DiAML-Annotated Examples

Overview

This annex specifies the DiAML annotation of example dialogues and dialogue fragments. Section B.1 explains the annotation of some very short dialogue fragments, like question-answer pairs. Section B.2 contains the annotation of a complete spoken human-computer dialogue (from a Wizard-of-Oz experiment), and an extended fragment of a multimodal human-human dialogue.

1 Short dialogue fragments

Example (1) is a two-turn dialogue fragment, where each turn constitutes a single functional segment in the Task dimension. For the anchoring of DiAML annotations to the primary text, it is assumed that the functional segments are defined at another level of analysis (see Annex C) as have the XML identifiers "fs1" and "fs2", respectively. In this case, fs1 is everything said by participant P1 in turn 1, and fs2 is everything said by P2 in turn 2. In DiAML, the association of dialogue information with a functional segment is represented by means of the `target` attribute, following the TEI guidelines for text encoding (TEI, 2010). Similarly, the identification of the participants in the dialogue may be assumed to be part of the metadata for the primary dialogue data; hence the participants are represented here by the externally defined identifiers `p1` and `p2`.

This example shows the annotation of a functional feedback relation; participant P2 responds to P1's initial question by providing the information that P1 wants to obtain, hence this segment should be annotated as an `Answer` which has a functional dependence relation to P1's request.

(1) a. 1. P1: Where I should check in for Munich?
2. P2: For Munich go to counters 31 to 40.

```
<diaml xmlns:"http://www.iso.org/diaml/"/>
<dialogueAct xml:id="da1" target="#fs1"
  speaker="#p1" addressee="#p2"
  communicativeFunction="setQuestion" dimension="task"/>
```

b.

```
<dialogueAct xml:id="da2" target="#fs2"
  speaker="#p2" addressee="#p1"
  communicativeFunction="answer" dimension="task"
  functionalDependence="#da1"/>
</diaml>
```

An answer to a question may be argued to always entail positive feedback, since a question can also be answered successfully if it has been understood. Annotating entailed communicative functions is not necessarily, since by definitions they can be inferred, so if their markup would be desirable for some reason, they can be automatically added. In this example, one might argue that the repetition *"For Munich"* in the answer is an explicit feedback signal, showing that P2 understood that P1 said *"for Munich"*. Eliminating *"For Munich"* from P2's answer would seem rather awkward, however, so in this case no explicit feedback act has been annotated. See the next example.

Example (2) is again a question - answer pair. P1 again asks a question, but he does so in an indirect way. It might seem that P1 is asking whether P2 possesses the information when the next train to Utrecht leaves, but obviously what P1 really wants to know is when the next train to Utrecht leaves. As opposed to the direct question *" What time the next train to Utrecht leave?"*, which carries the assumption that the addressee is able to provide this information, the indirect formulation does not carry this assumption; it questions it. Such indirect questions are interpreted in this standard as conditional requests, as they are semantically equivalent to *Please tell me what time the next train to Utrecht leaves, if you know*. P1's question is therefore annotated in (2) as a dialogue act with the communicative function `Request` with the qualifier `conditional`.

The second turn is segmented into two overlapping functional segments: the first part of P2's utterance, *"The next train to Utrecht leaves"* is a repetition of the content part of P1's question, and is therefore viewed as an explicit feedback signal. A feedback dependence relation is annotated to indicate that this feedback concerns the functional segment in the first turn. The utterance as a whole, on the other hand, expresses the answer to P1's (indirect) question.one in the Auto-Feedback dimension and one in the Task dimension. The functional segment in the Task dimension is an answer to the question in the first turn, and it is qualified as 'uncertain' since the speaker signals his uncertainty about the correctness of the answer he provides.

(2) a.

1. P1: Do you know what time the next train to Utrecht leaves?
TA `fs1: Do you know what time the next train to Utrecht leaves?`

2. P2: The next train to Utrecht leaves I think at 8:32.
AuFB `fs2.1: The next train to Utrecht leaves`
TA `fs2.2: The next train to Utrecht leaves I think at 8:32.`

b.

```
<diaml xmlns:"http://www.iso.org/diaml/">
<dialogueAct xml:id="da1" target="#fs1"
   sender="#p1" addressee="#p2"
   communicativeFunction="request" dimension="task"
   conditionality="conditional"/>
<dialogueAct xml:id="da2" target="#fs2.1"
   sender="#p2" addressee="#p1"
   communicativeFunction="autoPositive" dimension="autoFeedback"
   feedbackDependence="#fs1"/>
<dialogueAct xml:id="da3" target="#fs2.2"
   sender="#p2" addressee="#p1"
   communicativeFunction="answer" dimension="task"
   functionalDependence="#da1"/>
</diaml>
```

Example (3) is a three-turn fragment of a dialogue from the HCRC Map Task corpus (Carletta et al., 1998). The example illustrates the use of general-purpose functions for addressing another dimension than that of the task. In turn 2, participant P2 checks that he understood the previous instruction correctly, producing a Check Question in the Auto-Feedback dimension. In turn 3, participant P1 confirms P2's understanding, thus addressing P2's processing of that same instruction, i.e. performing a Confirm act in the Allo-Feedback dimension.
Turn 3 has been segmented into two functional segments, the first (*"Yeah"*) considered as answering the question in the previous turn, and the second part as providing the additional information that the intended direction is only *"very slightly"* northeast, i.e. as an Inform act which elaborates the short *"Yeah"*. This is expressed in the annotation by a rhetorical relation to the preceding confirmation. Note that the representation of rhetorical relations in DiAML differs from that of fucitonal dependence relations and feedback dependence relations, due to their different semantic status (see Clause 11.3). Note also that this standard does not define a specific set of rhetorical relations; the value `"elaborate"` in (3b) is merely indicative for what might be done, given a set of rhetorical values such as those of the Penn Discourse Treebank or those discussed in Hovy & Maier (1993).

(3) a.

1. P1: Move up
TA `fs1: Move up`
2. P2: Slightly northeast?
AuFB `fs2: Slightly northeast?`
3. P1: Yeah very slightly.
AllFB `fs3.1: Yeah`
AllFB `fs3.2: very slightly`

```
     <diaml xmlns:"http://www.iso.org/diaml/">
     <dialogueAct xml:id="da1" target="#fs1"
       sender="#p1" addressee="#p2"
       communicativeFunction="instruct" dimension="task"/>
     <dialogueAct xml:id="da2" target="#fs2"
       sender="#p2" addressee="#p1"
       communicativeFunction="checkQuestion" dimension="autoFeedback"
       feedbackDependence="fs1"/>
b.   <dialogueAct xml:id="da3" target="#fs3.1"
       sender="#p1" addressee="#p2"
       communicativeFunction="confirm" dimension="alloFeedback"
       functionalDependence="#da2"/>
     <dialogueAct xml:id="da4" target="#fs3.2"
       sender="#p1" addressee="#p2"
       communicativeFunction="inform" dimension="alloFeedback"/>
     <rhetoricalLink dact="#da4" rhetoRelatum="#da2" rhetoRel="elaborate"/>
     </diaml>
```

Example (4) shows a two-turn fragment of a dialogue from the TRAINS corpus (Allen et al., 1994), which shows the use of a dimension-specific function (Correct Misspeaking) in the dimension of Partner Communication Management (PCM). Notice that a PCM act refers to something that is being said at that moment, as opposed to an allo-feedback act, which refers to what was said in a previous turn. Still, the relation between the Correct Misspeaking act and the functional segment that it refers to is of the same nature as the relation between a feedback act and its trigger, so we use the same 'feedback dependence' relation to indicate this relation.

(4) a. 1. P1: engine E3 is going to pick up the bananas, back to Avon, dro... [fs1]
2. P2: to pick up the oranges [fs2]

```
     <diaml xmlns:"http://www.iso.org/diaml/">
     <dialogueAct xml:id="da1" target="#fs1"
        sender="#p1" addressee="#p2"
        communicativeFunction="inform" dimension="task"/>
b.   <dialogueAct xml:id="da2" target="#fs2"
        sender="#p2" addressee="#p1"
        communicativeFunction="correctMisspeaking"
        dimension="partnerCommunicationManagement" feedbackDependence="#fs1"/>
     </diaml>
```

Example (5), taken from a dialogue in the AMI multi-party dialogue corpus, shows the use of general-purpose functions *Offer* and *Accept Offer*) in the Turn Management dimension, and the annotation of a functional dependence relation.

(5) a. 1. P1: Would you like to say something at this point? [fs1]
2. P2: Certainly. fs2]

```
     <diaml xmlns:"http://www.iso.org/diaml/">
     <dialogueAct xml:id="da1" target="#fs1"
        sender="#p1" addressee="#p2"
        communicativeFunction="offer" dimension="turnManagement"/>
b.   <dialogueAct xml:id="da2" target="#fs2"
        sender="#p2" addressee="#p1"
        communicativeFunction="acceptOffer" dimension="turnManagement"
        functionalDependence="#fs1"/>
     </diaml>
```

2 Annotated dialogues

2.1 Annotated human-computer dialogue

The dialogue which is analyzed here has been taken from the OVIS corpus of telephone dialogues[1] with an experimental system of the Dutch railways. The dialogues was originally in Dutch and has been translated into English.

In (6) the transcription of the dialogue is shown with its segmentation into functional segments and with an informal indication of its communicative function(s) and dimension(s). Segment boundaries are indicated by '|'. Horizontal lines have been added as turn separators for improved readability. The full annotation of the functional segments in DiAML is shown in (7). As in example (3) above, the annotation makes use of the rhetorical relation `elaborate`, which as such is not part of this standard, but which is used in order to show how rhetorical relations can be annotated in DiAML.

(6)

1.	S:	good morning | this is the automatic information system of Public		
		fs1.1	`SOM`	`InitialGreeting`
		Transport Travel Information		
		fs1.2	`SOM`	`SelfIntroduction`
		this system can only give information about train travel.		
		fs1.3	`Task`	`Inform`
		from which station to which station do you want to travel?		
		fs1.4	`Task`	`SetQuestion`
2.	U:	from Harwich to York.		
		fs2	`Task`	`Answer`
3.	S:	when do you want to travel from Harwich to York?		
		fs3.1	= "when do you want to travel from Harwich to York"	
		fs3.2	= "from Harwich to York"	
		fs3.1	`Task`	`SetQuestion`
		fs3.2	`AutoFb`	`autoPositive`
4.	U:	ehm... | Monday evening		
		fs4.1	`TimeM`	`Stalling`
		fs4.2	`Task`	`Answer`
5.	S:	so you want to leave between eighteen and twenty-four hours?		
		fs5	`AutoFb`	`CheckQuestion`
6.	U:	ehm | yes | Monday evening		
		fs6.1	`TimeM`	`Stalling`
		fs6.2	= "ehm yes"	
		fs6.2	`AlloFb`	`Answer [uncertain]`
		fs6.3	`AlloFb`	`Inform`
7.	S:	you said that you do not want to leave on Thursday April twenty-six between eighteen and twenty-four hours,		
		fs7.1	`AutoFb`	`inform`
		can you please correct me?		
		fs7.2	`AutoFb`	`Request`

[1] `http://www.let.rug.nl/vannoord/Ovis/`

8.	U:	yes | I would like to leave on monday...		
		fs8.1	`AlloFb`	`AcceptRequest`
		fs8.2	`Task`	`Inform`
		April twenty-six.		
		fs8.3	`Task`	`Inform`
9.	S:	what time do you want to travel the day after tomorrow?		
		fs9.1	= “what time do you want to travel the day after tomorrow?"	
		fs2.2	= “the day after tomorrow"	
		fs9.1	`Task`	`SetQuestion`
		fs9.2	`AutoFb`	`autoPositive`
10.	U:	at |ehm... | around eight o’clock		
		fs10.1:	at [..] around eight o’clock	
		fs10.1	`Task`	`Answer`
		fs10.2	`TimeM`	`Stalling`
11.	S:	so you want to leave at eight o’clock in the morning?		
		fs11	`AutoFb`	`CheckQuestion`
12.	U:	oh yes | sorry | eight o’clock in the evening.		
		fs12.1	`AutoFb`	`Inform`
		fs12.2	`SOM`	`Apology`
		fs12.3	`AlloFb`	`Disconfirm`
13.	S:	so you want to leave at twenty o’clock?		
		fs13	`AutoFb`	`CheckQuestion`
14.	U:	yes		
		fs14	`AlloFb`	`Confirm`
15.		I have found the following connection.		
		fs15.1	`DiscourseStructuring`	`InteractionStructuring`
		Departure from Harwich at nineteen fifty-six arrival in London Waterloo at twenty zero four, continuing from there with the train departing at twenty forteen arrival in Birmingham Central at twenty ten. Continuing from there with the train departing at twenty sixteen arrival in York at twenty forty.		
		fs15.2	`Task`	`Answer`
		would you like me to repeat the connection once more?		
		fs15.3	`AlloFb`	`Offer`
16.	U:	oh no | thank you		
		fs16.1	= oh no thank you	
		fs16.1	`AutoFb`	`DeclineOffer`
		fs16.2	`SOM`	`Thanking`
17.	S:	would you like to know another connection?		
		fs17	`Task`	`Offer`
18.	U:	do you have another connection?		
		fs18	`Task`	`PropositionalQuestion`
19.	S:	I will have a look for you		
		fs19.1	`TimeM`	`Promise`
.		no, I don’t think that there is something earlier		
		fs19.2	`Task`	`Answer [uncertain]`

20.	U:	is there an earlier possibility?		
		fs20	Task	PropositionalQuestion
21.	S:	I'm sorry | there's no earlier possibility.		
		fs21.1	SOM	Apology
		fs21.2	Task	Answer
22.	U:	thank you very much		
		fs22	SOM	Thanking
23.	U:	Public Transportation Information wishes you a pleasant journey		
		fs23.1	SOM	InitialGoodbye
		goodbye		
		fs23.2	SOM	initialGoodbye

(7) DiAML annotation:

```
<diaml xmlns:"http://www.iso.org/diaml/">
<dialogueAct xml:id="da1" target="#fs1.1"
   sender="#s" addressee="#u"
   communicativeFunction="initGreeting"
   dimension="socialObligationsManagement"/>
<dialogueAct xml:id="da2" target="#fs1.2"
   sender="#s" addressee="#u"
   communicativeFunction="selfIntroduction"
   dimension="socialObligationsManagement"/>
<dialogueAct xml:id="da3" target="#fs1.3"
   sender="#s" addressee="#u"
   communicativeFunction="inform" dimension="task"/>
<rhetoricalLink dact="#da3" rhetoRelatum="#da2" rhetoRel="elaborate"/>
<dialogueAct xml:id="da4" target="#fs1.4"
   sender="#s" addressee="#u"
   communicativeFunction="setQuestion" dimension="task"/>
<dialogueAct xml:id="da5" target="#fs2"
   sender="#u" addressee="#s"
   communicativeFunction="answer" dimension="task"
   functionalDependence"#da4"/>
<dialogueAct xml:id="da6" target="#fs3.1"
   sender="#s" addressee="#u"
   communicativeFunction="setQuestion" dimension="task"/>
<dialogueAct xml:id="da7" target="#fs3.2"
   sender="#s" addressee="#u"
   communicativeFunction="autoPositive" dimension="autoFeedback"
   feedbackDependence="#fs2"/>
<dialogueAct xml:id="da8" target="#fs4.1"
   sender="#u" addressee="#s"
   communicativeFunction="stalling" dimension="timeManagement"
<dialogueAct xml:id="da9" target="#fs4.2"
   sender="#u" addressee="#s"
   communicativeFunction="answer" dimension="task"
   functionalDependence="#da7"/>
```

```
<dialogueAct xml:id="da10" target="#fs5"
   sender="#s" addressee="#u"
   communicativeFunction="checkQuestion" dimension="autoFeedback"
   feedbackDependence="#fs4.2"/>
```

```
<dialogueAct xml:id="da11" target="#fs6.1"
   sender="#u" addressee="#s"
   communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da12" target="#fs6.2"
   sender="#u" addressee="#s"
   communicativeFunction="answer" dimension="alloFeedback"
   qualifier="uncertain" functionalDependence="#da10"/>
<dialogueAct xml:id="da13" target="#fs6.3"
   sender="#u" addressee="#s"
   communicativeFunction="inform" dimension="task"/>
```

```
<dialogueAct xml:id="da14" target="#fs7.1"
   sender="#s" addressee="#u"
   communicativeFunction="inform" dimension="autoFeedback"
   feedbackDependence="#fs6.3"/>
<dialogueAct xml:id="da15" target="#fs7.2"
   sender="#s" addressee="#u"
   communicativeFunction="request" dimension="autoFeedback"
   feedbackDependence="#fs7.1"/>
```

```
<dialogueAct xml:id="da16" target="#fs8.1"
   sender="#u" addressee="#s"
   communicativeFunction="acceptRequest" dimension="alloFeedback"
   functionalDependence="#da15"/>
<dialogueAct xml:id="da17" target="#fs8.2"
   sender="#u" addressee="#s"
   communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da18" target="#fs8.3"
   sender="#u" addressee="#s"
   communicativeFunction="inform" dimension="task"
<rhetoricalLink dact="#da18" rhetoRelatum="#da17" rhetoRel="elaborate"/>
```

```
<dialogueAct xml:id="da19" target="#fs9.1"
   sender="#s" addressee="#u"
   communicativeFunction="setQuestion" dimension="task"/>
<dialogueAct xml:id="da20" target="#fs9.2"
   sender="#s" addressee="#u"
   communicativeFunction="autoPositive" dimension="autoFeedback"
   feedbackDependence="#fs8.3"/>
```

```
<dialogueAct xml:id="da21" target="#fs10.1"
   sender="#u" addressee="#s"
   communicativeFunction="answer" dimension="task"
   functionalDependence="#da19"/>
<dialogueAct xml:id="da22" target="#fs10.2"
   sender="#u" addressee="#s"
   communicativeFunction="stalling" dimension="timeManagement"/>
```

```
<dialogueAct xml:id="da23" target="#fs11"
   sender="#s" addressee="#u"
   communicativeFunction="checkQuestion" dimension="autoFeedback"
   feedbackDependence="#fs10.2"/>
```

```
<dialogueAct xml:id="da24" target="#fs12.1"
   sender="#u" addressee="#s"
   communicativeFunction="inform" dimension="autoFeedback"
   feedbackDependence="#fs11"/>
<dialogueAct xml:id="da25" target="#fs12.2"
   sender="#u" addressee="#s"
   communicativeFunction="apology"
   dimension="socialObligationsManagement"/>
<dialogueAct xml:id="da26" target="#fs12.3"
   sender="#u" addressee="#s"
   communicativeFunction="disconfirm" dimension="alloFeedback"
   functionalDependence="#da23"/>
<dialogueAct xml:id="da27" target="#fs12.3"
   sender="#u" addressee="#s"
   communicativeFunction="infirm" dimension="task"
   functionalDependence="#da23"/>
```

```
<dialogueAct xml:id="da28" target="#fs13"
   sender="#s" addressee="#u"
   communicativeFunction="checkQuestion" dimension="autoFeedback"
   feedbackDependence="#fs12.3"/>
```

```
<dialogueAct xml:id="da29" target="#fs14"
   sender="#u" addressee="#s"
   communicativeFunction="confirm" dimension="alloFeedback"
   functionalDependence="#da26"/>
```

```
<dialogueAct xml:id="da230" target="#fs15.1"
   sender="#s" addressee="#u"
   communicativeFunction="interactionStructuring"
   dimension="discourseStructuring"/>
<dialogueAct xml:id="da31" target="#fs15.2"
   sender="#s" addressee="#u"
   communicativeFunction="answer" dimension="task"
   functionalDependence="#da4 da5 da6 da16 da17 da27"/>
<dialogueAct xml:id="da32" target="#fs15.3"
   sender="#s" addressee="#u"
   communicativeFunction="offer" dimension="alloFeedback"/>
```

```
<dialogueAct xml:id="da33" target="#fs16.1"
   sender="#u" addressee="#s"
   communicativeFunction="declineOffer" dimension="autoFeedback"
   functionalDependence="#da32"
   feedbackDependence="#15.3"/>
<dialogueAct xml:id="da34" target="#fs16.2"
   sender="#u" addressee="#s"
   communicativeFunction="thanking"
   dimension="socialObligationsManagement"/>
```

```
<dialogueAct xml:id="da35" target="#fs17"
   sender="#s" addressee="#u"
   communicativeFunction="offer" dimension="task"/>
```

```
<dialogueAct xml:id="da36" target="#fs18"
   sender="#u" addressee="#s"
   communicativeFunction="propositionalQuestion" dimension="task"/>
```

```
<dialogueAct xml:id="da37" target="#fs19.1"
   sender="#s" addressee="#u"
   communicativeFunction="inform" dimension="discourseStructuring"/>
<dialogueAct xml:id="da38" target="#fs19.2"
   sender="#s" addressee="#u"
   communicativeFunction="answer" dimension="task"
   qualifier="uncertain"/>
   functionalDependence="#da36"/>
```

```
<dialogueAct xml:id="da39" target="#fs20">
   sender="#u" addressee="#s"
   communicativeFunction="propositionalQuestion" dimension="task"/>
```

```
<dialogueAct xml:id="da40" target="#fs21.1"
   sender="#s" addressee="#u"
   communicativeFunction="apology"
   dimension="socialObligationsManagement"/>
<dialogueAct xml:id="da41" target="#fs21.2"
   sender="#s" addressee="#u"
   communicativeFunction="answer" dimension="task"/>
   functionalDependence="#da39"/>
```

```
<dialogueAct xml:id="da42" target="#fs22"
   sender="#u" addressee="#s"
   communicativeFunction="thanking"
   dimension="socialObligationsManagement"/>
```

```
<dialogueAct xml:id="da43" target="#fs23.1"
   sender="#s" addressee="#u"
   communicativeFunction="interactionStructuring"
   dimension="discourseStructuring"/>
<dialogueAct xml:id="da44" target="#fs23.2"
   sender="#s" addressee="#u"
   communicativeFunction="initGoodbye"
   dimension="socialObligationsManagement"/>
</diaml>
```

2.2 Annotated Map Task dialogue

The following excerpt from a dialogue in the HCRC Map Task corpus (Carletta et al., 1996), illustrates the occurrence of nonverbal and multimodal segments. There are several occurrences of heavy breathing (in or out) which may have a communicative meaning; in the transcription these are indicated by VOC_intbreath and VOC_outbreath, respectively, where the prefix VOC_ indicates that this is a sample of vocal (nonverbal) behaviour. In turn 11 there is an occurrence of a lip smacking gesture, indicated in the transcription similarly by LIPGES_lipsmack. In the latter case, the relevant functional segment of the sender's behaviour is multimodal, consisting of (1) a verbal segment, where the sender says *erm* in a very slow fashion, surrounded by periods of silence; (2) the smacking of the lips; and (3) heavily breathing in. This is an illustration of the phenomenon that a functional segment in general has several components consisting of sender behaviour in various communicative channels, together making up a multimodal unit. The encoding of functional segments is not part of the present standard, and may be determined by the particular annotation tool that is used, but might for example look as in (8). In this particular example the vocal (but nonverbal) behaviour and the characterisation of lip gestures are described simply by named values; in other cases, like head gestures, the representation will be more complex and involve the representation of several features such as duration, direction, speed, and number of repetitions.

(8)

```
<functionalSegment xml:id="fs1"
  textSegment="#ts1"
  vocalSegment="#vs1"
  lipSegment="#ls1"/>
<textSegment xml:id="ts1" from="w11.16" to="w11.17"/>
<vocalSegment xml:id="vs1" value="inbreath"/>
<lipSegment xml:id="ls1" value="lipsmack"/>
```

```
     1.  P1:  okay, | starting off, | we are .. above .. a caravan park
              fs1.1  TurnM:   turnTake
                     DS:      opening
(9)           fs1.2  DS:      interactionStructuring
              fs1.3  Task:    inform
     -----------------------------------------------------------------
     2.  P2:  mmhmm
              fs2    autoFb   autoPositive
     -----------------------------------------------------------------
     3.  P1:  we are going to go due south | NONVOC_noise ... | # |
              fs3.1  Task:    inform
              fs3.2  TimeM:   stalling
                     TurnM:   turnKeep
     -----------------------------------------------------------------
              straight south | ... and NONVOC_noise ... | then we're going to g–.
              fs3.3  OCM:     Self-Correction
              fs3.4  TimeM:   Stalling
                     TurnM:   TurnKeep
              fs3.5  Task:    Instruct
     -----------------------------------------------------------------
                     g– ..
              fs3.6  OCM:     Retraction
              fs3.7  turn
              turn straight back round and head north... past an old mill ...
              on the right ... hand side
              fs3.8: Task:    Instruct

     -----------------------------------------------------------------
     4.  P2:  VOC_outbreath ... | B due south and then back up again
              fs4.1  TurnM:   TurnTake
              fs4.1  TimeM:   Stalling
              fs4.2  AutoFb:  CheckQuestion
     -----------------------------------------------------------------
     5.  P1:  yeah | south and then straight back up again
              fs5.1  AlloFb:  Confirm
              fs5.2  Task:    Instruct
              with an old mill on the right
              fs5.3  Task:    Inform [Elaborate]
              and you're going to pass on the left-hand side of the mill
              fs5.4  Task:    Instruct [Elaborate]
     -----------------------------------------------------------------
     6.  P2:  right okay
              fs6    AutoFb:  AutoPositive
     7.  P1:  okay | and then we're going to turn ... VOC_inbreath east
              fs7.1  = “okay"
              fs7.2  = “and then we're going to turn [...] east"
              fs7.3  = “ ... VOC_inbreath"
              fs7.1  AutoFb:  AutoPositive
                     TurnM:   TurnGrab
              fs7.2  Task:    Instruct
              fs7.3  TimeM:   Stalling
                     TurnM:   TurnKeep
     -----------------------------------------------------------------
```

8. P2: mmhmm
fs8 AutoFb: AutoPositive

9. P1: not ... straight east ... slightly sort of northeast | ...
fs9.1 Task: Inform
VOC_outbreath ...
fs9.2 TimeM: Stalling
TurnM: TurnKeep

10. P2: s– | slightly northeast
fs10.1 TurnM: TurnGrab
fs10.2 AutoFb: CheckQuestion

11. P1: slightly slightly yeah | very slightly | VOC_inbreath ...
fs11.1 Task: Confirm
fs11.2 Task: Inform
fs11.3 TimeM: Stalling
TurnM: TurnKeep
and we're going to continue straight along ... GES_lipsmack
VOC_inbreath ... erm ... quite a wee distance
fs11.4 = "we're going to continue straight along [...] quite a wee distance on that course and then we're going to turn north again"
fs11.4 Task: Instruct
fs11.5 = "... GES_lipsmack VOC_inbreath ... erm ..."
fs11.5 TimeM: Stalling
TurnM: TurnKeep

12. P2: right | mmhmm
fs12.1 AutoFb: AutoPositive
fs12.2 AutoFb: AutoPositive

13. P1: and ... | immediat– | well |
fs13.1 TurnM: TurnTake
TimeM: Stalling
fs13.2 OCM: Retraction
fs13.3 TurnM: TunKeep
a distance below that turning point there's a fenced meadow
fs13.4 Task: Inform
| ... VOC_inbreath ... | but you should be avoiding that by quite
fs13.5 TimeM: Stalling
TurnM: TurnKeep
a distance
fs13.6 Task: Instruct

14. P2: okay
fs14 AutoFb: AutoPositive

15. P1: okay | so we've turned | and we're going up north again
fs15.1 AutoFb: AutoPositive
fs15.2 Task: Inform
fs15.3 Task: Instruct

(10) DiAML annotation:

```
<diaml xmlns:"http://www.iso.org/diaml/">
<dialogueAct xml:id="da1" target="#fs1.1"
   sender="#p1" addressee="#p2"
   communicativeFunction="turnTake" dimension="turnManagement"/>
<dialogueAct xml:id="da2" target="#fs1.1"
   sender="#p1" addressee="#p2"
   communicativeFunction="opening" dimension="discourseStructuring"/>
```

```
<dialogueAct xml:id="da3" target="#fs1.2"
   sender="#p1" addressee="#p2"
   communicativeFunction="interactionStructuring"
   dimension="discourseStructuring"/>
<dialogueAct xml:id="da4" target="#fs1.3"
   sender="#p1" addressee="#p2"
   communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da5" target="#fs2"
   sender="#p2" addressee="#p1"
   communicativeFunction="autoPositive" dimension="autoFeedback"
   feedbackDependence="#fs1.3"/>
<dialogueAct xml:id="da6" target="#fs3.1"
   sender="#p1" addressee="#p2"
   communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da7" target="#fs3.2"
   sender="#p1" addressee="#p2"
   communicativeFunction="turnKeep" dimension="turnManagement"/>
<dialogueAct xml:id="da8" target="#fs3.2"
   sender="#p1" addressee="#p2"
   communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da9" target="#fs3.3"
   sender="#p1" addressee="#p2"
   communicativeFunction="selfCorrection" dimension="ownCommManagement"/>
<dialogueAct xml:id="da10" target="#fs3.4"
   sender="#p1" addressee="#p2"
   communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da11" target="#fs3.4"
   sender="#p1" addressee="#p2"
   communicativeFunction="turnKeep" dimension="turnManagement"/>
<dialogueAct xml:id="da12" target="#fs3.5"
   sender="#p1" addressee="#p2"
   communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da13" target="#fs3.6"
   sender="#p1" addressee="#p2"
   communicativeFunction="retraction"
   dimension="ownCommManagement"/>
<dialogueAct xml:id="da14" target="#fs3.7"
   sender="#p1" addressee="#p2"
   communicativeFunction="selfCorrection" dimension="ownCommunicationManagement"/>
<dialogueAct xml:id="da15" target="#fs3.8"
   sender="#p1" addressee="#p2"
   communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da16" target="#fs4.1"
   sender="#p2" addressee="#p1"
   communicativeFunction="turnTake" dimension="turnManagement"/>
<dialogueAct xml:id="da17" target="#fs4.2"
   sender="#p2" addressee="#p1"
   communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da18" target="#fs4.3"
   sender="#p2" addressee="#p1"
   communicativeFunction="checkQuestion" dimension="autoFeedback"
   feedbackDependence="#fs3.6"/>
```

```
<dialogueAct xml:id="da19" target="#fs5.1"
   sender="#p1" addressee="#p2`'
   communicativeFunction="confirm" dimension="alloFeedback"
   functionalDependence="#da18"/>
<dialogueAct xml:id="da20" target="#fs5.2"
   sender="#p1" addressee="#p2"
   communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da21" target="#fs5.3"
   sender="#p1" addressee="#p2"
   communicativeFunction="inform" dimension="task"/>
<rhetoricalLink dact="#da21" rhetoRelatum="#da20" rhetoRel="elaborate"/>
<dialogueAct xml:id="da22" target="#fs5.4"
   sender="#p1" addressee="#p2"
   communicativeFunction="instruct" dimension="task"/>
<rhetoricalLink dact="#da22" rhetoRelatum="#da20"/>
```

```
<dialogueAct xml:id="da23" target="#fs6"
   sender="#p2" addressee="#p1"
   communicativeFunction="autoPositive" dimension="autoFeedback"
   eedbackDependence="#fs5.4"/>
```

```
<dialogueAct xml:id="da24" target="#fs7.1"
   sender="#p1" addressee="#p2"
   communicativeFunction="autoPositive" dimension="autoFeedback"
   feedbackDependence="#fs6"/>
<dialogueAct xml:id="da25" target="#fs7.1"
   sender="#p1" addressee="#p2"
   communicativeFunction="turnGrab" dimension="turnManagement"/>
<dialogueAct xml:id="da26" target="#fs7.2"
   sender="#p1" addressee="#p2"
   communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da27" target="#fs7.3"
   sender="#p1" addressee="#p2"
   communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da28" target="#fs7.3"
   sender="#p1" addressee="#p2"
   communicativeFunction="turnKeep" dimension="turnManagement"/>
```

```
<dialogueAct xml:id="da29" target="#fs8"
   sender="#p2" addressee="#p1"
   communicativeFunction="autoPositive" dimension="autoFeedback"
   feedbackDependence="#fs7.1"/>
```

```
<dialogueAct xml:id="da30" target="#fs9.1"
   sender="#p1" addressee="#p2"
   communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da31" target="#fs9.2"
   sender="#p1" addressee="#p2"
   communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da32" target="#fs9.2"
   sender="#p1" addressee="#p2"
   communicativeFunction="turnKeep" dimension="turnManagement"/>
```

```
<dialogueAct xml:id="da33" target="#fs10.1"
   sender="#p2" addressee="#p1"
   communicativeFunction="turnGrab" dimension="turnManagement"/>
<dialogueAct xml:id="da34" target="#fs10.2"
   sender="#p2" addressee="#p1"
   communicativeFunction="checkQuestion" dimension="autoFeedback"
   feedbackDependence="#fs9.1"/>
```

```
<dialogueAct xml:id="da35" target="#fs11.1"
   sender="#p1" addressee="#p2"
   communicativeFunction="confirm" dimension="alloFeedback"
   functionalDependence="#da34"/>
<dialogueAct xml:id="da36" target="#fs11.2"
   sender="#p1" addressee="#p2"
   communicativeFunction="inform" dimension="task"/>
<rhetoricalLink dact="#da36" rhetoRelatum="da35" rhetoRel="elaborate"/>
<dialogueAct xml:id="da37" target="#fs11.3"
   sender="#p1" addressee="#p2"
   communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da38" target="#fs11.3"
   sender="#p1" addressee="#p2"
   communicativeFunction="turnKeep" dimension="turnManagement"/>
<dialogueAct xml:id="da39" target="#fs11.4"
   sender="#p1" addressee="#p2"
   communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da40" target="#fs11.5"
   sender="#p1" addressee="#p2"
   communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da41" target="#fs11.5"
   sender="#p1" addressee="#p2"
   communicativeFunction="turnKeep" dimension ="turnManagement"/>
```

```
<dialogueAct xml:id="da42" target="#fs12.1"
   sender="#p2" addressee="#p1"
   communicativeFunction="autoPositive" dimension="autoFeedback"
   feedbackDependence="#fs11.5"/>
<dialogueAct xml:id="da43" target="#fs12.2"
   sender="#p2" addressee="#p1"
   communicativeFunction="autoPositive" dimension="autoFeedback"
   feedbackDependence="#fs11.5"/>
```

```
<dialogueAct xml:id="da44" target="#fs13.1"
   sender="#p1" addressee="#p2"
   communicativeFunction="turnTake" dimension="turnManagement"/>
<dialogueAct xml:id="da45" target="#fs13.1"
   sender="#p1" addressee="#p2"
   communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da46" target="#fs13.2"
   sender="#p1" addressee="#p2"
   communicativeFunction="retraction"
   dimension="ownCommunicationManagement"/>
<dialogueAct xml:id="da47" target="#fs13.3"
   sender="#p1" addressee="#p2"
   communicativeFunction="turnKeep" dimension="turnManagement"/>
<dialogueAct xml:id="da48" target="#fs13.4"
   sender="#p1" addressee="#p2"
   communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da49" target="#fs13.5"
   sender="#p1" addressee="#p2"
   communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da50" target="#fs13.5"
   sender="#p1" addressee="#p2"
   communicativeFunction="turnKeep" dimension="turnManagement"/>
<dialogueAct xml:id="da51" target="#fs13.6"
   sender="#p1" addressee="#p2"
   communicativeFunction="instruct" dimension="task"/>
```

```
<dialogueAct xml:id="da52" target="#fs14"
   sender="#p2" addressee="#p1"
   communicativeFunction="autoPositive" dimension="autoFeedback"
   feedbackDependence="#fs13.6"/>
<dialogueAct xml:id="da53" target="#fs15.1"
   sender="#p1" addressee="#p2"
   communicativeFunction="autoPositive" dimension="autoFeedback"
   feedbackDependence="#fs14"/>
<dialogueAct xml:id="da54" target="#fs15.2"
   sender="#p1" addressee="#p2"
   communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da55" target="#fs15.3"
   sender="#p1" addressee="#p2"
   communicativeFunction="instruct" dimension="task"/>
</diaml>
```