The Potsdam Dialogue Corpora
Transcription and Annotation Manual

David Schlangen *and* Raquel Fernández
`das|raquel@ling.uni-potsdam.de`
Universität Potsdam
Germany

Contents

1 Preface

This manual documents and collects the various instructions and guidelines given to and used by the transcribers and annotators of the experiments / data collections performed in 2006/07 as part of the projects "DEAWU" (funded by the EU, Marie Curie Transfer of Knowledge) and "InPro" (funded by DFG in the Emmy Noether Program) at the University of Potsdam. Note that while close to what the annotators and transcribers actually used, the material in this document has been edited and cleaned up and so is not a completely faithful representation of this.

If this document is read as a description of the actual annotation of the DEAWU data, please keep in mind that not all levels are annotated for all of the DEAWU corpora. Consult the Experiment Documentation for an overview of what is available for which corpus. Also note that the annotation levels were created successively for the experiments and for slightly different purposes each time; this explains why some levels are partially overlapping in terms of information they encode. Finally, note that these instructions are meant to cover both the German language transcriptions / annotation as well as the English language ones; where distinctions need to be made they will be noted in the text.

The annotation schemas described in this manual have been devised by David Schlangen and Raquel Fernández, the final editing of this document has been done by David Schlangen, with lots of help by Janine Wolf.

2 Transcription

2.1 Preparations for Transcription

2.1.1 Basics

We use `Praat` (Boersma 2001) for the transcription work. `Praat` is a program for phonetic analysis but also works well for transcribing speech. `Praat` is free and there are Versions for Mac, Windows, Linux, and various UNIXes.

The audio file to be transcribed needs to be in WAV or AIFF format. You should have been provided with files in one of these formats. If however your file is in mp3 format, on Mac you can use `iTunes` to convert it to AIFF; on Windows there are a lot of free converters, for example `Mp3Decode`. On Linux Systems you can use `mpg123` for this step.

To start transcribing, you need to load the audio file into `Praat`. It can be useful to load the audio file in two different ways: both channels as one mono file (`Read→Open long sound file...`) and the two channels separately (`Read→Read two sounds from stereo file...`). This way you can choose to listen to both channels together or to either of them separately, depending on what you find easier for the transcription at each point. For instance, when the speakers overlap, you may find it easier to listen to them separately to transcribe the utterances; while to decide on turn boundaries you need to play both channels at the same time.

The transcription will be stored in a separate file, henceforth this file is called the `TextGrid` file. You will either have been provided with pre-configured `TextGrid` files or will have to create your own; see next section. Don't forget to save frequently while you are working.

The tasks at this step are a) to *segment* the contributions of each speaker into underlineutterances (see Section 2.3.1 below), b) to *transcribe* these utterance (Section 2.2), and c) to *segment* the contributions (or, equivalently, to group the utterances) into turns (Section 2.3.2).

2.1.2 Transcription Tiers / Segmentation Levels

If you have been provided with a `TextGrid` for your transcription task, load it. (Load the audio file as described above, load the `TextGrid` with `Read→Read from file...`, select both the audio object and the `TextGrid` object and click `Edit`.)

If not, create a `TextGrid` object and tiers as follows. Select the audio object, click `Annotate→To TextGrid`, enter the following tier names into the field `Tier names: E-utts E-turns P-turns P-utts`, leave the field `Point tiers` empty. Now select both the audio object together with the new `TextGrid` object and click `Edit`. (Note: In new transcriptions the tiers should

be named IF-utts, IF-turns, IG-turns, IG-utts; we changed our terminology from *Executor* to *Instruction Follower* and from *Player* to *Instruction Giver*.)

Either way, you should now have open a `TextGrid` that contains a minimum of 4 tiers:

1. E-utts: tier used for segmentation and transcription of IF utterances
2. E-turns: tier containing the IF turn boundaries
3. P-turns: tier containing the IG turn boundaries
4. P-utts: tier used for segmentation and transcription of IG utterances

Tiers 1 and 4 are used to transcribe utterances. In these tiers transcribers need to mark utterance boundaries, and then for each utterance interval transcribe the utterance in question (see the guidelines below).

Tiers 2 and 3 are used to mark turn boundaries. Inside the turn intervals, just write an `E` when the interval corresponds to the IF's turn in tier 3, and a `P` when it corresponds to the IG's turn in tier 4. (I.e., don't repeat the textual transcription here.) See the turn segmentation guidelines below.

Marking utterance and turn boundaries should be done as precisely as possible.

Noise Condition (`NP` and `ND`) In some of the spoken dialogues to be transcribed, there is occasional additionally noise in the IG speech channel. (The sound files you have been given will contain the noise mixed into the audible speech for you to be able to hear what was said; the original listener will only have heard the noise.) The TextGrid files of such dialogues contain an extra tier:

5. Noise: tier containing noise intervals

This tier is created automatically and just for your information (e.g., you can use it to click on the marked noise intervals to listen to exactly this portion). Nothing should be entered into this tier.

2.2 Transcription Guidelines

Please conform to the following guidelines when transcribing utterances.

Tags. The following tags can be used where appropriate:

Tag	Description	Example
`<self> </self>`	self directed talk	`<self>`*hm how can I do this*`</self>`
`<laughter/>` or `<laughter></laughter>`	laughter	`<laughter>`*yeah*`</laughter>` *that's cool*`<laughter/>`
`<unclear/>`	unclear material, left out	*I'm* `<unclear/>` *this*
`<unsure></unsure>`	inconfident transcription	*to* `<unsure>`*the side*`</unsure>`
`<german></german>`	German word(s) (in English dialogue)	*yes* `<german>`*genau*`</german>`

Phonetic issues. The following symbols can be used where appropriate:

Symbol	Description	Example
:	lengthening	*eh:m:*
!	rising intensity	*come on !*
?	rising tone	*do you mean this one ?*
_?	gap intonation	*take the _?*

Fillers. When transcribing filler sounds, please use the transcription that seems most adequate from the following list:

`uhm / uhu / mhm / eh / ehm` (for English);
`äh / ähm / hm` (for German)

Pauses. To transcribe pauses, please follow this rule of thumb:

`. (0-0.33 sec) / .. (0.33-0.66 sec) / ... (> 0.66 sec)`

For example: *I mean . on top .. of the other one*

Dysfluencies. Speech dysfluencies are transcribed according to this scheme:

`( LHS { Editing Term } + RHS )`

where the editing term is optional, and the RHS is not present in a restart. For instance:

`I went to ( the:  { um } + the ) garden`
`There ( is a + ) is a house`

Transcription of Noise For the dialogues in the noise condition, the transcription of the IG's utterances (done in tier 4) should include the extent of the noise intervals. (The audio files you have been given will have on IG's channel a mix of what IG said–i.e., before the noise–and the noise as heard by IF; for you to be able to judge the extent of the noise.)

To denote the extent of the noise intervals within the transcription, transcribers should use square brackets as a convention: left square bracket '[' when the noise starts and right square bracket ']' when the noise ends. (Of course beginning and end of noise can be in the middle of a word or in short silences between words, in which case you should "round" to the nearest word.) For example:

```
it looks like [a stairc]ase to me
```

Where `[a stairc]` is speech within a noise interval. Tier 5 can be very useful to decide where to put the square brackets: transcribers can select and play a noise interval in this tier to hear exactly which bit of speech is overlapping with noise.

2.3   Segmentation Guidelines

2.3.1   Utterances

Our notion of utterance basically follows that of the *slash unit* of (Meteer & Taylor 1995): "A slash-unit is maximally a sentence but can be a smaller unit. [. . .] Intuitively, slash-units below the sentence level correspond to those parts of the narrative which are not sentential but which the annotator interprets as complete." (See therein for more details.)

Some general notes:

– The general principle to follow is: segment more rather than less.

– Cut off leading "okay", "right", "so"s, first proper sentence is definitely an utterance. If there were several "okay", "right"s etc., group them according to intonatory boundaries.

– Leading "so": if it can be replaced by "hence", don't segment it off. (And so on—if you can paraphrase particles, they belong to the sentence.)

– Also separate leading "e:m".

– "e:m" is part of utterance if it comes within a coherent syntactic structure (modulo dysfluencies).

– "e:m" at end of abandoned utterance is also part of that utterance.

– Interruption by parenthetical utterance: all parts are individual utterances.

– The only exception of "split off as much as possible" principle: restarts are part of their utterance, are marked up with dysfluency markup.

2.3.2 Turns

All (more or less) continuous stretches of speech by one speaker until other speaker takes over (some pauses are allowed within the stretch) are one turn. Exception: if it really sounds as if the speaker finishes and waits for other DP, and only resumes speaking after a silence in which other DP does not start speaking, this should be annotated as two consecutive turns by the same speaker.

The following is an example of a single turn with four utterances (stars indicate turn boundaries, pipes utterance boundaries):

```
*|Yeah, that's right| |Now {I've got this +} . the shapes I've
got here| |The bottom right hand side . I think it's the
elephant's leg| |I've got this el shape that is an el shape
facing the wrong way|*
```

The next example shows two consecutive turns by the same speaker:

```
*|The longest part is horizontal, the three blocks| |{And then
it's got e:m +} the first block of these is got a block on top
of that . and then a block to the left of that block|* *|So|*
```

To delay the decision whether something is a backchannel or not (and hence, a proper turn or not), we mark even overlapping "yeah", "right"s as forming their own turn; we can later (at a later annotation or analysis step) have a rule that excludes turns consisting solely of an utterance labelled BC.

3 Annotation

For annotation, we use MMAX2 (Müller & Strube 2006). MMAX2 is text-based; the transcriptions and segmentations created in the previous step form the basis for the so-called *MMAX annotation projects* (more on this below). However, please also make use of the audio and video files you will have been given for each dialogue, as some of the annotation levels need information that is not present in the transcription. (E.g., intonation, information about non-linguistic actions performed.)

3.1 Use of the Annotation Tool; Terminology; Notes

Annotation Project You will have been given an *MMAX annotation project* for each corpus you will be annotating. You will only need to be able to locate the *dialogue files* (in `Dialogues/`; see next step), but for your information: the project consists of a *tokenisation* of the transcript into individual words (`Basedata/`), various schemes defining the annotation (`Schemes/`) and

display on the screen (`Styles/`), annotations files (`Markables/`; just dummy files before you start annotating; this is where your actual annotations will be stored).

Basic Operations for Annotation   Here are the steps to follow / the basic operations for annotation.

1. Load file: `file→load`, select the dialogue to annotate from the directory `Dialogues`.

2. The first time you load a file, MMAX will ask you whether you want to validate it. You can click `validate now`. This will take a while. It inserts the default values for all features.

3. Now do the actual annotation, by first creating the *Markable*.

   a) Terminology: *Markable* is the name for the span of text (= stretch of transcribed speech) that is being annotated. The markables are separate for each *annotation level* (= roughly, for each phenomenon that is being investigated, and for which annotation features are defined in `Schemes/`).

   That is, annotation is always an annotation of some span of text that must be defined; i.e., there are two basic steps to the annotation task: i) definition of the span that is being annotated, and ii) the actual annotation of it. Some *Markable*s (and hence, annotation levels) come pre-defined (see next section) and some you have to define yourself.

   b) If you have to create the markable yourself:

      i. Select a span of text with the mouse, just like you would in a text editor. (I.e., by clicking and dragging the mouse along the span.)

      ii. You will then be prompted with a choice `Create Markable on level ..` and a list of the annotation levels that are defined in `Schemes/`. (As will be explained in more detail below: Do not create markables on levels utterances and turns; these come pre-defined!)

      iii. It is possible to add to already defined markables, and even to create discontinuous markables. To do this, simply select an existing markable (see below) and then do another selection step as described above, but while holding down the shift key. This will prompt a question *Add to this markable?*.

   c) If you made a mistake when defining a markable, you can delete it as follows: right-click on a word belonging to the markable and then select `delete this markable`.

4. With the markable created, you can do the actual annotation.

   a) Select a markable:

      i. click on a word that is contained within the markable (you have to do this even for markables you've just created)
      ii. you will be presented with a choice of markables to which this word belongs. (If it belongs to only one, this step is skipped.) Select one.
      iii. Now the span of the markable is highlighted in yellow.

   b) With the markable selected, go to the annotation window. The tab corresponding to the level you want to annotate should be selected. Annotate. (See next sections for the meaning of the features.)

   c) Some features create a link between markables, they are annotated slightly differently: Select the markable as above, but then select another markble *with the right mouse button*, this will present you with a choice of link features (e.g., *Mark as CR antecedent?*).

5. Save. And save again. And don't forget to save.

6. The files that contain your annotations are in the directory `Markables/`, beginning with the name of the dialogue you annotated, and ending with "markables.xml". (One file for each markable level.) Please send these to us when done.

General remarks on annotating: Please keep a log of what you're doing, if you note anything interesting, are unclear about which value to chose for a feature, disagree with a definition / classification, etc. just write down filename and position (# of utterance, a few lines of the utterance to find it, perhaps even position in audio file (as praat gives it)).

Conventions for Feature Descriptions in Next Sections   In the following, combinations of feature and value will be printed as `feature:value`.

Some features are conditional on the values selected for other features. E.g., the fine-classification of clarification requests is only shown (in some annotation schemes) for `is_cr:yes`. This is not always explicitly noted in the descriptions in the next Sections.

In the examples for the meaning of features / values below, sometimes some context needs to be shown. The convention is that the markable that is being annotated is printed in *`slanted typewriter font`*, whereas the context is printed in `normal typewriter font`.

Display Styles   You can display the dialogues in different styles. The default is `utts.xsl`, which displays in each line one utterance, beginning with the start time of the utterance, the end time, the speaker and utterance ID, and finally the text of the utterance itself.

An alternative style, to be selected in the *Markable level control panel* under `Settings→StyleSheet` is `moves.xsl`, which additionally marks move boundaries (see below Section 3.4) with lines.

In all styles, clarification requests (i.e., utterances with `is_cr:yes`) are highlighted with a green background, utterances with noise (`noise_found:yes`) with a yellow background.

3.2 Utterance Level

The annotation on this level will be used for an analysis of the minimal actions performed in the dialogue, and especially to investigate the effect of the different manipulations / conditions (independent variables).

3.2.1 Identifying Markables

Utterance boundaries were added at the transcription stage (see above) and automatically transferred to MMAX markables. *Please do not define any markables on this level!*

3.2.2 Features

The values for the features `number` (identifier for utterance), `speaker`, `start` (start time), `end` (end time), and `noise_found` ("was there noise within this utterance?", only for noise corpora) are determined automatically by the transformation script. *Please do not change any values in these fields!*

All following features are to be annotated manually. Please note that not all of these features are defined in the annotation scheme for all corpora; for example, the features referring to (the artificially added) noise are of course only present in the noise corpora. You need to annotate only those features that are presented to you in the annotation window.

Clarification Requests

`first_or_last: yes` for first and last utterance of the task-related part of dialogue, `no` otherwise.

`what_was_noisy:` with values `part of word`, `whole word`, `whole phrase`, `whole utterance`

how_much_was_noisy: with values `less_than_ten` (percent), ..., `everything`.

> I.e., `what_was_noisy` gives a measure of "damage" by noise w.r.t. syntactic units, whereas this feature measures damage relative to the whole utterance.

where_was_noise: rough placements within utterance, `first_third`, `second_third`, `last_third`. The noise-related features are only available if there was indeed noise in the utterance (i.e., `noise_found:yes`).

is_cr: boolean, is this utterance a clarification request? Clarification requests (CRs) are utterances that point out understanding problems in the widest sense (e.g., hearing problems, semantic problems etc.) of previous utterances and ask for repair.

> Example: A: I saw Peter. B: *Peter? / Who? / What?*

The following six features are only available for `is_cr:yes`. They are taken from (Rodríguez & Schlangen 2004), with some modification as noted below.

cr_mood: This is about the sentence "mood", either syntactically encoded or intonationally. (Hence, to annotate this feature, the audio of the dialogue must be consulted.) Note that this is really about the *form* of the CR and not about the denoted semantic object. E.g., `cr_mood:decl` can very well denote a polar question.

> **decl:** Declarative word order; or, more precisely, non-interrogative word order (as it covers fragments / phrases as well). Rising boundary tone.
> Example: A: I saw Peter. B: *Peter?*
> Example: A: I saw Peter. B: *You saw Peter?*
> Example:
> A: If you could put that on the elephant's back [foot]?
> B: *The back foot?*
> A: Yeah.

> **decl-f:** As above, but with falling boundary tone.
> Example:
> A: Genau, wo der Pfeil jetzt ist...das Ding.
> B: *Das da?*
> A: Ja.

> **polq:** The CR is a polar question, i.e., a yes-no-question.
> Example: A: I saw Peter. B: *Did you say you saw Peter?*

> **wh_q:** The CR is a wh-question.
> Example: A: I saw Peter. B: *Who?*

Example:
A: `That fits directly under the last step.`
B: *Under the what?*
A: `Last step.`

alt_q: The CR is an alternative question.
Example: A: `I saw Peter.`   B: *Peter Miller or Peter Smith?*
Example:
A: `Uhm, you need to get it so...`
B: *The long one or the ?*
A: `Yah you want the long ...  and the short b.`

gap: The CR has a gap intontation (elongated vowel, flat mid-level tone). This value is not in (Rodríguez & Schlangen 2004).
Example: A: `I saw Bo.`   B: *B[o:]* ⎯ (with the intuitive interpretation of asking for the last name).
Example:
A: `Should meet the t[op] of the T shape.`
B: *The top of the _?*
A: `Not the top of the T shape, the bottom of the T.`

imp: The CR is an imperative sentence.
Example:
A: `And the [long part] is on the [le]ft.`
B: *Say that again.*
A: `The long part [of the] T is on the left.`

other: Rubbish-bin category, if none of the above applies.

cr_form: Syntactic completeness w.r.t. notion of full syntactic sentence (with tensed verb).

complete: Syntactically complete utterances (i.e., utterances with tensed verb).
Example:
A: `The next one...I would d[escri]be as a W shape [in the f]ont tha[t] you [write "Wario"] in.`
B: *You have to start again, it keeps cutting up.*
A: `It's basically like a W shape.`

partial: Syntactically incomplete utterances; utterance fragments.
Example:
A: `That original [piece?]`

B: *The original but which?*
A: [You kno]w the original?   There's a piece [that's a-]

particle: This feature value denotes conventional phrases like "pardon?", "what?", "huh".
Example:
A: And it's a [T shape, you're] looking for.
B: *Sorry?*
A: A T shape, the letter T.

cr_rel_ant: This feature annotates the relation of the CR to its antecedent.

repetition: Parts of the antecedent utterance are repeated.
Example: A: I saw Peter.   B: *Peter?*
Example:
A: Nimm mal zuerst das Teil, was in der unteren Reihe das Dritte von links is, ne?
B: *Das Dritte von links.*
A: Ja genau.

reformulation: The CR contains a reformulation of (parts of) its antecedent.
Example: A: I saw Peter.   B: *My cousin?*
Example:
A: Hier unten rechts, hier...ja genau, rechts, rechts!
B: *In die Hinterbeine.*
A: Nee, nen Stck nach...genau, in die Hinterbeine.

addition: The CR does not only contain the linguistic material from the antecedent utterance, but also new material. (Added wh-words do not count here, they have their own feature, see below.)
Example: A: I saw Peter.   B: *Peter Miller?*
Example:
A: Dis is ahm denk ich mal in der unteren Reihe das Dritte von links.
B: *In der unteren, in der untersten waagerechten?*
A: Achso, nee.

add-wh: Only wh-words have been added.
Example: A: I saw Peter.   B: *Which Peter?*
Example:
A: There should have been a [normal one]?
B: *There should have been what?*

A: Just a straight one?

independent: The CR is formulated independently of the antecedent
  utterance.
  Example: A: I saw Peter. B: *Who is that? / Who?*
  Example:
  A: You know the second piece which is [the front leg].
  B: *Sorry...you're cutting out there?*
  A: You know the [second pie]ce which is the front leg.

cr_extent: Boolean feature: does the CR point out the exact part of the
  antecedent utterance that was problematic or not.

  no: The CR does not point out the problematic part of the antecedent
    utterance. (Usually, CRs with **cr_form particle** do not point
    out any exact part at all.)
    Example: A: I saw Peter. B: *Pardon?*
    Example:
    A: So the end of the [Z] should be righ[t] on the bott[o]m
    the grid.
    B: *Pardon?*
    A: The end of the Z should be facing right down to the bottom
    of the grid.

  yes: ftexI saw Peter.Who is that? / Who?
    Example:
    A: And so the little square that is sticking out faces down.
    B: *Faces down?*
    A: So it's horizontal.

cr_severity: Does the CR present a hypothesis about antecedent or not?

  no hypothesis: The CR does not present any hypothesis about what
    the problematic part might have been.
    Example: A: I saw Peter. B: *Pardon?*
    Example:
    A: So, we need that [touching corners with] that trunk bit.
    B: *I need to do what?*
    A: You need it touching corners with the trunk bit.

  hypothesis presented: The CR does present some hypothesis about
    the antecedent utterance.
    Example: A: I saw Peter. B: *My cousin?*

Example:
A: `Oh sorry, the [sixth] row down.`
B: *The second row down yeah?*
A: `Sixth row down.`

hypothesis demonstrated: This value is only available in the Vis-Pento dataset, where there was a visual channel that enabled a 'trial and error' strategy for placing the pieces. This feature accounts for CRs like "Here?" or "Is this (points) right?".
Example:
A: `Genau da hin.`
B: *Ganz...da?*
A: `Du machst das perfekt.`

answer_to_cr: How did the original speaker reply to the CR? (Note that this is annotated on the CR itself, i.e., for this feature one needs to look ahead in the dialogue. E.g, in the examples below, the CR is printed in italics, but the feature value depends on the utterance that follows the CR.)

elaboration: The utterance or part of the utterance that caused the problem is neither repeated nor reformulated, but elaborated.
Example:
A: `It'[s a bit has] an e[longat]ed bottom.`
B: *Which one is it sorry?*
A: `You know the piece that was [pre]placed that was there in the beginning?`

repetition: The utterance or part of the utterance that was the problem is repeated.
Example:
A: `You [definitely] do?`
B: *Now I do what?*
A: `You definitely do?`

reformulation: The utterance or part of the utterance that was the problem is reformulated.
Example:
A: `You should be able to slot [that] in [the front] leg and [below the bac]k.`
B: *I should be able to slot it in _?*
A: `You should be able to fit in like a jig[saw] puzzle.`

yes-no: The answer to the CR is yes or no.

    Example:

    A: `And you just fit that [int]o the first piece.`

    B: *`Into the first piece?`*

    A: `Yeah.`

no reaction: The original speaker does not react to the CR at all.

Dialogue Acts; 2006 Schema   If a dialogue act annotation scheme is defined, the following features will also be present for each utterance. (This scheme is inspired by DAMSL (Allen & Core 1997*a*), but tailored to the task at hand.) Note that for all these main DA-level features, **na** (= not applicable) is a valid choice, if the annotator thinks that the utterance does not have a discernible function on a given level. **na** will not be listed again below explicitly.

task_execution: What is being done in terms of executing the task? Possible values are:

    present_description: The utterance presents a description of a piece or location on the pentomino board.

        If this value is chosen, the feature **description_type** becomes available, with the possible values: **describe_piece** and **describe_position**.

    request_info: Utterance requests information about piece or location.

    request_action: Utterance asks for performance of particular task-level action; e.g. placement or rotation.

    suggest_error: Utterance suggests problem with current game situation.

task_management: Utterance is about how to tackle the task. Values are:

    discuss_setting: Utterance discusses setting of the game, rules etc.

    discuss_strategy: Utterance discusses concrete strategy for solving problem.

    coordinate_task: Utterance indicates transition from move to move (= from piece to piece).

    stalling: Utterance informs about delays in executing task.

feedback: Roughly, DAMSL's backward-looking function. Gives feedback about understanding of previous utterance. Values are:

    feedback_pos: Utterance claims / indicates explicitly that previous utterance has been understood.

    ask_confirmation: Utterance requests explicit confirmation of a proposed understanding.

clarification: Utterance requests further clarification of what is to be done. **Important:** Note that this is a slightly different definition of clarification as compared to `is_cr` above. This feature here is supposed to cover task-level clarifications ("how shall I do this?") as well.

If this value is chosen, the additional feature `relation_antecedent` becomes available, with the values:

repetition: Requests repeats (parts or all of the) previous utterance verbatim.

addition: Repetition (of parts or all) together with additional material.

reformulation/partner_version: Utterance offers for confirmation own version of previous content or independent description of game state.

independent: Utterance is formulated independently of antecedent utterance.

feedback_neg: Utterance signals non-understanding of previous utterance, stronger than clarification.

other: Marking abandoned or interrupted utterances (`incomplete`) or off-topic utterances (`comment`).

Dialogue Acts; DAMSL Schema   Some dialogues are annotated with a (slightly modified) DAMSL schema. See (Allen & Core 1997*b*) for details on the schema.

3.3 Topic Level

This level records the general topic of dialogue segments, potentiall spanning several utterances, and thus enabling a slightly more abstract, higher level analysis of DP's strategy.

3.3.1 Identifying Markables

Markables on this level must be created by the annotator. They are independent of the utterance level and can comprise less than one or more than one utterance. The idea is that one topic stretch spans all consecutive speech that has the same topic (as classified by the feature values below).

This notion of topic stretch is clearly related to the dialogue acts above, but is on a higher (and more coarse-grained) level, potentially grouping several dialogue acts together.

3.3.2 Features

There is only one feature on this level, namely `topic`, with the following possible values. Note that the classification is a bit awkward, as the values encode functions that are strictly speaking on different levels, and could very well occur at the same time. Always go for what you perceive is the 'main function' of the dialogue span. The purpose of this feature is to make a rough analysis possible of how much time is spent on which sub-task.

other: Default value; for anything that doesn't follow in any of the categories below.

piece: Identifying the piece that is currently being dealt with.

orientation: Determining its orientation; but only if this is done (mostly) independently from identifying position on board.

location: Identifying where to place the piece. (N.B.: this is not really independent from orientation, which can and often is expressed together with describing the location. Chose the value that seems to correspond to the major concern of the span.)

task_setup: DPs discuss constraints on the setting, who can see what (in general terms), that they have to proceed in order, this kind of thing. Not things that pertain to the concrete situation. E.g., "can you see what I do?" fits, "I have a block in the upper corner" doesn't. Also includes things like "oh, it's an elephant". Anything that is about general features of the task, independent from the current placement of pieces.

ground_stat: Things like "I think it's in the right place", "not sure, let's see what happens." "yeah, that's in". I.e., utterances about grounding status, or perhaps "task completion status".

meta_ling: CRs and the replies. (Utterances about previous utterances.)

problem_det: Locations where problems are made explicit. "I think we have a problem". "no. it doesn't fit"

state_descr: Description of current state of game (but not what goes under `task_setup`). "my board looks like..."

action_descr: What is being done, or should be done. "I'm moving the thing over to the edge", "let's go back to piece 2"

off_topic: "Oh, I'm glad I'm not doing your part"

Special Case: Dictation Corpus (ND) In the dictation task (ND), there is only one topic value of interest, `dictation`. It is used to compute an *effort* metric, by calculating the ratio of words from the dictation item uttered vs. words in the dictation item. (I.e., if the value for the metric is 1, there are exactly as many dictation words uttered as there are words in the dication item—presumably each word has been mentioned exactly once, but the metric doesn't guarantee that. If it is two, then some words have been repeated, etc.)

From this use of the annotation follow certain constraints: there really should only be dictated material in `topic:dict`. No interpunctuation (as in the dictated item it doesn't count as one word, and in the transcriptions it has a different meaning altogether, representing prosody), no instructions like "space", "comma", "apostrophe" etc. And if there were disfluencies in an utterance, include only the last bit (to the right of the "+", if the disfluency was properly annotated).

A difficult case: words that are spelled out. They should not be included in full (because they would be counted by the analysis script as many individual words, driving up the count way too much), but it would be fair to represent somehow the increased effort. What I did was to include the first two or three letters, which mean that they do count as two or three words rather than as just one. This is a bit of a hack, but not entirely non-defendable.

Re. the spaces in the numbers (and to some extent commas and apostrophies): it could be argued that they do belong to the dictation material, as so should be counted. However, I think it's better not to count them, even though they are most of the time treated as dictatable information. Unlike "real" parts of the items like the individual numbers or words, these strings are always optional. And even should it be the case that they are always dictated, not including them doesn't hurt as long as it's done consistently.

3.4 Move Level

The annotation on this level captures the most abstract information about the progress of the tasks and the successes and failures of the participants on the way to completing the task.

3.4.1 Identifying Markables

The move annotation segments the dialogues into large chunks corresponding to the highest level task segmentations. I.e., for the pentomino task, one move is all (continous) talk that deals with one piece. At the very minimum, there have to be 12 moves in a pentomino dialogue, as there are 12 pieces to place. (11 for the noise dialogues, as one piece is already placed.) There can be a higher number of moves, however, as any step back to dealing with a piece that was previously considered placed (or was agreed to be put aside for later)

will count as a move as well.

For the dictation task, there will be exactly as many moves as there were items to dictate, as in this task there was no possiblity for the dialogue participants to return to a previous item.

This markable must be defined by the annotator; please try to always include full utterances. (I.e., make the boundaries of moves utterance boundaries.)

N.B.: Ideally, one would use a recursive annotation here, to be able to insert repair moves into regular moves (see below for definition of this). For easier representation, we decided to forgo this level of detail and flatten out the structure. Hence, embedding of move B in move A is represented by the left part of A (before the "interruption" through B) being marked as `on_hold`. That is, from a sequence `A:regular on_hold B:repair A:regular` can be reconstructed that B is embedded in A, and presumably part of dealing with A. This means, a regular move for piece X can be `on_hold`, i.e. piece X has not been placed yet. (Otherwise, the placing of the piece would have served as regular move.) After the first placement of piece X, every move that deals with it is a repair move. Of course it is possible to embed repair moves or even regular moves concerning other pieces in a regular move.

3.4.2 Features

The features for annotation are:

piece: ID of the piece that is being placed. 1-12 in `PTT` and `VisPent` (simply the numbers on the solution); 1-11 + a piece descriptor in `NP`: 1l, 2n, 3v, 4t, 5y, 6w, 7x, 8f, 9c, 10p, 11i.

> **move_type: regular** This feature value is for moves that are a first attempt at placing a piece (but that may interrupted by repair moves). Use `repair` for moves that return to previously placed pieces.

grounding_status: This feature is about the explicitly expressed 'strength' of the grounding, or confidence in the placement.

> **confident:** The DPs move on and seem happy with what they did. ("Okay, I did it. Next one.")
>
> **unconfident:** DPs move on, but seem to be wary about their placement. ("Let's see what happens").
>
> **on_hold:** DPs don't finish placement (normally to go back to another piece that causes problems) and explicitly acknowledge this. ("Let's go back to number three for now.")
>
> **reconfirm:** The DPs went back to a piece that was already placed, but leave it where they put it and are (now) confident about the

placement.
("A: But you're sure you put X at position Y? B: Yes.")

move_world: This feature is about the actual status of the placement in the world, i.e. the game board.

The first options are about the success (or not) of a placement (w.r.t. the solution): `success`/`wrong`.

If a piece has not been moved, either because its placement has been reconfirmed (`grounding_status:reconfirm`) or because the move its in is being interrupted before any action is being taken on the piece (`grounding_status:on_hold`), `not_moved` should be selected as value for this feature.

For some dialogues the videos are corrupt and hence no value can be selected here, to distinguish these cases from accidental omissions, select `no_video`.

3.5 Referring Expression Level

This level finally captures in a very fine-grained way one central task in the dialogues, namely that of creating expressions that refer to either pieces of the pentomino game or to locations on the board.

(This level is not present in ND.)

3.5.1 Identifying Markables

Markables on this level must be created by the annotator. They are typically on a sub-utterance level, i.e. maximally the size of a whole utterance, typically shorter. This is the only level on which overlapping markables can occur; i.e., one markable can (partially or wholly) be part of another. This can occur because here different kind of entities are targeted that can contain each other.

Technically this does not change the basic process of creating markables in MMAX as described above.

Roughly speaking, the markables here are all expressions that in some way contribute to the reference to a piece or a location in the pentomino game. Here is one example to illustrate this (and that the markables can be overlapping); there are more examples below in Section 5.

```
"The first piece looks like a Z"
    "the first piece" --> type=piece, form: def_NP
    "a Z" --> type=comp_piece, form: indef_NP
    "the first piece looks like a Z" --> type=desc_piece,
    form: comp_sent
```

Note: the existing transcription for `PTT` is buggy in that pronouns with reduced pronunciation are often not split off. E.g., there's stuff like "jetzt legst'n Teil dort hin" ('now you put a piece there') where the pronoun "ein" is reduced to "'n", which is transcribed as part of the verb. As the minimal unit that can be selected in MMAX is the word (at least given how our projects are defined), the work-around here is to select the word that the target item is attached to as well; i.e. in our case, that would be "legst'n".

3.5.2 Features

In this section, the features that are to be annotated are described in detail.

(Note: we use a different typographic convention in the examples here. The examples are all printed in `typewriter font` (not italics), and if surrounding context is shown, it is bracketed off with round brackets. E.g., 'This piece (is ugly)' is an example for a reference "this piece" together with context "is ugly".)

`item_ID`: The ID of the item that is being referred to (in case of puzzle pieces) or the ID of the target location. Codes as for `piece` of the move level; with the special convention that "99" is used when the annotator cannot guess the intended reference, and that "$CODE_1$ i / $CODE_2$ u" is used when there is a mismatch between what the speaker *i*ntended and what the hearer *u*nderstood. (E.g., "10i / 9u" denotes a situation where the speaker intended to refer to piece 10, but the hearer understood – as evidenced by her actions – the reference to be to piece 9.)

`referent_type`: This feature classifies the type of the expression's referent.

`piece`: The expression refers to a piece. (If the referring expression is a noun with a relative clause, the latter should be included in the markable.)

```
The first piece
A piece
It (is long on the right side and short to the left)
(There are two pieces.) One (is equal on both sides)
(Do you see the piece?)(Yeah, I see) that
Dis, was am unsymmetrischsten aussieht
```

`part_of_piece`: The expression does not refer to a whole piece, but to a part of it.

```
The long part (is on the left side)
The bottom of the backwards L
(Above those) three squares on the right
(There is) one square sticking out
```

location: Use this value for expressions which refer to a location. Locations do not necessarily have to be static positions ('the upper right corner of the game board'), but can also be positions relative to other pieces or the game board, as the examples show.

```
(Du legst dis ma so, dass es) genau den Fuss fortsetzt
Am M entlang
(Dis is) unser Ruessel
Underneath the horizontal bar
In the lap of the L
At the top of your puzzle
(Shall I put it) against the joining
```

comp_piece: This value is for expressions that are used in comparisons, that is, where pieces are compared to something. Usually, these expressions are introduced by something like "it looks like". Note that if the introductory words are something like "it is", the expression is more likely to be annotated as `piece`, since the reference to the piece is not via a comparison, but rather through naming.

```
(It looks like) a backwards L
(now the next one is ,[...] which looks like) C
```

Whether to this value is to be chosen or not always depends on the context. There is no fixed syntactic form for references of this type; as the next example shows, even sentential descriptions can denote a comp_piece:

```
Also das S is quasi nich wie man's schreibt, sondern wie 'n Z
```

comp_loc: Like `comp_piece`, but for locations.

```
(Und das kommt an dieses W ran. Wie so'n) Sattel.
```

desc_piece: This value is for sentential descriptions of a piece. As said earlier, such sentential descriptions often include other referring expressions. There will be some detailed examples for this at the end of this chapter.

```
Das is dis erste Teil, das eingefuegt wird
The first piece looks like a backwards L
(Wir beginnen mit einem Stueck,) das sieht aus wie ein S
```

form: This feature classifies the form of the markable. This corresponds roughly to classical syntactical phrase definitions, but is partially semantically motivated.

pronoun: Use this feature value to annotate pronouns.

```
(And) it (is not symmetrical you know)
(Und jetzt musst du) es (einmal nach rechts drehen)
```

def np: This feature value is for definite noun phrases. Be aware of elision: The determiner can be elided, and in German dialogues even the noun can be elided ("das"). Those cases also count as definite noun phrases.

```
 The cross (is top left yeah)
(And) that (fits in with the C)
(Und) das (musst du erstmal flippen)
(Nee, eher)  Hinterbauch (so)
```

indef np: For indefinite noun phrases.

```
(There's a piece that's like) an el shape
(There is) a bar of three units
```

quant np: This is the appropriate feature value for quantifier phrases, i.e. phrases which consist of a quantifier and a (possibly elided) noun.

```
(It's like about four units by) two units
(Then there is) one kind of unit (coming off it)
```

comp sent: Use this feature value for sentences which express a comparison. (This will most often also be `type:desc piece`.)

```
(yeah) that's like a trunk (yeah)
A piece that's like a U shape
```

pp: This feature value marks a prepositional phrase. (In most cases, these denote locations).

```
(You should be able to slot that in) above the front leg
(And you just fit that) into the first piece
(Und dann kommt das sozusagen) am Rcken (hin)
```

attribution sent Attribution sentences are connected with `type:desc piece`. They express properties of a piece. As a test, consider: Is the semantic content (or even the wording) of the sentence an expression of the form "Piece X has part Y"?

```
It's four units and then there is one kind of unit coming off it
(I think) it's like three units three units
```

other sent: This value is again connected to `type:desc piece`. It should be used for sentential descriptions which cannot be otherwise classified.

```
Und das ist der Hinterlauf sozusagen
```

other: This is a catch-all category for forms that cannot be classified.

perspective: This feature is about the perspective taken in the reference, or the strategy used in referring. The top-level distinction here is whether the piece or position is referred to via a decomposition of its shape (analytical), a more holistic comparison of its shape to some other shape (analogical), or via its relation to other referents.

> **none:** No perspective is taken and no strategy of referring was used. This is the case for pronouns.
>
> > `(Ja, da muss) es (hin)`
> > `(Yeah,) it (should fit under it)`
> > `This (is horizontally now)`
>
> Additionally, in the Vispento data (setting with visual channel), this value was also chosen for references to pieces via their grid position:
>
> > `(Gut, dann das Kreuz,)das Dritte von oben`
> > `(Dis is) in der unteren Reihe das Dritte von links`
>
> **TO DO: This should be revised, should be separate value!**
>
> **unspecific:** This feature value is for expressions whose referent cannot be identified.
>
> > `(Alright , so now where do I put) this fucker?`
> > `(Now, hopefully,) this other one (should fit)`
> > `Das (muss man aber erst noch irgendwie drehen)`
>
> **name_number:** The items are referred to via the number on the solution (ID). Notice that sometimes it is quite easy to confuse `name_number` and `relative_to_dialogue`. Only if the uttered number is identical to the ID of the piece, this feature value is appropriate.
>
> > `(Ok,) Nummer 1`
> > `Die 8 (ist dis, was oben links noch liegt)`
> > `(Alright,) number nine (is a U shape)`
>
> **analytical:** "Analytical" means viewing the referent as built up from parts.
>
> > **analytical_blocks:** Here, the pieces are analyzed into blocks.
> >
> > > `(Is it) the four vertical ones (and then one)?`
> > > `It is like four in a line and then one off to the side`
> > > Sometimes, the classification of an expression can be difficult. Especially with this feature value, there are ambiguous cases, as seen here:
> > >
> > > `Eine lange Saeule mit drei Teilen`

This could be either `analytical_blocks` or `analogical_geometrical_shape`. Use the first possibiliy, since the kind of composition is more important to our concerns.

`analytical_other` This feature value is for all other types of analyzing the referent by viewing it as built up from parts.

```
(Instead of putting the additional cube on top of the middle one
you put it) on one of the sides
```

```
The horizontal bit of the original square shape that was in there
```

`analogical:` Reference by analogy of the shape of the referent to something.

`analogical_letter:` Analogy to a letter shape.
```
(It's) an el shape
(We will say it is) a C
(Das Ganze sieht aus wie) ein Z
```

`analogical_elephant:` Use this feature value for analogies to the elephant's parts.
```
The top left of the elephant's back
(It is like) the back leg
The elephant's trunk
Into the head
An den Hinterkopf
```

`analogical_geometrical_shape:` In case there is some analogy to a geometrical shape, use this feature value.
```
The angle (goes right into the bottom right corner)
Der grosse massive Teil
Das Kreuz / The cross
```

`analogical_other:` This is a catch-all category for analogies that cannot be classified otherwise.
```
The stem
The trunk of the T
That Tetris shape
It is like a visor
It is a kind of strange shape
```

`relative:` These feature values are about the referent of the expression being in a spacial relation to a known object. (For nd and PTT this can only be done for locations, since the locations of pieces outside the outline is not known to IG; in `visPent` this can also be done for the identification of pieces.)

`relative_to_board:` This feature value is used for expressions, which express a relation to the game board. Be sure not to confuse this with `analogical_elephant`, since the game board

is shaped like an elephant. As soon as there is some elephant
in the expression, classify it as `analogical elephant`.

```
Into the bottom right corner

(And that goes) towards the left of the grid

(Dann muss das linke untere Quadrat der momentan benutzten Form)
in die fuenfte Spalte

(Und dis kommt) in den oberen Teil
```

relative to placed pieces: Use this feature value, in case there
is a relation to pieces, that have already been placed on the
game board.

```
Around the left hand of the fifth piece
The one that's in between two and four
Neben das, was aussieht wie ne Treppe
```

relative to current position: This feature value is for expres-
sions, whose perspective is seen from a known location or a
piece on the board. It is also used for parts of pieces, which
are relative to the orientation of the piece on the game board,
as seen in the last three examples.

```
(Genau, jetzt noch) nen Stck nach rechts
Nach rechts (genau so)
(Oder noch) eins weiter nach hinten
The top edge of that (piece)
Das untere Ende vom Z
Die Spitze vom M
```

relative to dialogue: These referring expressions are often a
lot like those with **name number**. Draw the line between "Piece
number seven" (for piece 7 in the solution) and "The sev-
enth piece". The first gets **name number**, while the latter gets
`relative to dialogue`.

```
The first piece (is gonna go in the far
bottom right corner)

The next one

This one that could also possibly match
that description

The one we had problems with earlier

The second piece you have put in
```

```
        the piece, that youre manipulating now
```
relative_to_other: This category catches all relations that could
not be classified otherwise.
```
    (It is) a mirror shape
```

4 Decision Trees

This section presents a collection of decision trees for annotating the corpus. This section can be given separately to annotators (as has been done in the 2008 reliability test), hence there is some repetition of material already presented above.

The trees are to be traversed top-down. The trees are binary, so for each node that contains a question, there are two choices. Go to the left child if the answer is "yes", otherwise choose the right one. For the guidelines for manual creation of markables (e.g. move level and referring expression level) please consult the descriptions above.

Some features take values that are to be typed in (rather than chosen from a list), for example the piece ID. The IDs of the pieces depend on annotation conventions above. In our experiments these are the numbers 1 - 12 in PTT and VisPento; numbers 1 - 11 with a descriptor in ND: 1l, 2n, 3v, 4t, 5y,6w, 7x, 8f, 9c, 10p, 11i with the special convention that "99" is used when the annotator cannot guess the intended reference, and that "$CODE_1$ i / $CODE_2$ u" is used when there is a mismatch between what the speaker *i*ntended and what the hearer *u*nderstood.

[This section was written by Janine Wolf.]

4.1 Utterance level

4.1.1 is_cr

Feature for marking CRs. The key question for deciding whether something is a CR or not is: "Is the utterance pointing out an understanding problem of a previous utterance and asking for repair?"

Clarification requests can concern several kinds of problems, e.g. when the antecedent utterance has not been understood acoustically ("Say that again.") or parts of the utterance weren't clear for the player ("I should put that piece where?"; "I need to do what?"). They range from really short utterances like "Huh?" to whole sentences, as seen above.

Once identified, i.e. when the value for **is_cr** is set to "yes", a bundle of features have to be annotated.

cr_form   This feature concerns the syntactic form of the utterance.

(1) complete sentence?

cr_form: complete   (2) conventional phrase/particle?

cr_form: particle   **cr_form**: partial
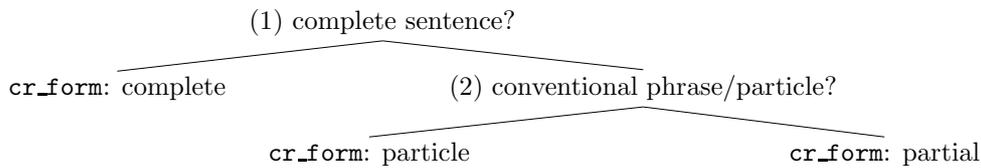
(1) Is the utterance a complete sentence (i.e. it has a tensed verb)?
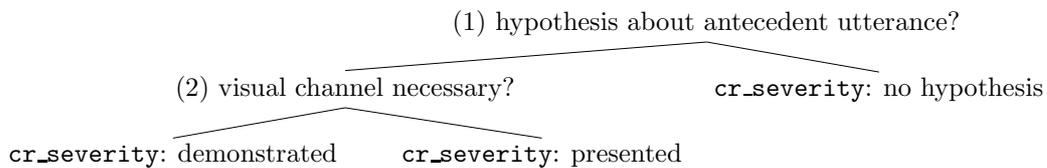
(2) Is the utterance a conventional phrase or particle, such as "Sorry?",
     "Huh?", "Pardon?", "What?" ?

cr_extent

(1) CR points out problematic part of antecedent utterance?

cr_extent: yes   **cr_extent**: no
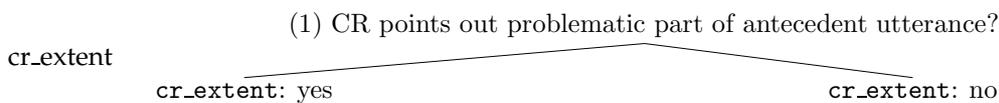
(1) Does the CR point out the exact part of the antecedent utterance, which
     was problematic? E.g. A: We need the cross. B: The cross? (yes) / We
     need the what? (yes) vs. Pardon? (no)

cr_severity   This feature is about the presentation of a hypothesis uttered
by the executer inside the CR.

(1) hypothesis about antecedent utterance?

(2) visual channel necessary?   **cr_severity**: no hypothesis

cr_severity: demonstrated   **cr_severity**: presented

(1) Does the CR give a hypothesis about what the antecedent could have
     said or meant? E.g. A: I saw Peter. B: Peter? (yes) / Pardon? (no) /
     Who? (no)

(2) Is a visual channel between IG and IF necessary in order for the ad-
     dressee to fully understand this CR? This feature accounts for CRs like
     "So?" or "Da?".

cr_rel_ant   This feature is about the relation of the CR to its antecedent.

(1) parts of ant.utt. repeated?

(2) new material?   (3) reformulation?

(4) new wh-word?   **cr_rel_ant**: repet.   **cr_rel_ant**: reform. **cr_rel_ant**: indep.

cr_rel_ant: add_wh   **cr_rel_ant**: add.
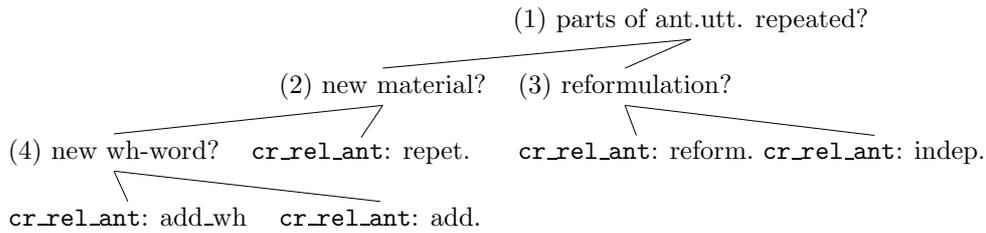
(1) Are parts of the antecedent utterance repeated?

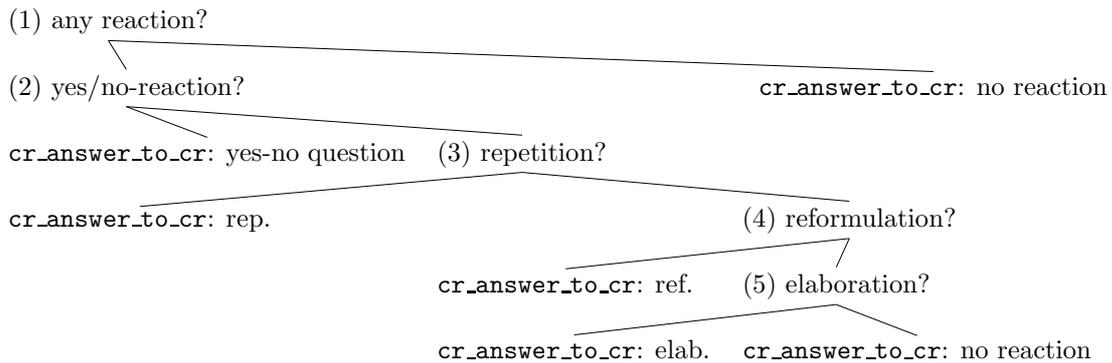(2) Does the CR contain any linguistic material, that was not present in the antecedent utterance, such as wh-words? E.g. A: I saw Peter. B: Peter Miller? (yes) / Which Peter? (yes) / Who? (no)

(3) Is the CR a reformulation of the antecedent utterance?

(4) Is the new material a wh-word?

add = addition ; repet. = repetition; indep. = independent; reform. = reformulation

cr_answer_to_cr   This is about the addressee's reaction on the CR.

(1) any reaction?

(2) yes/no-reaction?   **cr_answer_to_cr**: no reaction

cr_answer_to_cr: yes-no question   (3) repetition?

cr_answer_to_cr: rep.   (4) reformulation?

cr_answer_to_cr: ref.   (5) elaboration?

cr_answer_to_cr: elab.   **cr_answer_to_cr**: no reaction
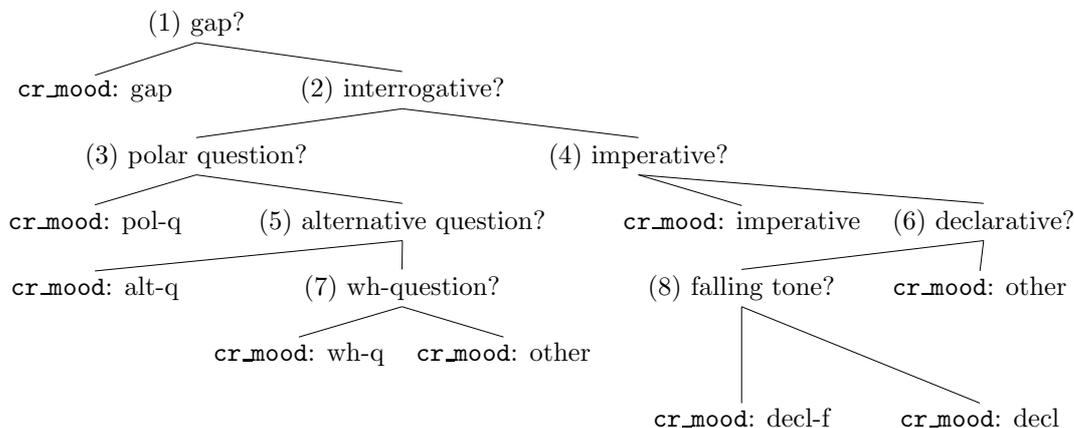
(1) Does the antecedent react on the CR at all?

(2) Does the antecedent react with either "yes" or "no" ?

(3) Is the answer a repetition of the problematic part of the antecedent utterance?

(4) Is the answer a reformulation of the problematic part of the antecedent utterance?

(5) Does the antecedent elaborate the problematic part of the antecedent utterance?

    elab. = elaboration; repet. = repetition; reform. = reformulation

cr_mood   This feature is concerned with sentence mood.

            (1) gap?

  cr_mood: gap   (2) interrogative?

      (3) polar question?   (4) imperative?

  cr_mood: pol-q   (5) alternative question?   **cr_mood**: imperative (6) declarative?

  cr_mood: alt-q   (7) wh-question?   (8) falling tone?   **cr_mood**: other

            cr_mood: wh-q   **cr_mood**: other

                cr_mood: decl-f   **cr_mood**: decl

(1) Does the CR contain a gap? E.g. A: Get the cross. B: Get the _?

(2) Does the CR have an interrogative word order?

(3) Is the CR a polar question (i.e. a y/n-question)? E.g. A: I saw Peter. B: Did you say you saw Peter?

(4) Is the CR an imperative sentence? E.g. A: Get the cross. B: Say that again.

(5) Is the CR an alternative question? E.g. A: I saw Peter. B: Peter Miller or Peter Smith?

(6) Is the CR a declarative sentence (i.e. non-interrogative word order and the tone is rising at the end) E.g. A: I saw Peter. B: You saw Peter? (decl) / Did you see Peter? (no, that's interrogative)

(7) Is the CR a wh-question? E.g. A: I saw Peter. B: Who is Peter?

(8) Is the CR basically a declarative sentence (as described above), but is the tone falling at the end? E.g. A: I saw Peter. B: You saw Peter.
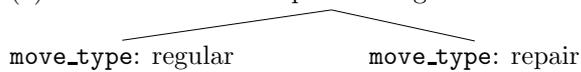
4.2 Move Level

This subsection is concerned with annotating the moves within a dialogue.
These markables have to be created manually.

4.2.1 Piece

The feature `piece` is to be filled with the ID of the piece that the DPs
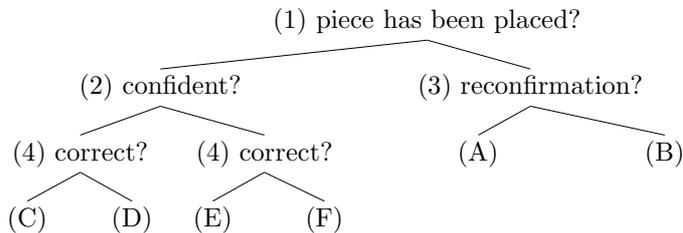currently work on.

4.2.2 Move type

(1) first time to discuss piece or regular move on hold?

`move_type`: regular   `move_type`: repair

(1) Is it the first time the current piece is discussed? Or if it has been
    discussed already, is there a previous move concerning this piece, which
    is `move_type`: regular and `grounding_status`: on_hold ? (This compo-
    sition of feature values means, that the piece has not been placed on
    the board in that previous attempt.)

4.2.3 Grounding and world status

(1) piece has been placed?

(2) confident?   (3) reconfirmation?

(4) correct?   (4) correct?   (A)   (B)

(C)   (D)   (E)   (F)

(1) Has the piece been (actively) placed somewhere on the game board
    in the course of the move? This includes moving the piece from one
    location on the board to another one on the board.

(2) Are the participants confident about their decision, i.e. do they seem
    happy with what they did? E.g. "Okay, I did it. Next one."

(3) Is this move just a reconfirmation about a piece, which is already on
    the game board? There is no movement involved. E.g. "The L shape
    is in the leg, right? Okay. ¡...discussion of another piece...¿"

    (4) Have the participants placed the piece correctly, i.e. according to the solution? This requires the video data to be consulted. If there is no video, choose `move_world`: no video

    (A) `grounding_status`: reconfirm; `move_world`: not moved

    (B) `grounding_status`: on_hold; `move_world` : not moved

    (C) `grounding_status`: confident; `move_world` : success

    (D) `grounding_status`: confident; `move_world` : wrong

    (E) `grounding_status`: unconfident; `move_world` : success

    (F) `grounding_status`: unconfident; `move_world` : wrong

4.3 Referring Expression Level

This level concerns referring expressions, i.e. expressions that refer to either pieces of the pentomino game or to locations on the board. The markables on this level have to be created manually.

The referring expressions, that we are interested in, are expressions, which refer to
`pieces` ("The first piece"; "it"),
`their parts` ("the long part"; "two blocks on the right") or
`objects/things/locations, which they are compared with` ("an L"; "a staircase"; the back leg"; "the trunk"; "das Kreuz").
Referring expressions, which denote a location, a piece is compared with ("a saddle"), are rather rare.
Referring expressions can be pronouns, NPs, PPs or sentences. The latter can either express, what a piece has/consists of ("It's two blocks down and one to the right"), or what it is like ("It's like a trunk, yeah").

Referring expressions often occur nested in other referring expressions. Look at these fully annotated examples (for more, please consult the annotation manual!):

```
It's like about four units by two units
it's --> type=piece, form=pronoun, perspective=none
It's like about four units by two units --> type=desc_piece,
  form=attribution_sent,
perspective=analytical_blocks
four units --> type=part_of_piece, form=quant_NP,
perspective=analytical_blocks
two units --> type=part_of_piece, form=quant_NP,
  perspective=analytical_blocks
```

```
yeah that's like a trunk yeah
that's like a trunk --> type=desc_piece, form=comp_sent,
perspective=analogical_elephant
that --> type=piece, form=def_NP, perspective=unspecific
a trunk --> type=comp_piece, form=indef_NP,
perspective=analogical_elephant

We will say it is a C
it --> type=piece, form=pronoun, perspective=none
 a C --> type=piece, form=indef_NP, perspective=analogical_letter

We will say it looks like a C
it --> type=piece, form=pronoun, perspective=none
it looks like a C --> type=desc_piece, form=comp_sent,
 perspective=analogical_letter
 a C --> type=comp_piece, form=indef_NP, perspective=analogical_letter
```
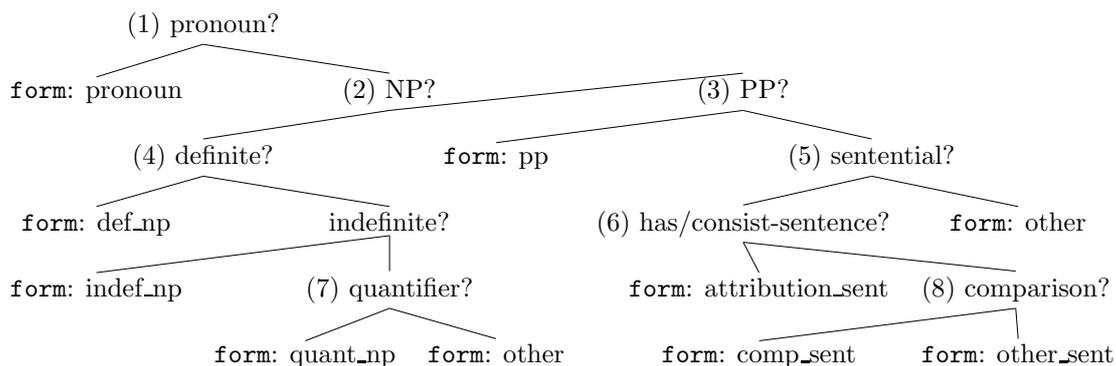
4.3.1 Form

This feature is about the form of a referring expression.

```
                    (1) pronoun?
form: pronoun              (2) NP?                    (3) PP?
           (4) definite?          form: pp              (5) sentential?
   form: def_np      indefinite?        (6) has/consist-sentence?   form: other
 form: indef_np    (7) quantifier?         form: attribution_sent   (8) comparison?
           form: quant_np  form: other        form: comp_sent   form: other_sent
```

(1) Is the referring expression a pronoun (es, it)? Please consider that pronouns could be sticking to the previous word, e.g. "dasses" in german. Annotate the whole word as pronoun.

(2) Is the referring expression a nominal phrase?

(3) Is the referring expression a prepositional phrase?

(4) Is the NP definite, i.e. does it have a determiner? Please consider that determiners and even nouns could be elided. These cases also count as definite NP. E.g. "das [elided: Stueck]"; "That [fits in there]"

(5) Is the referring expression a sentence?

(6) Is the referring expression an attribution sentence? They express, what
a piece has or consists of. If one wants to make up some kind of memory
hook to identify an attribution sentence, it could be like this: Is the
semantic content (or even the wording) of the sentence an expression
of the form "Piece X has part Y"? E.g. "It's four units and then there
is one kind of unit coming off it"

(7) Is the NP a quantifier phrase, i.e. does it consist of a noun and a
quantifier? E.g. "(Und jetzt) nen Stck hher"; "(Then there is) one
kind of unit (coming off it)"

(8) Does the sentence express a comparison? E.g. "A piece that's like a U
shape"

4.3.2 Type

This feature denotes the type of the expression's referent.
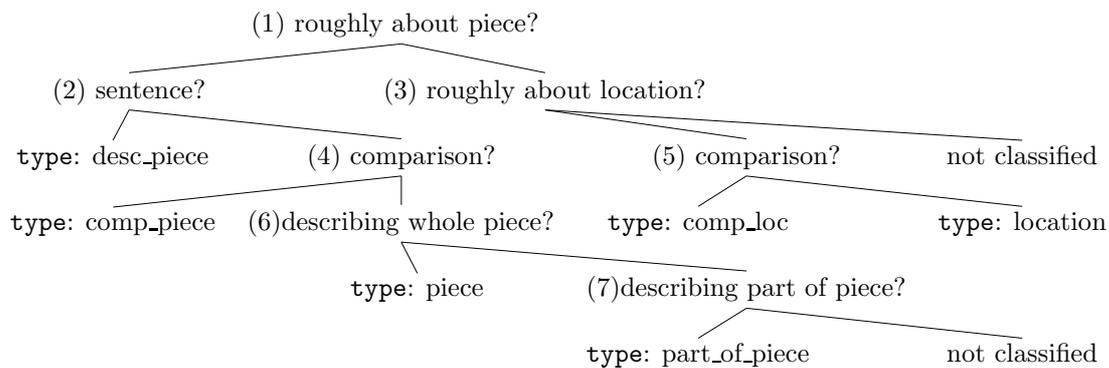
(1) Is the expression's referent roughly about a piece or it's parts or shape?

(2) Is the referring expression sentential, i.e. is it a sentence?

(3) Is the expression's referent roughly about a location or a comparison of
a piece with a location?

(4) Is the expression's referent compared to a piece? Usually these expres-
sions are introduced by "it looks like". E.g. "(It looks like) a backwards
L"; "(now the next one is ,[...] which looks like) C"

(5) Is the expression's referent compared to the location? This is fairly a
rare phenomenon. E.g. "(Und das kommt an dieses W ran. Wie so'n)
Sattel."

(6) Is the referring expression describing the whole piece? E.g. " It (is long on the right side and short to the left)"; "Dis, was am unsymmetrischsten aussieht"; "The first piece"; "A piece"

(7) Is the referring expression describing a part of the piece? E.g. "The bottom of the backwards L"; "(Above those) three squares on the right "; "The long part (is on the left side)"

4.3.3 Perspective

This feature is about the perspective taken in the reference, or the strategy used in referring.

(1) any perspective?

   (2) identifiable referent?   `perspective`: none

      (3) name+number?   `perspective`: unspecified

`perspective`: name_number   (4) decomposition of shape?

  (5) decomp. into blocks?   (6) comparison to shape?

`perspective`: analyt. bl.   `perspective`: analyt. oth.

             (7) compared to letter?   not classified

     `perspective`: analogical_letter   (8) compared to elephant?

    `perspective`: analogical_elephant   (9) compared to geometr. shape?

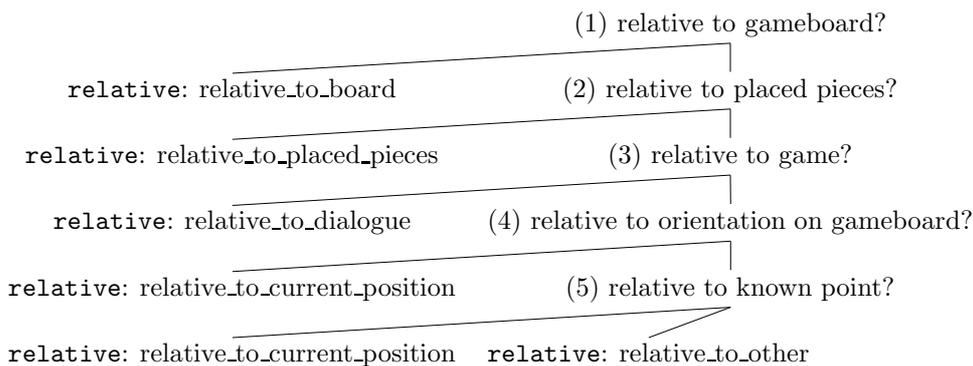   `perspective`: geom. sh.   `perspective`: analogical_other
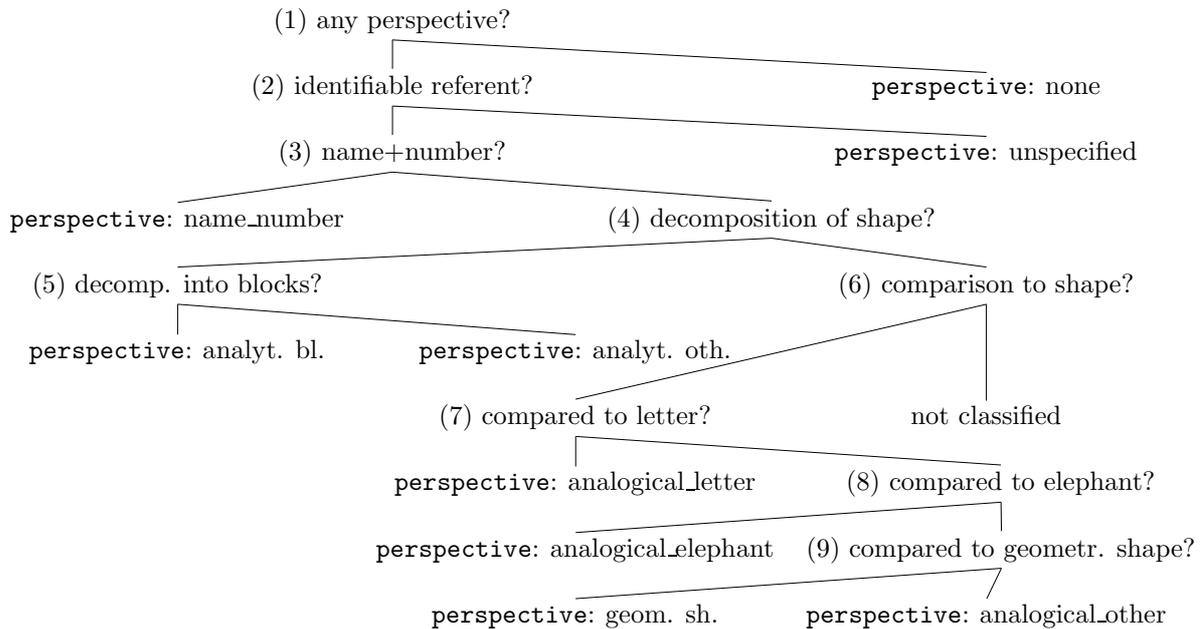
(1) Is a perspective taken? This is to be answered with "no" for pronouns. Additionally, in the Vispento data (setting with visual channel), the use of this feature value was extended to the addressing of pieces that lay beside the game board (not having been dealt with yet), e.g. "(Gut, dann das Kreuz,)das Dritte von oben"

(2) Is the referent identifiable, i.e. do we know, what this expression is about? E.g. "(Alright , so now where do I put) this fucker?" (no); "(Now, hopefully,) this other one (should fit)" (no); "Das (muss man aber erst noch irgendwie drehen)" (no); "The previous piece" (yes)

(3) Is the piece referred to via it's name and number within the game? E.g. "the fifth piece". Notice that sometimes it is quite easy to confuse `name_number` and `relative_to_dialogue`. Only if the uttered number is identical to the ID of the piece, this feature value is appropriate. E.g. "(Alright,) number nine (is a U shape)"

(4) Is the referent referred to analytically? "Analytical" means viewing the referent as built up from parts. E.g. "The piece has two parts"

(5) Here, the pieces are analyzed into blocks. E.g. "(Is it) the four vertical ones (and then one)?"; "It is like four in a line and then one off to the side"

(6) Is the referent referred to analogically? "Analogical" means comparing the shape of the referent to something.

(7) Is there an analogy to a letter? E.g. "(We will say it is) a C"

(8) Is there an analogy to an elephant? E.g. "The elephant's trunk"; "Into the head"

(9) Is there an analogy to a geometrical shape? E.g. "The angle (goes right into the bottom right corner)"; "Der grosse massive Teil"; "Das Kreuz / The cross"

geom.sh. = analogical_geometrical_shape; analyt. bl. = analytical_blocks, analyt. oth. = analytical_other

4.3.4   Relative

These feature values are about the referent of the expression being in a spatial relation to a known object. (For `nd` and `PTT` this can only be done for locations, since the locations of pieces outside the board is not known to IG; in `VisPento` this can also be done for the identification of pieces.)

```
                                          (1) relative to gameboard?
         _____
relative: relative_to_board               (2) relative to placed pieces?
      _____
relative: relative_to_placed_pieces         (3) relative to game?
      _____
relative: relative_to_dialogue      (4) relative to orientation on gameboard?
      _____
relative: relative_to_current_position       (5) relative to known point?
      _____
relative: relative_to_current_position   relative: relative_to_other
```

(1) Does the referring expression relate to the game board? E.g. "(And that goes) towards the left of the grid"; "(Dann muss das linke untere Quadrat der momentan benutzten Form) in die fuenfte Spalte"

(2) Does the referring expression relate to pieces, that have already been placed? E.g. "Around the left hand of the fifth piece"; "Neben das, was aussieht wie ne Treppe"

(3) Does the referring expression include "dialogue internal" references, such as e.g. "The one we had problems with earlier"; "The next one" or "The first piece"?

(4) Does the referring expression relate to the orientation of pieces or parts of pieces relative to the game board? E.g. "The top edge of that (piece)"; "Das untere Ende vom Z"

(5) Does the referring expression relate to some known (i.e. fixed) point? E.g. "(Genau, jetzt noch) nen Stck nach rechts"; "(Oder noch) eins weiter nach hinten"

5 Examples

In this section the full annotation of all examples that were shown above is presented, and some additional ones.

5.1 Utterance Level

5.1.1 Examples for CRs

Example:
A: `If you could put that on the elephant's back [foot]?`
B: *`The back foot?`*
A: `Yeah.`
mood:decl form:partial rel_ant:repetition extent:yes severity:hypo presented answer:yes-no

Example:
A: `Genau, wo der Pfeil jetzt ist...das Ding.`
B: *`Das da?`*
A: `Ja.`
mood:decl form:partial rel_ant:independent extent:yes severity:hypo demonstrated answer:yes-no

Example:
A: `That fits directly under the last step.`
B: *Under the what?*
A: `Last step.`
mood:wh-q form:partial rel_ant:add-wh extent:yes severity:no hypo answer:repetition

Example:
A: `Uhm, you need to get it so...`
B: *The long one or the ?*
A: `Yah you want the long ...   and the short b.`
mood:alt-q form:partial rel_ant:reformulation extent:yes severity:hypo presented answer:reformulation

Example:
A: `Should meet the t[op] of the T shape.`
B: *The top of the _?*
A: `Not the top of the T shape, the bottom of the T.`
mood:gap form:partial rel_ant:repetition extent:yes severity:no hypo answer:reformulation

Example:
A: `And the [long part] is on the [le]ft.`
B: *Say that again.*
A: `The long part [of the] T is on the left.`
mood:imp form:complete rel_ant:independent extent:no severity:no hypo answer:repetition

Example:
A: `The next one...I would d[escri]be as`
`a W shape [in the f]ont tha[t] you [write "Wario"] in.`
B: *You have to start again, it keeps cutting up.*
A: `It's basic like W shape.`
mood:imp form:complete rel_ant:independent extent:no severity:no hypo answer:reformulation

Example:
A: `That original [piece?]`
B: *The original but which?*
A: `[You kno]w the original?`
`There's a piece [that's a-]`
mood:wh-q form:partial rel_ant:add-wh extent:yes severity:no hypo answer:reformulation

Example:
A: `And it's a [T shape, you're] looking for.`

B: *Sorry?*
A: `A T shape, the letter T.`
mood:decl form:particle rel_ant:independent extent:no severity:no hypo answer:reformulation

Example:
A: `Nimm mal zuerst das Teil, was in der unteren Reihe das Dritte`
`von links is, ne?`
B: *Das Dritte von links.*
A: `Ja genau.`
mood:decl form:partial rel_ant:addition extent:yes severity:hypo presented answer:yes-no

Example:
A: `Hier unten rechts, hier...ja genau, rechts, rechts!`
B: *In die Hinterbeine.*
A: `Nee, nen Stck nach...genau, in die Hinterbeine.`
mood:decl-f form:partial rel_ant:reformulation extent:yes severity:hypo presented answer:yes-no

Example:
A: `Dis is ahm denk ich mal in der unteren Reihe das Dritte von links.`
B: *In der unteren, in der untersten waagerechten?*
A: `Achso, nee.`
mood:decl form:partial rel_ant:addition extent:yes severity:hypo presented answer:yes-no

Example:
A: `There should have been a [normal one]?`
B: *There should have been what?*
A: `Just a straight one?`
mood:wh-q form:partial rel_ant:add-wh extent:yes severity:no hypo answer:reformulation

Example:
A: `You know the second piece which is [the front leg].`
B: *Sorry...you're cutting out there?*
A: `You know the [second pie]ce which is the front leg.`
mood:decl form:complete rel_ant:independent extent:no severity:no hypo answer:repetition

Example:
A: `So the end of the [Z] should be righ[t] on the`
`bott[o]m the grid.`

B: *Pardon?*
A: The end of the Z should be facing right down
to the bottom of the grid.
mood:decl form:particle rel_ant:independent extent:no severity:no hypo answer:repetition

Example:
A: And so the little square that is sticking out faces down.
B: *Faces down?*
A: So it's horizontal.
mood:decl form:partial rel_ant:repetition extent:yes severity:hypo presented answer:elaboration

Example:
A: So, we need that [touching corners with] that trunk bit.
B: *I need to do what?*
A: You need it touching corners with the trunk bit.
mood:wh-q form:partial rel_ant:add-wh extent:yes severity:no hypo answer:repetition

Example:
A: Oh sorry, the [sixth] row down.
B: *The second row down yeah?*
A: Sixth row down.
mood:decl form:complete rel_ant:repetition extent:yes severity:hypo presented answer:repetition

Example:
A: Genau da hin.
B: *Ganz...da?*
A: Du machst das perfekt.
mood:decl form:partial rel_ant:independent extent:yes severity:hypo demonstrated answer:yes-no

Example:
A: It'[s a bit has] an e[longat]ed bottom.
B: *Which one is it sorry?*
A: You know the piece that was [pre]placed that was there in the
beginning?
mood:wh-q form:complete rel_ant:independent extent:yes severity:no hypo answer:elaboration

Example:
A: You [definitely] do?

B: *Now I do what?*
A: `You definitely do?`
mood:wh-q form:partial rel_ant:add-wh extent:yes severity:no hypo answer:repetition

Example:
A: `You should be able to slot [that] in [the front] leg and [below`
`the bac]k.`
B: *I should be able to slot it in _?*
A: `You should be able to fit in like a jig[saw] puzzle.`
mood:gap form:partial rel_ant:repetition extent:yes severity:no hypo answer:reformulation

Example:
A: `And you just fit that [int]o the first piece.`
B: *Into the first piece?*
A: `Yeah.`
mood:decl form:partial rel_ant:repetition extent:yes severity:hypo presented
answer:none

5.1.2 Examples for non-CRs

5.2 Referring Expression Level


```
The first piece --> type=piece, form=def_NP,
  perspective=relative_to_dialogue

A piece --> type=piece, form=indef_NP,
perspective=unspecific

It is long on the right side and short to the left
It is long on the right side and short to the left -->
type=desc_piece, form=attribution_sent,
perspective=analytical_other
It --> type=piece, type=piece, form=pronoun, perspective=none

(There are two pieces.) One (is equal on both sides) --> type=piece,
form=quant_NP,
perspective=unspecific

Do you see the piece? Yeah, I see that
that --> type=piece, form=def_NP, perspective=unspecific
the piece --> type=piece, form=def_NP, perspective=unspecific
```

```
Dis, was am unsymmetrischsten aussieht --> type=piece, form=def_NP,
perspective=analogical_geometrical_shape

The long part (is on the left side) --> type=part_of_piece, form=def_NP,
 perspective=analytical_other

The bottom of the backwards L
The bottom of the backwards L --> type=part_of_piece,
 form=def_NP,
 perspective=relative_to_current_position
the backwards L --> type=piece, form=def_NP,
  perspective=analogical_letter

Above those three squares on the right
Above those three squares on the right --> type=location, form=pp,
perspective=relative_to_current_position
three squares on the right --> type=part_of_piece, form=quant_NP,
 perspective=analytical_blocks

(There is) one square sticking out --> type=part_of_piece, form=quant_NP,
   perspective=analytical_blocks

Da, wo ich drei Stueck hab am Stueck --> type=desc_piece,
  form=attribution_sent,
  perspective=analytical_blocks

Du legst dis ma so, dass es genau den Fuss fortsetzt
dis --> type=piece, form=def_NP, perspective=unspecific
es  --> type=piece, form=pronoun, perspective=none
genau den Fuss fortsetzt --> type=location, form=other,
  perspective=analogical_elephant

Am M entlang --> type=location, form=pp,
  perspective=relative_to_placed_pieces

Dis is unser Ruessel
dis --> type=piece, form=def_NP, perspective=unspecific
unser Ruessel --> type=location, form=def_NP,
perspective=analogical_elephant

Underneath the horizontal bar
Underneath the horizontal bar --> type=location, form=pp,
 perspective=relative_to_placed_pieces
the horizontal bar --> type=part_of_piece, form=def_NP,
  perspective=analogical_geometrical_shape
```

```
In the lap of the L
In the lap of the L --> type=location, form=pp,
perspective=relative_to_current_position
the L --> type=piece, form=def_NP, perspective=analogical_letter

At the top of your puzzle --> type=location, form=pp,
perspective=relative_to_board

Shall I put it against the joining
it --> type=piece, form=pronoun, perspective=none
against the joining --> type=location, form=pp,
perspective=relative_to_placed_pieces

(It looks like) a backwards L
It looks like a backwards L --> type=desc_piece,
  form=attribution_sent,
  perspective=analogical_letter
it --> type=piece, form=pronoun, perspective=none
a backwards L --> type=comp_piece, form=indef_NP,
perspective=analogical_letter

now the next one is ,[...] which looks like C
the next one is, which looks like C --> type=desc_piece, form=comp_sent,
 perspective=analogical_letter
the next one --> type=piece, form=def_NP, perspective=relative_to_dialogue
C --> type=comp_piece, form=indef_NP, perspective=analogical_letter

Also das S is quasi nich wie man's schreibt, sondern wie 'n Z
Also das S is quasi nich wie man's schreibt,
sondern wie 'n Z
--> type=comp_piece, form=other_sent,
 perspective=analogical_letter
'n Z --> type=comp_piece, form=indef_NP,
perspective=analogical_letter

Und das kommt an dieses W ran. Wie so'n Sattel.
das --> type=piece, form=def_NP, perspective=unspecific
an dieses W ran --> type=location, form=pp,
  perspective=relative_to_placed_pieces
dieses W --> type=piece, form=def_NP, perspective=analogical_letter
so'n Sattel --> type=comp_loc, form=indef_NP,
 perspective=analogical_other

Das is dis erste Teil, das eingefuegt wird
```

```
Das is dis erste Teil, das eingefuegt wird --> type=desc_piece,
  form=other_sent,
perspective=relative_to_dialogue
das --> type=piece, form=def_NP, perspective=unspecific
dis erste Teil, was eingefuegtwird --> type=piece, form=def_NP,
perspective=relative_to_dialogue

The first piece looks like a backwards L
The first piece looks like a backwards L --> type=desc_piece,
form=comp_sent,
perspective=analogical_letter
The first piece --> type=piece, form=def_NP,
  perspective=relative_to_dialogue
a backwards L --> type=comp_piece, form=indef_NP,
perspective=analogical_letter

Wir beginnen mit einem Stueck, das sieht aus wie ein S
einem Stueck, das sieht aus wie ein S --> type=piece, form=indef_NP,
perspective=analogical_letter
das sieht aus wie ein S --> type=desc_piece, form=comp_sent,
 perspective=analogical_letter
ein S --> type=comp_piece, form=indef_NP,
 perspective=analogical_letter

(And)it (is not symmetrical you know) --> type=piece,
form=pronoun,
perspective=none

(Und jetzt musst du) es (einmal nach rechts drehen) --> type=piece,
  form=pronoun,
  perspective=none

 The cross is top left yeah
the cross --> type=piece, form=def_NP,
  perspective=analogical_geometrical_shape
top left --> type=location, form=pp, perspective=relative_to_board

And that fits in with the C
that --> type=piece, form=def_NP, perspective=unspecific
the C --> type=piece, form=def_np, perspective=analogical_letter

(Und) das (musst du erstmal flippen) --> type=piece, form=def_np,
  perspective=unspecific

(Nee, eher)  Hinterbauch (so) --> type=location, form=def_np,
```

```
  perspective=analogical_elephant

There's a piece that's like an el shape
a piece that's like an el shape --> type=piece, form=indef_NP,
perspective=analogical_letter
an el shape --> type=comp_piece, form=indef_NP,
 perspective=analogical_letter

(There is) a bar of three units --> type=part_of_piece, form=indef_np,
perspective=analytical_blocks

It's like about four units by two units
It's like about four units by two units --> type=desc_piece,
  form=attribution_sent,
perspective=analytical_blocks
four units --> type=part_of_piece, form=quant_NP,
perspective=analytical_blocks
two units --> type=part_of_piece, form=quant_NP,
  perspective=analytical_blocks

Then there is one kind of unit coming off it
Then there is one kind of unit coming off it --> type=desc_piece,
 form=attribution_sent,
perspective=analytical_blocks
one kind of unit coming of it --> type=part_of_piece, form=quant_NP,
 perspective=analytical_blocks
it --> type=piece, form=pronoun, perspective=none

(Und jetzt) nen Stck hher --> type=location, form=quant_np,
  perspective=relative_to_current_position

yeah that's like a trunk yeah
that's like a trunk --> type=desc_piece, form=comp_sent,
perspective=analogical_elephant
that --> type=piece, form=def_NP, perspective=unspecific
a trunk --> type=comp_piece, form=indef_NP,
perspective=analogical_elephant

A piece that's like a U shape
A piece that's like a U shape --> type=piece, form=indef_NP,
 perspective=analogical_letter
that's like a U shape --> type=desc_piece, form=comp_sent,
  perspective=analogical_letter
a U shape --> type=comp_piece, form=indef_NP,
  perspective=analogical_letter
```

```
%%%ff%%%
```

You should be able to slot that in above the front leg
that --> type=piece, form=def_NP, perspective=unspecific
above the front leg --> type=location, form=pp,
perspective=analogical_elephant

And you just fit that into the first piece
that --> type=piece, form=def_NP, perspective=unspecific
into the first piece --> type=location, form=pp,
 perspective=relative_to_placed_pieces

Und dann kommt das sozusagen am Rcken hin
das --> type=piece, form=def_NP, perspective=unspecific
am Rcken --> type=location, form=pp, perspective=analogical_elephant

It's four units and then there is one kind of unit coming off it
It's four units and then there is one kind of unit coming off it
--> type=desc_piece, form=attribution_sent, perspective=analytical_blocks
it --> type=piece, form=pronoun, perspective=none
four units --> type=part_of piece, form=quant_NP,
perspective=analytical_blocks
one kind of unit coming off it --> type=part_of piece, form=quant_NP,
  perspective=analytical_blocks

I think it's like three units three units
I think it's like three units three units --> type=desc_piece,
 form=attribution_sent,
perspective=analytical_blocks
three units --> type=part_of_piece, form=quant_NP,
 perspective=analytical_blocks

Und das ist der Hinterlauf sozusagen --> form: other_sent
Und das ist der Hinterlauf sozusagen --> type=desc_piece,
  form=other_sent,
  perspective=analogical_elephant
das --> type=piece, form=def_NP, perspective=unspecific

Nee genau da muss es hin
da --> type=location, form=other,
 perspective=relative_to_current_position
es --> type=piece, form=pronoun, perspective=none

(Ja, da muss) es (hin) --> type=piece, form=pronoun, perspective=none

```
(Yeah,) it (should fit under it) --> type=piece, form=pronoun,
 perspective=none

This (is horizontally now) type=piece, form=pronoun, perspective=none


Gut, dann das Kreuz, das Dritte von oben
das Kreuz --> type=piece, form=def_NP,
  perspective=analogical_geometrical_shape
das Dritte von oben --> type=piece, form=def_NP, perspective=none

(Dis is) in der unteren Reihe das Dritte von links --> type=piece,
 form=def_NP,
 perspective=none

(Alright , so now where do I put) this fucker? --> type=piece,
form=def_NP,
perspective=unspecified

(Now, hopefully,) this other one (should fit) --> type=piece, form=def_NP,
  perspective=unspecified

Das (muss man aber erst noch irgendwie drehen) --> type=piece, form=def_NP,
perspective=unspecified

(Ok,) Nummer 1: --> type=piece, form=def_NP, perspective=name_number
Die 8 (ist dis, was oben links noch liegt) -->  type=piece, form=def_NP,
perspective=name_number

Alright, number nine is a U shape
number nine -->  type=piece, form=def_NP, perspective=name_number
a U shape -->  type=piece, form=indef_NP, perspective=analogical_letter

Is it the four vertical ones and then one?
the four vertical ones and then one --> type=piece, form=def_NP,
 perspective=analytical_blocks
the four vertical ones -->  type=part_of_piece, form=quant_NP,
 perspective=analytical_blocks
one -->  type=part_of_piece, form=quant_NP, perspective=analytical_blocks

It is like four in a line and then one off to the side
it --> type=piece, form=pronoun, perspective=none
four in a line --> type=part_of_piece, form=quant_NP,
 perspective=analytical_blocks
one off to the side --> type=part_of_piece, form=quant_NP,
```

```
perspective=analytical_blocks

Eine lange Saeule mit drei Teilen --> type=piece, form=indef_NP,
  perspective=analytical_blocks


Instead of putting the additional cube on top of the middle one .. .
you put it on one of the sides
the additional cube --> type=part_of_piece, form=def_NP,
perspective=analytical_blocks
the middle one -->  type=part_of_piece, form=def_NP,
  perspective=relative_to_current_position
one of the sides --> type=part_of_piece, form=quant_NP,
perspective=analytical_other


The horizontal bit of the original square shape that was in there
the horizontal bit of the original square that was in there
--> type=part_of_piece, form=def_NP,
 perspective=relative_to_current_position
the original square that was in there --> type=piece, form=def_NP,
perspective=analogical_geometrical_shape


It's an el shape
 it --> type=piece, form=pronoun, perspective=none
 an el shape --> type=piece, form=indef_NP,
  perspective=analogical_letter


We will say it is a C
it --> type=piece, form=pronoun, perspective=none
 a C --> type=piece, form=indef_NP, perspective=analogical_letter


Das Ganze sieht aus wie ein Z
Das Ganze sieht aus wie ein Z --> type=desc_piece, form=comp_sent,
 perspective=analogical_letter
das Ganze --> type=piece, form=def_NP, perspective=unspecific
ein Z --> type=comp_piece, form=indef_NP,
 perspective=analogical_letter


The top left of the elephant's back --> type=location, form=def_NP,
 perspective=analogical_elephant


It is like the back leg
it --> type=piece, form=pronoun, perspective=none
the back leg --> type=piece, form=def_NP,
  perspective=analogical_elephant
```

```
The elephant's trunk --> type=piece, form=def_NP,
 perspective=analogical_elephant

Into the head --> type=location, form=pp,
perspective=analogical_elephant

An den Hinterkopf --> type=location, form=pp,
 perspective=analogical_elephant

The angle goes right into the bottom right corner
the angle --> type=piece, form=def_NP,
  perspective=analogical_geometrical_shape
into the bottom right corner --> type=location, form=pp,
perspective=relative_to_board

Der grosse massive Teil --> type=part_of_piece, form=def_NP,
 perspective=analogical_geometrical_shape

Das Kreuz / The cross --> type=piece, form=def_NP,
  perspective=analogical_geometrical_shape

The stem --> type=part_of_piece, form=def_NP,
 perspective=analogical_other

The trunk of the T
The trunk of the T --> type=part_of_piece, form=def_NP,
  perspective=analogical_other
the T --> type=piece, form=def_NP,
 perspective=analogical_letter

That Tetris shape --> type=piece, form=def_NP,
 perspective=analogical_other

It is like a visor
it --> type=piece, form=pronoun, perspective=none
a visor --> type=piece, form=def_NP, perspective=analogical_other

It is a kind of strange shape
it --> type=piece, form=pronoun, perspective=none
a kind of strange shape --> type=piece, form=indef_NP,
 perspective=analogical_other

Into the bottom right corner --> type=location, form=pp,
perspective=relative_to_board
```

```
(And that goes) towards the left of the grid
that --> type=piece, form=def_NP, perspective=unspecific
towards the left of the grid --> type=location, form=pp,
perspective=relative_to_board

(Dann muss das linke untere Quadrat der [..] Form) in die fnfte Spalte
das linke untere Quadrat der momentan benutzten Form --> type=part_of_piece,
form=def_NP,
perspective=analytical_blocks
der benutzten Form --> type=piece, form=def_NP,
  perspective=relative_to_dialogue
in die fuenfte Spalte --> type=location, form=pp,
  perspective=relative_to_board

Und dis kommt in den oberen Teil
dis --> type=piece, form=def_NP, perspective=unspecified
in den oberen Teil --> type=location, form=pp,
  perspective=relative_to_board

Around the left hand of the fifth piece
Around the left hand of the fifth piece --> type=location, form=pp,
perspective=relative_to_placed_pieces
the fifth piece --> type=piece, form=def_NP,
  perspective=relative_to_dialogue

The one that's in between two and four
The one that's in between two and four --> type=piece, form=def_NP,
perspective=relative_to_placed_pieces
in between two and four --> type=location, form=pp,
 perspective=relative_to_placed_pieces
two --> type=part_of_piece, form=def_NP, perspective=name_number
four --> type=part_of_piece, form=def_NP, perspective=name_number

Neben das, was aussieht wie ne Treppe
Neben das, was aussieht wie ne Treppe --> type=location, form=pp,
perspective=relative_to_placed_pieces
das, was aussieht wie ne Treppe --> type=piece, form=def_NP,
perspective=analogical_other
ne Treppe --> type=comp_piece, form=indef_NP,
  perspective=analogical_other

(Genau, jetzt noch) nen Stck nach rechts --> type=location, form=quant_NP,
perspective=relative_to_current_position

Nach rechts (genau so) --> type=location, form=pp,
```

```
perspective=relative_to_current_position

(Oder noch) eins weiter nach hinten --> type=location, form=quant_NP,
perspective=relative_to_current_position

The top edge of that piece
The top edge of that piece --> type=part_of_piece, form=def_NP,
 perspective=relative_to_current_position
that piece --> type=piece, form=def_NP, perspective=unspecific

Das untere Ende vom Z
Das untere Ende vom Z --> type=part_of_piece, form=def_NP,
  perspective=relative_to_current_position
vom Z --> type=piece, form=def_NP, perspective=analogical_letter

Die Spitze vom M
Die Spitze vom M --> perspective: relative_to_current_position
vom M --> type=piece, form=def_NP, perspective=analogical_letter

The first piece is gonna go in the far bottom right corner
The first piece --> type=piece, form=def_NP,
  perspective=relative_to_dialogue
in the far bottom right corner --> type=location, form=def_NP,
  perspective=relative_to_board

The next one --> type=piece, form=def_NP,
  perspective=relative_to_dialogue

This one that could also possibly match that description --> type=piece,
 form=def_NP,
perspective=relative_to_dialogue


The one we had problems with earlier --> type=piece, form=def_NP,
  perspective=relative_to_dialogue

The second piece you have put in -->  type=piece, form=def_NP,
  perspective=relative_to_dialogue

the piece, that youre manipulating now --> type=piece, form=def_NP,
 perspective=relative_to_dialogue

(It is) a mirror shape --> type=piece, form=indef_NP,
perspective=relative_to_other
```

```
"The first piece looks like a Z"
      "the first piece" --> type=piece, form=def_NP,
perspective=relative_to_dialogue
      "a Z" --> type=comp_piece, form=indef_NP,
perspective=analogical_letter
      "the first piece looks like a Z" --> type=desc_piece,
form=comp_sent,
    perspective=analogical_letter

"Und es handelt sich dabei um eins, das man mit viel  gutem  Willen
als 'n Z bezeichnen knnte"
      "eins, das man mit viel  gutem  Willen als 'n Z bezeichnen knnte"
      --> type=piece, form=indef_NP,  perspective=analogical_letter
      "'n Z" --> type=piece, form=indef_NP, perspective=analogical_letter

"it looks like a C and it's right next to the cross"
      "it" --> type=piece, form=pronoun, perspective=none
      "a C" --> type=comp_piece, form=indef_NP,
  perspective=analogical_letter
      "it looks like a C" --> type=desc_piece, form=comp_sent,
 perspective=analogical_letter
      "it" --> type=piece, form=pronoun, perspective=none
      "next to the cross" -->  type=location, form=PP,
  perspective=relative_to_placed_pieces
      "the cross" --> type=piece, form=def_NP,
  perspective=analogical_geometrical_shape
      [note that here 'the cross' is not used as a comparison but
       directly refers to a piece; also note that the item_id here
       should be that of the cross-like piece, while the id of the
       other ref exps in this example should be the one of the C-like
       piece]

"das erste Teil ist zwischen den Beinen des Elefanten anzubringen"
      "das erste Teil" --> type=piece, form=def_NP,
  perspective=relative_to_dialogue
      "zwischen den Beinen des Elefanten" --> type=location, form=PP,
perspective=analogical_elephant

"Und das muss jetzt af dem lngeren Teil steh'n, aufrecht, so dass das
krzere Teil nach rechts schaut"
      "das" -->  type=piece, form=pronoun, perspective=unspecific
      "dem lngeren Teil" --> type=parts_of_piece, form=def_NP,
  perspective=analogical_geometrical_shape
      "das krzere Teil" -->  type=parts_of_piece, form=def_NP,
  perspective=analogical_geometrical_shape
```

```
"Also man hat dann dieses Dreiteilige links und dis Zweiteilige rechts,
und die Ding steht aufrecht"
    "dieses Dreiteilige" --> type=parts_of_piece, form=def_NP,
  perspective=analytical_blocks
    "dis Zweiteilige" --> type=parts_of_piece, form=def_NP,
  perspective=analytical_blocks
    "die Ding" --> type=piece, form=def_NP, perspective=unspecific

"it has one square sticking out"
    "it" -->  type=piece, form=pronoun, perspective=none
    "one square sticking out" --> type=parts_of_piece, form=quant_NP,
 perspective=analytical_blocks
    "it has one square sticking out" --> type=desc_piece,
  form=atribution_sent,
  perspective=analytical_blocks

"move it two squares up and one to the right"
     "it" -->  type=piece, form=pronoun, perspective=none
     "two squares up and one to the right" --> type=location,
   form=quant_NP,
perspective=relative_to_current_position
```

References

Allen, J. & Core, M. (1997*a*), Draft of DAMSL: Dialog act markup in several layers. Discourse Research Initiative.

Allen, J. & Core, M. (1997*b*), Draft of damsl: Dialog act markup in several layers. Discourse Research Initiative.

Boersma, P. (2001), 'Praat, a system for doing phonetics by computer', *Glot International* **5**(9–10), 341–345.

Meteer, M. & Taylor, A. (1995), Dysfluency annotation stylebook for the switchboard corpus. http://www.cis.upenn.edu/ bies/manuals/DFL-book.pdf.

Müller, C. & Strube, M. (2006), Multi-level annotation of linguistic data with MMAX2, *in* S. Braun, K. Kohn & J. Mukherjee, eds, 'Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods', Peter Lang, Frankfurt a.M., Germany, pp. 197–214.

Rodríguez, K. J. & Schlangen, D. (2004), Form, intonation and function of clarification requests in german task-oriented spoken dialogues, *in*

E. Vallduví, ed., 'Proceedings of Catalog (the 8th workshop on the semantics and pragmatics of dialogue; SemDial04)', Barcelona, Spain, pp. 101–108.
URL: *http://www.ling.uni-potsdam.de/ das/papers/rodrschl_catalog.pdf*