

Understanding the basics of EXMARaLDA

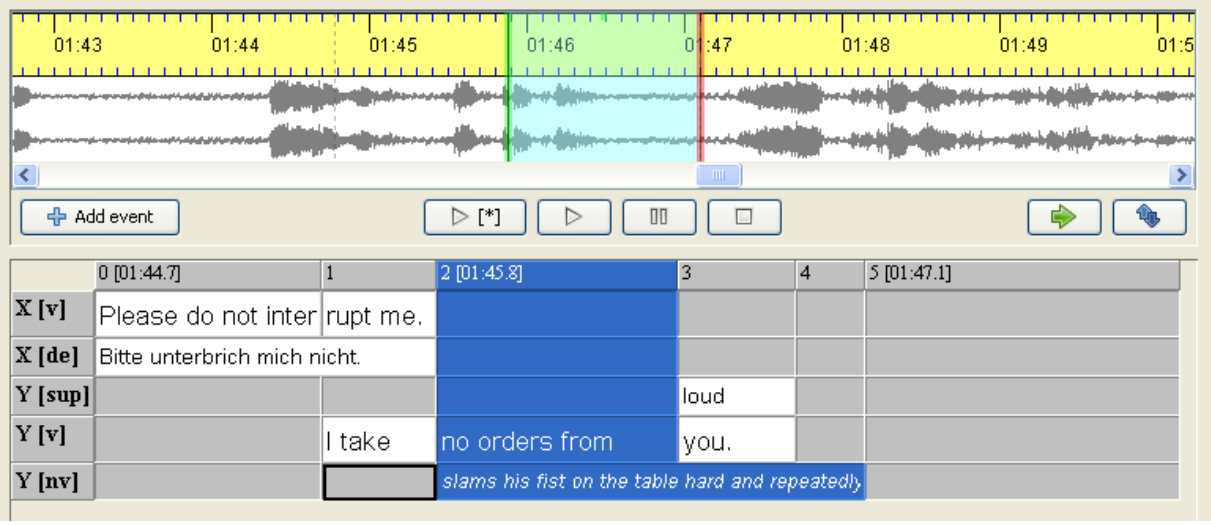
This document explains the basic concepts of the EXMARaLDA system and the terminology used in the tools' menus and dialogs and in all other documentation.

Contents

[System version 1.5.1.]

A. Transcriptions

An EXMARaLDA transcription is a well-defined structure consisting of a few basic entities which are put in relation to one another. In order to explain the basic entities of an EXMARaLDA transcription and their relationships, we will use the following short transcription as an example:



1. Events, timeline, tiers and speakers

The fundamental unit in an EXMARaLDA transcription is the **event**. An event contains a piece of text describing something that happened in the transcribed recording.

The above example contains altogether eight events – the white cells in the partitur interface. Five of these describe words (or word parts) uttered by the two speakers ('Please do not inter', 'rupt me. [1]', 'I take ', 'no orders from ' and 'you. '). One event ('slams his fist on the table hard and repeatedly') describes nonverbal behaviour. The remaining two events contain a German translation of the X's utterance ('Bitte unterbrich mich nicht') and a suprasegemental characterisation ('loud') of the word 'you' uttered by speaker Y.

Each event is assigned a place in the transcription through its reference to the **timeline** and a **tier**.

The **timeline** is simply an ordered sequence of time points. Each time point can be assigned to an absolute time value which is interpreted as a pointer into the corresponding audio or video recording. The above example has a timeline containing altogether six time points – the grey, numbered cells in the top row of the partitur interface. Three of these time points (0, 2 and 5) are assigned to absolute time values ('01:44.7', '01:45.8' and '01:47.1', respectively). Assigning an event to the timeline means specifying a start and an end time point for it on the timeline. For instance, the marked event labelled 'no orders from ' starts at time point 2 and ends at time point 3. The event labelled 'slams his fist...' also starts at time point 2 but ends later at time point 5 and thus stretches over a longer interval. Since time point 2 is assigned an absolute time value, we have the additional information that 'no orders from ' and 'slams his fist on the

[1] For the word boundaries to be recognized as such, use a space after each word.

table hard and repeatedly' start at time '01:45:8' in the transcribed recording. The editor uses this information to select a corresponding stretch in the waveform view of the recording.

A **tier** assembles events with similar properties, typically all events describing the same type of action (e.g. verbal or nonverbal) of one speaker. The above example contains altogether five tiers – the rows after the top row of the partitur including tier labels in the grey, leftmost cells (labelled 'X [v]', 'X [de] etc.).

Each tier in turn can be assigned to a **speaker**. Speakers are assembled in a speakertable. The speakertable of the above example contains two speakers labelled 'X' and 'Y'.

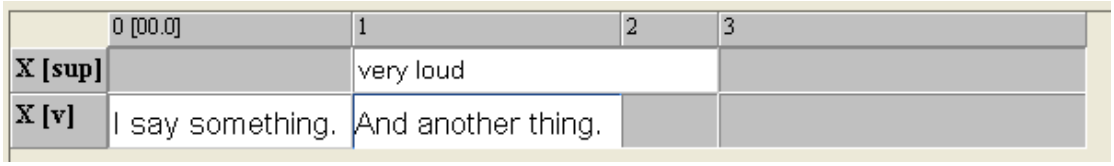
2. Tier types and categories

Besides the (optional) speaker assignment, each tier also gets an assignment to a **type** and a **category**. These assignments are very important for many automatic processing steps, so it is crucial that you use them correctly if you want to do reliable searches and transformations with your data.

Categories specify the exact type of information described in the events of a tier. In the above example, the two tiers with events describing verbal behaviour have the category 'v', the other three tiers have the categories 'de', 'sup' and 'nv' for 'German (deutsche) translation', 'suprasegemtal' and 'non-verbal', respectively. Categories can be freely chosen (i.e. you might as well label a verbal tier with the category 'verbal'), but you should make sure that category assignment is consistent throughout your corpus (i.e. don't give verbal tiers the category 'v' in one transcription, 'V' in the next, and 'verbal' in another).

Types are predefined classifications for tiers. There are three different types:

- Type **'t'** stands for 'transcription'. This is normally the type for tiers in which verbal behaviour is described. You should have only one tier of this type for each speaker, and you should not have a tier of this type without a speaker assignment. In the above example, the first and the fourth tier are of this type.
- Type **'d'** stands for 'description'. This is the type for tiers in which non-verbal behaviour is described. You can have as many tiers of this type as you need for each speaker (e.g. three if you want to distinguish facial expression, hands or body movement, or none if you do not transcribe non-verbal behaviour). It is also possible to have tiers of this type without a speaker assignment (e.g. for background noises, applause etc.). In the above example, the last tier is of this type.
- Type **'a'** stands for 'annotation'. This is the type for tiers containing additional analytic information about events in tiers of type **'t'**. You can have as many tiers of this type as you need for each speaker (e.g. one for a German translation, one for a morphological transliteration). Unlike the other tier types, annotation tiers are not independent, since annotations depend on something to be annotated, events in tiers of this type must always have a corresponding event (or sequence of events) in a tier of type 't' assigned to the same speaker. This distinction between transcription and annotation is a part of the EXMARaLDA data model. In the above example, the second and third tiers are of this type. Note that for the events in these tiers, there are corresponding events in tiers of type 't' assigned to the same speaker ('Bitte unterbrich mich nicht' corresponds to the two events 'Please do not inter' and 'rupt me ', while 'loud' corresponds to the single event 'you. '). In contrast, the event 'very loud' in the figure below does not have a corresponding set of events in the transcription tier (type 't', category 'v'). Such structures are not allowed and will cause structure errors.



3. Segment chains

Having understood the definition of tiers, events and tier types, understanding **segment chains** is straightforward: a segment chain is defined as an uninterrupted sequence of events in a tier of type 't'. Thus, the above example contains two segment chains – the first ('Please do not interrupt me.') consists of the two events, the second ('I take no orders from you.') of three events. Segment chains can be used to generate non-partitur output formats: if you order all segment chains of a transcription by their start points, you can produce a drama-script-like presentation like the following one:

X: Please do not interrupt me.
Y: I take no orders from you.

Segment chains also play a crucial role for the segmentation of basic transcriptions (see "How to use segmentation").

B. The Structure of Coma Metadata

Coma metadata consists of five predefined data containers: corpora, communications, speakers, transcriptions and recordings. There are also further data types that within these containers. It is important to understand the connection between these containers and data types.

1. Corpora

Corpora are the top-level containers for all other containers and datatypes. Apart from corpus-wide metadata and associated files they consist of the containers on the next level; those for Speakers and Communications.

2. Communications

Communications are used to describe the situation where the transcribed conversations took place and manage all material belonging to this situation. Communications typically feature speakers and there can be recordings and transcriptions of the conversation. In the coma data model, recordings, transcriptions and speakers are linked to communications. Furthermore, all things noteworthy of the communicative situation (time, place and circumstances, languages spoken) are stored with the communication.

3. Speakers

Speakers are – as the name suggests – the persons that participate in the communication. Speakers don't have to be actual persons (automated dialog systems also qualify) and they don't have to actually speak – as long as they are important for understanding what is happening in the conversation, they should be registered. The speaker datatype should contain everything that is important about that speaker, like date and time of birth, language learning history etc. Since speakers can be linked to multiple communications, data that is only

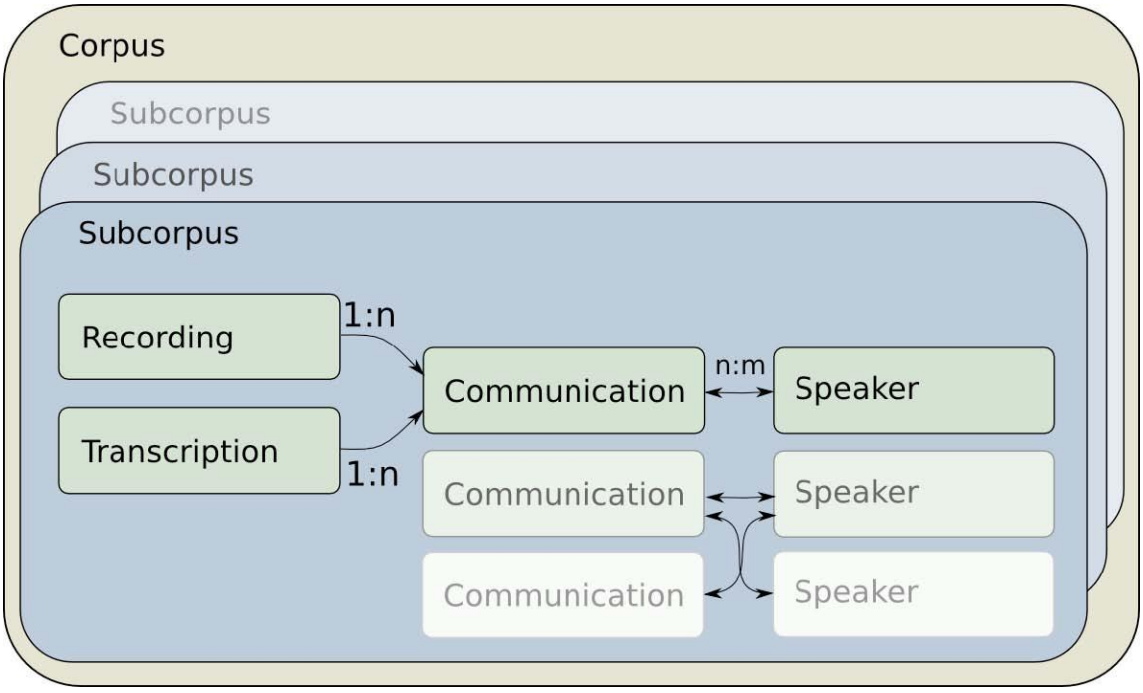
relevant for one communication should not be saved with the speaker, but with the communication.

4. Recordings

Recordings reference to the actual (audio or video) recording of the communication. Recordings are always connected to a communication and cannot exist on their own.

5. Transcriptions

Transcriptions establish the link to actual EXMARaLDA transcription files. Coma manages basic as well as segmented transcriptions. There is an option inside the Coma preferences panel to toggle whether basic transcriptions are to be shown or not, since tools like the EXMARaLDA search tool "EXAKT" only handle segmented transcriptions. Like recordings, transcriptions must always be linked to a communication. They are also linked to the transcribed recordings through the communications.



6. Further Datatypes

To capture actual metadata, further datatypes exist. Two of them are of special importance:

a. Location

Locations represent a location at a certain time.

A location does not have to hold place and time information, but it can: In the example above, one location encodes birth date and location of a speaker, the second location encodes only the location of residence. It is important to remember to use locations even if one only wants to register the time of a special event.

b. Description

Since it is not possible to define a standardized set of metadata fields for all areas of research, most of the metadata in Coma is encoded through free key-value pairs. These pairs are collected inside Description fields. These exist in all Coma data types: There can be descriptions for corpora, for communications, for recordings etc.

The example shows a description belonging to a speaker. Since the keys inside descriptions can be named freely, it is very important to create a unified vocabulary of description keys for corpus metadata. Coma's templates can help to harmonize descriptions and simplify their input.