
*PAUL:  <hi peter. > [>]  •
%xcom:  turning to Peter
*MARY:  <hi there! > [<]  •
%xpho:  haɪðɜr
*PAUL:  how are  •
        <you > [>]  •
        doing?  •
%xcom:  turning to Peter
*MARY:  <hi! > [<]  •
*PETER: no complaints.  •
%xcom:  nods
@End

23apr10[E|CHAT]   1
```

Figure 5: … and corresponding exported CHAT transcript in CLAN (export based CHAT segmentation)

Note that, in the exported CHAT file:

- The first event “good morning Paul and Mary. are you okay?” is transformed into two utterances in the exported transcript because it contains two utterance terminators.
- The three events “How are”, “you” and “doing?” are transformed into one utterance in the exported transcript because only the last of them contains an utterance terminator.
- As in the above example, comments and phonetic annotations end up in dependent tiers of the CHAT file.
- Overlaps (as between Paul’s and Mary’s utterances) are now marked explicitly.

The CHAT export **Based on HIAT segmentation**, finally, works analogous to the export based on CHAT segmentation except that it determines utterance boundaries according to the HIAT transcription system.

3. Importing CHAT data into EXMARaLDA

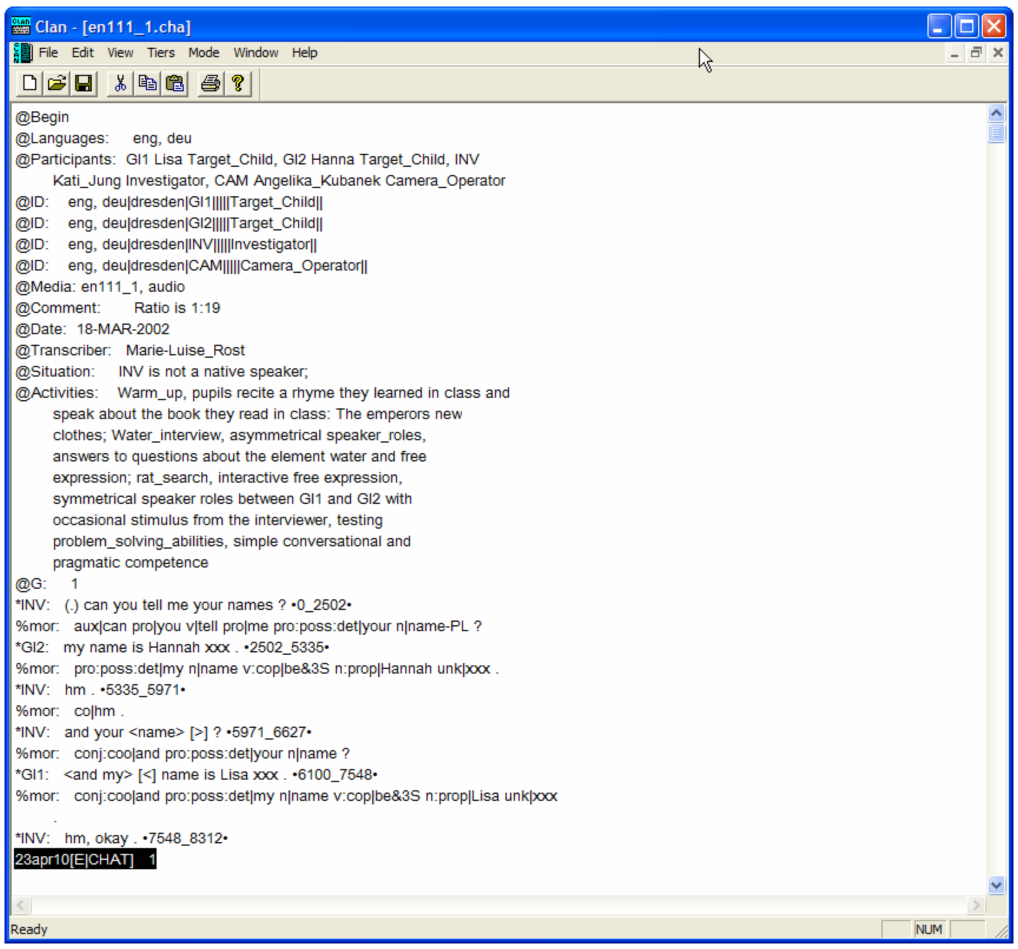
In order to import a single CHAT file into EXMARaLDA you can use the Partitur-Editor's **File > Import** menu item. In order to import a whole corpus, it may be more convenient to use the **CHAT corpus wizard** inside EXAKT.
EXMARaLDA's CHAT import has been designed to be robust. That means that, even if it encounters a problematic structure in a CHAT file, it will still try to transform it rather than throw an error message.

The most common problematic structure in a CHAT file is the case where temporal assignments are not sufficiently precise (or simply wrong) and contradict one another. Consider, for example, the following excerpt from a transcript in the CallFriend corpus:

```
*F1:  ⌈·hhhh⌉→ *1070688_1071520*
*F2:  ⌊·hhhh⌋→ *1070688_1071520*
*F1:  °sur° pris ingly enough: n:o::→ *1071360_1072704*
```

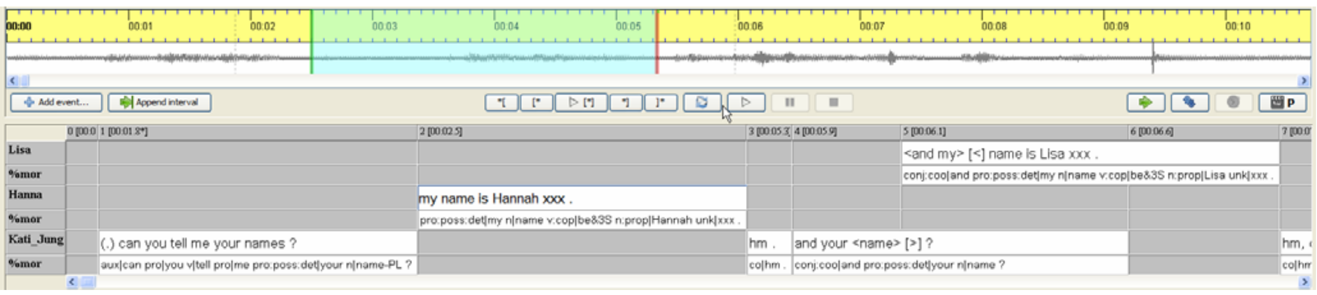
Here, the timing information contained in the bullets says that the second utterance of speaker F1 starts before his previous utterance ends. This kind of "self-overlap" is not a problem inside CLAN because CLAN has no explicit timeline. When importing the file into EXMARaLDA, however, the events corresponding to the overlapping utterances cannot go into one and the same tier because EXMARaLDA does not allow events in a tier to overlap. When importing such a file, the events will therefore be distributed across two tiers. To avoid this, you can either fix the cause of the problem inside the CHAT file (in this case by setting the start point of F1s second utterance to ***1071520***), or you can fix it after the import inside EXMARaLDA (in this case by cutting and pasting the text of the second utterance to an appropriate place into the tier which contains the first utterance.

Apart from problems of this kind, importing CHAT into EXMARaLDA is straightforward: utterances on main tiers will end up in events of tiers of type 'T(ransription)', entries on dependent tiers will go into events of tiers of type 'A(nnotation)', and the timing information will be extracted and put into the corresponding timeline item(s). EXMARaLDA will also try to find and link the media file(s) specified in the CHAT file. If this does not work, media files can be relinked after the import via Transcription > Recording in the Partitur-Editor.
For example, the CHAT file in figure 6 (taken from Talkbank > BilingBankDresden) will be transformed into the EXMARaLDA file in figure 7.



```
@Begin
@Languages:    eng, deu
@Participants:  Gl1 Lisa Target_Child, Gl2 Hanna Target_Child, INV
       Kati_Jung Investigator, CAM Angelika_Kubanek Camera_Operator
@ID:    eng, deu|dresden|Gl1|||||Target_Child||
@ID:    eng, deu|dresden|Gl2|||||Target_Child||
@ID:    eng, deu|dresden|INV|||||Investigator||
@ID:    eng, deu|dresden|CAM|||||Camera_Operator||
@Media: en111_1, audio
@Comment:       Ratio is 1:19
@Date:  18-MAR-2002
@Transcriber:   Marie-Luise_Rost
@Situation:     INV is not a native speaker;
@Activities:    Warm_up, pupils recite a rhyme they learned in class and
       speak about the book they read in class: The emperors new
       clothes; Water_interview, asymmetrical speaker_roles,
       answers to questions about the element water and free
       expression; rat_search, interactive free expression,
       symmetrical speaker roles between Gl1 and Gl2 with
       occasional stimulus from the interviewer, testing
       problem_solving_abilities, simple conversational and
       pragmatic competence
@G:     1
*INV:   (.) can you tell me your names ? •0_2502•
%mor:   aux|can pro|you v|tell pro|me pro:poss:det|your n|name-PL ?
*Gl2:   my name is Hannah xxx . •2502_5335•
%mor:   pro:poss:det|my n|name v:cop|be&3S n:prop|Hannah unk|xxx .
*INV:   hm . •5335_5971•
%mor:   co|hm .
*INV:   and your <name> [>] ? •5971_6627•
%mor:   conj:coo|and pro:poss:det|your n|name ?
*Gl1:   <and my> [<] name is Lisa xxx . •6100_7548•
%mor:   conj:coo|and pro:poss:det|my n|name v:cop|be&3S n:prop|Lisa unk|xxx
       .
*INV:   hm, okay . •7548_8312•
23apr10[E|CHAT]   1
```

Figure 6: CHAT original...



	0 [00:0	1 [00:01.8*]	2 [00:02.5]	3 [00:05.3(	4 [00:05.9]	5 [00:06.1]	6 [00:06.6]	7 [00:0
Lisa						<and my> [<] name is Lisa xxx .		
%mor						conj:coo|and pro:poss:det|my n|name v:cop|be&3S n:prop|Lisa unk|xxx .		
Hanna			my name is Hannah xxx .					
%mor			pro:poss:det|my n|name v:cop|be&3S n:prop|Hannah unk|xxx .					
Kati_Jung		(.) can you tell me your names ?		hm .	and your <name> [>] ?			hm,
%mor		aux|can pro|you v|tell pro|me pro:poss:det|your n|name-PL ?		co|hm .	conj:coo|and pro:poss:det|your n|name ?			co|hm

Figure 7: ...and corresponding EXMARaLDA file after import