A Framework for Annotating Co-working Dialogues in Complex Task Settings

Emma Barker
Department of Computer Science
University of Sheffield
E.Barker@sheffield.ac.uk

Robert Gaizauskas
Department of Computer Science
University of Sheffield
R.Gaizauskas@sheffield.ac.uk

Abstract

There is immense potential for applications involving human-machine co-working. Building dialogue agents for co-working is a key challenge to realising this potential. We propose schemes to annotate co-working dialogues at two levels – the task level and the interaction level – in order to better understand the structure and nature of co-working dialogues and to provide the basis for annotating texts for training dialogue agent models. Our work is grounded in an analysis of part of the Apollo 11 air-to-ground mission transcripts.

1 Introduction

Many of human-kind's most impressive accomplishments – from building the Pyramids to landing a man on the moon or photographing a black hole – are the result of *co-working*, the coordinated activity of two or more human agents working together to achieve a common goal. Communication between co-workers is an essential part of co-working and in general the most efficient and effective form of communication for co-working is spoken dialogue. We refer to linguistic interaction between co-workers whose function is to directly facilitate or enable progress towards the common goal as *co-working dialogue*.[1]

Why do co-working agents A and B need to communicate? Reasons to do so include: to coordinate actions that need to be carried out in a certain order or at a certain time (e.g. A and B both need to push together to jump start the car); to provide or query prior knowledge that A has about the task and B does not (e.g. expert and novice mechanics working together on a car engine); to provide or check the perspective that A has and B does not (e.g. A is above the engine looking down, B underneath the engine looking up); to divide the roles in a multi-agent task or divide the tasks in a parallelisable task; to schedule tasks over a coming work period. Dialogue can also play an important role in maintaining social relations between co-workers, building trust, camaraderie, and so on, and therefore contributes indirectly to task completion; but here we focus on the task-oriented aspects of dialogue in co-working.

Why study co-working dialogues? As an important and ubiquitous sub-type of dialogue, co-working dialogues are, of course, worthy of study in their own right. However, there are also important practical reasons for studying them. Currently there is considerable excitement around the potential for human-machine co-working, where the machine may be a robot or a disembodied intelligent agent (cf. the Industry 4.0 vision (Hermann et al., 2016)). For example, a human and robot might work together in a manufacturing setting, where the robot is doing the heavy lifting under direction of a human; or, a human might be repairing a complex electrical or mechanical fault while an agent provides relevant information, e.g. schematic plans and instructions. In both these scenarios spoken dialogue would significantly increase the ease and effectiveness of the interaction. With advances in speech recognition spoken language interfaces are now becoming possible, but limited understanding of how to design intelligent co-working dialogue agents remains a major obstacle.

There has been substantial prior work on collecting and analysing extended human-human co-working dialogues – we review this in Section 5. However, this work has significant limita-

[1] A more common term is *task-oriented dialogue*. We view co-working dialogue as a sub-type of task-oriented dialogue, which includes only genuinely collaborative, task-focussed dialogue, excluding cases that could be deemed task-oriented but which are not genuinely collaborative, e.g. certain types of negotiation and debate, where one participant's gain is typically the other participant's loss.

tions with respect to the challenge of fully understanding co-working dialogue. First, the task settings studied are generally artificial and/or very restricted, in particular: (1) they typically involve single tasks, unlike many real world workflow settings where co-workers are involved in multiple, overlapping tasks and must switch between them (Lemon et al., 2002); (2) they are typically static, not dynamic, i.e the world does not change independently of the participants' actions during the dialogue, requiring an unanticipated shift of focus in the dialogue. Second, the analytical schemes developed to study these dialogues are limited in that they: (1) are typically designed for single task settings and do not distinguish between tasks in complex multi-task settings; (2) do not take into account linking to external task specification or domain ontology resources that are frequently available in complex real world task settings; (3) often focus on generic "dialogue acts", leaving interpretation of the content of utterances to a task-specific module, hence missing potential generalizations across interaction types that recur in many co-working settings.

In this paper we report our initial efforts to address these issues. First (Section 2) we identify a very substantial, publicly available real world corpus of co-working dialogues – the NASA manned space flight mission dialogues – in a setting where (1) there are multiple tasks to be carried out by multiple actors that may be sequential, concurrent or partially overlapping (2) tasks are co-ordinated in accordance with a high level pre-specified plan, and (3) the task environment is dynamic and only partially known, potentially throwing up unforeseen events or outcomes that may need to be dealt with immediately by unplanned activity and may require task rescheduling. Second (Section 3) we show how, by aligning dialogue transcripts from the corpus with an external task specification or plan, multi-task dialogues can be segmented into interleaved task-related chunks. We illustrate this through a case study in which two annotators separately annotate a 3 hour chunk of co-working dialogue and achieve high accuracy in both segment boundary identification and aligning tasks with the external pre-specified task plan. Third (Section 4) we propose an initial set of dialogue move or interaction types that capture not only the broad communicative function of utterances (e.g. "inform", "query", etc.) but also aspects of the semantics of utterances in co-working dialogues that we claim are generic across co-working settings. We illustrate these interaction types by means of examples taken from the corpus. Our motivating hypothesis here is that a generic co-working dialogue agent can be constructed that can interpret these interaction types in conjunction with external domain- and task-specific knowledge resources, such as ontologies and task or workflow specifications.

Together our proposals for task segmentation and interaction types form the basis of a novel annotation scheme for co-working dialogues. Applied at scale to real world co-working dialogue corpora this scheme can yield both data for training dialogue agents for complex co-working scenarios as well as deeper insights into co-working dialogue itself.

2 The NASA Manned Space Flight Program Data Resources

The US National Aeronautics and Space Administration (NASA), via the Johnson Space Center's History Portal, has made available audio recordings and transcripts of its entire manned space flight programme in the period 1961-1972, including air-to-ground and onboard conversations for all of the Mercury, Gemini and Apollo missions.[2] This is an incredible data resource, especially for investigating co-working dialogue, and much understudied in the computational linguistics community. The only prior work on this data by members of the CL community that we are aware of is Mann (2002), who considered a small excerpt from the Apollo 13 mission transcripts to illustrate his dialogue "macrogame" theory, but did not consider the resource more broadly or from the specific perspective of co-working. Clancey (2004) used a portion of the Apollo 17 mission transcripts ($\sim$ 1.5 hours) to investigate interactions between the ground-based NASA flight controller (CapCom) and the mission crew that took place during a series of lunar surface activities. He argued that the coordination role of the CapCom provided a model for future disembodied agent assistants working to support humans in similar remote working scenarios on Earth or in space. In particular, he identified various CapCom services that could be automated, such as taking logs, an-

[2] historycollection.jsc.nasa.gov/JSCHistoryPortal/history/mission_trans/all_transcripts.htm.

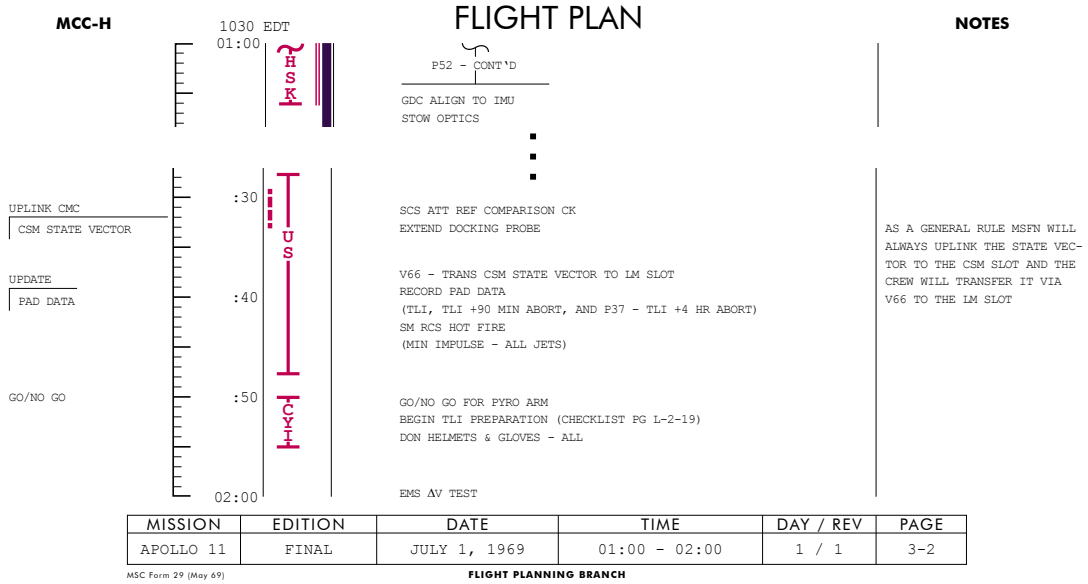

Figure 1: An excerpt from the Apollo 11 Flight Plan. Rule on left shows elapsed time in hours:minutes from launch. Next column to right (in red) shows which ground-based comms station will be handling communications with the spacecraft at that time. Middle column shows activities scheduled to be carried out at the specified time.

swering queries, and managing lunar surface activities and plans. This was a qualitative study, resulting in a fairly coarse grained analysis, and to the best of our knowledge was not developed more fully into a linguistic annotation scheme.

While not optimally curated for CL research, the total volume of dialogue data available far exceeds all other co-working dialogue resources put together and has the further advantage of being spontaneous and from a real task setting. Of course space flight is not a day-to-day experience, but our claim is that the patterns of co-working and co-working dialogue we see here are common to a multitude of other co-working settings.

The dialogues we have chosen to analyse and report on here are from Day 1 of the Apollo 11 mission. They are part of the air-to-ground interaction (onboard interactions are available separately) and have been transcribed and made available in both html and pdf form[3] as part of the Apollo 11 Lunar Surface Journal.[4] To help identify and label task threads (see Section 3) we used the NASA flight plan for the mission[5] a sample of which is shown in Figure 1. The flight plan is similar to workflow specifications found in many domains and task settings.

The air-to-ground transcripts provide a good example of remote co-working, where the parties in the dialogue are at different locations, have access to different sources of information and have different capabilities in terms of the task-related actions they can carry out.

We chose to analyse in detail an approximately 3 hour portion of the air-to-ground mission transcript from 00:01:34:33 to 00:04:28:49 (i.e from 0 days, 1 hour, 34 minutes and 33 seconds into the mission until just under 3 hours later). We refer to this corpus as the A11-MCC – Apollo 11 Mini Co-working Corpus. Each utterance in the transcripts is labelled with a time stamp and a speaker id. An example dialogue segment from the A11-MCC is shown in Figure 2.

3 Task Threads

We pursue a two-stage approach to analysing the extended NASA co-working dialogues. The first is to segment the dialogue into task-specific chunks. As is common in many real world task settings, our co-workers engage in multiple tasks in an overlapping fashion (e.g. a process may be set going, then something else done, then the process returned to for monitoring or adjustment). Unsur-

[3] hq.nasa.gov/alsj/a11/a11trans.html. Here we have used the html version.

[4] hq.nasa.gov/alsj/a11/a11.html This integrates all of the air-to-ground and onboard interactions, the NASA Public Affairs Officer commentary, including both transcripts and audio, and also includes additional helpful commentary and images. It is invaluable in providing context and background necessary to understand the dialogues.

[5] https://www.hq.nasa.gov/alsj/a11/a11fltpln_final_reformat.pdf

00 01 39 54	CDR	Houston, Apollo 11 is ready to go ahead with the - extend the docking probe,	EDP
		and ready to go with the RCS hot fire when you're ready to monitor. Over.	HF
00 01 40 06	CC	Roger. Go ahead with the probe, now.	EDP
00 01 40 13	CDR	Roger.	EDP
00 01 41 33	CDR	Okay. We're ready to - for the hot fire check when you're ready.	HF
00 01 41 39	CC	Roger. We're ready 11. Go ahead.	HF
00 01 41 48	CDR	Roger. Here's the pitch.	HF
00 01 42 13	CC	Apollo 11, this is Houston. We are seeing the pitch hot firing and it looks good.	HF
00 01 42 18	CDR	Roger. Be advised that we are unable to hear them.	HF
00 01 42 22	CC	Roger. We copy.	HF
00 01 42 24	CDR	Have you seen all three axes fire?	HF
00 01 42 31	CC	We've seen pitch and yaw; we've not seen roll to date.	HF
00 01 42 36	CDR	Okay. I'll put in a couple more rolls.	HF
00 01 42 42	CC	Okay. We've got the roll impulses, and you're looking good here.	HF
00 01 42 48	CDR	Roger. Houston, Apollo 11. We're standing by for a GO for sequence logic ON.	PA
00 01 43 03	CC	Apollo 11, this is Houston. Go ahead and we'll watch you on TM.	PA
00 01 43 07	CDR	Okay. Sequence logic, two of them. Sequence logic 1 and 2 coming up and ON.	PA
00 01 43 36	CC	Apollo 11, this Houston. You are GO for PRYRO ARM.	PA
00 01 43 40	CDR	Roger. Thank you.	PA
	⋮		
00 01 47 06	CC	. . . Would you verify that you have extended the probe? Over.	EDP
00 01 47 16	CDR	Roger. That's verified; the probe is extended.	EDP

Figure 2: A Short Sample of the Apollo 11 Air-to-Ground flight transcript. CDR = Commander (Armstrong); CC = Capsule Communicator (Mission Control). Final column is our addition and shows our mapping to activities in flight plan. EDP = Extend Docking Probe; HF = SM RCS Hot Fire; PA = Go/No Go for Pyro Arm

prisingly we find the dialogue pertaining to these overlapping tasks also overlaps. Therefore the task of separating the dialogue into task-specific chunks is not one of simple segmentation but one of identifying task-specific *threads*.

The second stage is to identify recurring *interaction types* in the dialogues we analyse. Here our methodology is one of iteratively analysing the interactions within a mission transcript, hypothesising interaction types with a view to them generalising across other domains, testing the hypotheses against the corpus and refining them to fit. We discuss this process further in Section 4.

3.1 Identifying and Annotating Task Threads

The flight plan (Figure 1) shows a list of tasks the astronauts are meant to be carrying out at each point in the mission. Of course in the event they are not able to stick exactly to schedule; also, some tasks get dynamically rescheduled by ground control. But the flight plan serves as a good guide to what is going on and provides labels for the tasks.

Two annotators (the authors) independently carried out the task segmenting the utterances in the A11-MCC into threads corresponding to a named task in the flight plan. At first glance the sample dialogue in Figure 2 appears to be an undifferentiated stream of mission-related conversation. But on more careful inspection and cross checking with the flight plan, sequences of turns can be aligned with activities in the flight plan (final column in Figure 2). Note the threaded nature of the task discussions: e.g., first, second and third turns mention the "Extend Docking Probe" task, which is then not mentioned again until the last two turns in the Figure, seven minutes later in the dialogue.

To date we have used an informal annotation scheme to mark up task threads. The key idea is to introduce an abstract "task" element that is realised by one or more "task segments" – sequences of turns where each utterance in the sequence pertains exclusively to a single task. This picture is complicated by the fact that some turns may refer to more than one task (e.g. the first turns in Figure 2). Thus, task-turn relation is many-to-many.

We plan to develop a concrete XML-based syntax consistent with other dialogue annotation formalisms, e.g. Bunt et al. (2012).

3.2 Results and Discussion

Following our double annotation of task threads in the A11-MCC corpus, we discussed divergences on a case-by-case basis and produced a consensus annotation.[6]. Some summary statistics on the consensus data set are presented in Table 1.

As can be seen from the table there were 243 turns across the 3 hour period examined in which

[6]This consensus version is available via the DOI: 10.5281/zenodo.3364099

Turns	Tasks	Segs Per Task			Turns Per Seg		
		Avg	Min	Max	Avg	Min	Max
243	23	1.52	1	3	5.51	1	26

Table 1: Task Threading in A11-MCC

23 tasks were discussed.[7] Additionally there was what we called a "COMS" task, which had to do with checking and assuring radio connectivity with various receiving sites on the Earth. Since a COMS-related task is not scheduled in the flight plan but is assumed ongoing across the whole mission[8], we did not count turns, or parts of turns, relating to COMS as a separate task or in computing segments per task or turns per segment. Such turns comprised 67 of the 243 turns in our corpus.

Of the 23 tasks identified 6 were deemed to be "Unscheduled", i.e. we could not confidently associate them with any task in the flight plan. Inter-annotator agreement was high, though we do not have precise quantitative agreement figures to report as the annotation exercise was a preliminary investigation of the feasibility of the scheme. There are two distinct tasks that can be assessed: one is determining the boundaries of the task segments and the other is the mapping from task segments to named tasks in the flight plan. Comparing the two annotators to the consensus "gold standard" we found that annotator1 correctly identified the boundaries for 44 out of 44 segments (including the COMS segments), while proposing 2 non-matching segments, for a recall and precision of 100%, while annotator 2 correctly identified boundaries for 42 out of 44 segments, while proposing 5 non-matching segments, for a recall of 95.5% and a precision of 89.4% (for each of the 2 missed segments the annotator proposed finer grained segmentation).

Considering the correct segments only, annotator 1 made 3 labelling errors for a labelling accuracy of 93.2%, while annotator 2 made 5 errors, for a labelling accuracy of 88.1%.

Thus, we are confident that task threads can be identified with high accuracy, especially the boundaries of task segments. Mapping these segments to the flight plan is a somewhat harder task as some technical knowledge in the domain is needed to understand, for instance, which particular parts of the spacecraft or particular readings, which may be the subject of conversation, are related to which tasks in the flight plan.

[7]Note that one turn may discuss more than one task, though in practice no turn ever contributed to more than two task segments. In counting turns per task segment, if a turn contributed to more than one task segment it was counted for each segment to which it contributed.

[8]As noted in the caption to Figure 1, the red vertical bars in the flight plan show through which terrestrial receiving site communications are meant to be passing at any given time in the mission, e.g. CYI = Canary Islands.

4 Interaction Types

Components of dialogue turns that have a specific task-related function in the interaction we refer to as *interaction elements*. Like some before us (see Section 5) we propose these interaction elements can be grouped into *interaction types*. However, our primary interest is not to categorise interaction elements by broad communicative intent (*inform*, *query*, etc.) but to type them according to the broad class of task activity to which they relate. Our hypothesis is that a general set of interaction types can be defined that reflect both the types of actions (e.g. assemble, check, configure) that are typically carried out in complex physical co-working contexts, such as manufacturing or space flight, and the meta-actions involved in their realisation (e.g. schedule, co-ordinate, check task status). If such a set of interaction types can be defined, then a generic co-working dialogue agent could be defined that could be readily specialised into a task-specific agent by coupling it with a domain-specific ontology and a task-specific workflow specification.

4.1 Task and Domain Modelling

To describe our proposed set of interaction types we presuppose the existence of a task and domain model, i.e. a model of the world in which the co-workers carry out their actions. We do not here want to articulate in detail such a model or to propose a preferred formal representation language for doing so. However, we do need to identify the principal types of components that domain and task models must contain, as our interaction types will be defined in terms of them. Specifically domain and task models must be able to represent:

1. *Objects, Attributes and Relations:* Objects are things that act and are acted upon in the task domain. They have attributes and stand in relations, which change over time. Agents are one type of object, as are docking probes, 0_2 valves, etc. It is useful to be able to distinguish object types and instances and to allow for the hierarchical arrangement of object types within a taxonomy.

Generic Conversational Interaction Types		
Interaction Type	Function	Example
Hail(R,S)	Sender S attempts to attract Receiver R's attention	"Apollo 11, this is Houston."
Acknowledge(R,S)	Receiver R confirms receipt of message to Sender S	"Roger."
Over(R,S)	Sender S informs receiver R that his transmission is complete	"Over."
Co-working Interaction Types		
Execute(G1,G2,Act,T*)	Agent G1 instructs agent G2 to execute activity Act at time T	"Go ahead with the probe, now." "You can start PTC at your convenience"
Configure(G1,G2,<O,A>,V,T*)	Agent G1 instructs Agent G2 to set the attribute A of object O to value V at time T	"We'd like at this time for you to place all four CRYO heaters to AUTO"
CoordinateActivity(G1,G2,Act1,Act2,T*)	Agent G1 requests Agent G2 to carry out activity Act2 at time T so that G1 can carry out Act1	"If you will give us P00 and ACCEPT, we have a state vector update for you." "When you are ready to copy, I have your TLI PAD."
AskPermission(G1,G2,Act,T)	Agent G1 asks Agent G2 for permission to do activity Act at time T	"We'd like to arm our logic switches."
ReportStatus(G1,G2,Act | <O,A>,T*)	Agent G1 reports to Agent G2 the status of activity Act or the value of attribute A of object O) at time T	"We have the PYRO's armed." "The REPRESS package valve is now in the OFF position".
ReportPlan(G1,G2, Act, T*)	Agent G1 informs Agent G2 that they are going to do activity Act at time T	"And, Buzz, we'll be terminating the battery charge in about a half hour."
QueryStatus(G1,G2,Act | <O,A>,T*)	Agent G1 asks Agent G2 to report the status of activity Act or the value of attribute A of object O at time T	"What have you been reading for O2 flow on your onboard gauge?"
CheckStatus(G1,G2,Act | <O,A>,RV,T*)	Agent G1 asks Agent G2 to confirm that the status of activity Act or the value of attribute A of object O at time T matches reference value RV	"Would you verify that you have extended the probe? "Would you confirm that your RCS heater switch for quad Bravo is in PRIMARY?"
Ready(G1,G2,Act,T*)	Agent G1 informs agent G2 that G1 is ready to begin activity Act at time T	"I am ready with your TLI-plus-90-minute abort PAD."
VoiceData(G1,G2,D)	Agent G1 reads out a block of data D to agent G2 (typically for G2 to copy down)	"P37 format, TLI plus 5: 00744 6485, minus 165, 02506."
ComparePerspective(G1,G2,<0,A>,V,T)	Agent G1 reports the value V of attribute A for object O at time T and invites Agent G2 to report the value he perceives	"And, Houston, looked like we saw about 87 or 88 psi on chamber pressure that time. I'd like you to look at that on the ground."

Table 2: Basic Interaction Types. *'ed arguments are optional with a default assumed if absent.

2. *Actions (or Activities)* In classical planning models (Fikes and Nilsson, 1971; Ghallab et al., 2016), actions have associated *preconditions* and *effects* and are specified in terms of the change they effect in the world, given that world is in a certain state when the action is performed and that state meets the action's preconditions. Actions may either be primitive or may specify a set of *sub-actions*, which must be performed for the higher level action to be accomplished. This recursive structure of actions is something we need for our account of co-working dialogues. As with objects, we need to type actions and distinguish action types from instances.

3. *Goals* Goals are distinguished states to be achieved or actions to be completed.

4. *Plans* Plans are sequences of actions, or partially ordered set of actions, which lead to a goal state or the completion of a goal action.

5. *Time* We require a model of actions, plans and goals in which time and temporal relations figure explicitly, since in many co-working situations scheduling of activities both relative to clock time and to each other is an essential part of what gets discussed.

As noted above, for current purposes we do not need to chose a particular formalism for representing task and domain models. There are,

however, several to choose from. These have emerged from the automatic planning community, which needs models of the world and of the tasks to be performed as input to the planning process (Fikes and Nilsson, 1971; Fox and Long, 2003; Gil, 2005; Ghallab et al., 2016)) and from the community focussed on exchange formats or standards for describing plans and activities in various real world domains, such as NIST's Process Specification Language (PSL) for manufacturing (Grüninger and Menzel, 2003) and the Shared Planning and Activity Representation (SPAR), sponsored by DARPA and the US Air Force for military planning[9].

4.2 Identifying and Annotating Interaction Types

Table 2 summarises our proposed set of co-working interaction types. It is divided into a list of generic converational interaction types and a list of co-working interaction types. In the first column we give the label and associated argument structure for that particular interaction type. In the second column, we explain this notation. The final column shows a dialogue segment which would be classified by this interaction type. So, for example, in the first row of the Co-working interaction types, we find "Instruct(G1,G2,Act,T)", a label that is applied to dialogue which communicates an instruction from Agent G1 to Agent G2, for G2 to do activity Act at time T. The example text includes "Go ahead with the probe now"; G1 is the CC; G2 is the CDR; Act is "the probe"; T is "now"; (note that the local dialogue context, as shown in Figure 1, reveals that the action referred to is "extend the docking probe"). The list of types we show in table 2 is not exhaustive. The co-working types were identified to accommodate the majority of dialogue in the A11-MCC (excluding the COMS segments); nonetheless our preliminary qualitative analysis of the entire Apollo 11 transcript suggests that these co-working types are applicable throughout the mission dialogue. In future work we plan to annotate a larger sample of dialogue taken from across the 8 day mission, and to extend this list where the data suggests there is a requirement for further co-working types. However, we believe the original list in Table 2 will form the majority of an extended co-working type set. Moreover, our belief is that these types will be applicable in other domains, such as automotive maintenance and cooking. For example, a Configure(G1, G2, <O,A>, V,T) could apply to a request from a cook to an assistant to "now set the oven to 200" or a CheckStatus(G1,G2,Act,<O,A>,RV,T) could describe a request from a mechanic for an apprentice to check that the clearance for a piston intake valve is within the range of 0.18-0.22mm.

Table 2 also lists a few examples of generic interaction types, e.g. "Hail", "Acknowledge". In future work we plan to extend this list, drawing from the extensive list of communicative functional types in the ISO 24617-2 dialogue act annotation standard in order to annotate more general features of task oriented dialogue such as communication management, feedback, turn taking, etc.

[9]See www.aiai.ed.ac.uk/project/spar/.

5 Related Work

In this section we review prior work on co-working dialogue corpora and on analytical frameworks for describing them. This review is not exhaustive but highlights key related work.

5.1 Previous Co-working Dialogue Corpora

An extensive review of dialogue corpora can be found at Serban et al. (2015). Here we focus solely on corpora of co-working dialogues.

Several dialogue corpora have been built with a view to studying dialogue in co-working settings. These include: the Map Task Corpus (Anderson et al., 1991), in which pairs of participants collaborate via spoken dialogue to reproduce a route draw on one map on another map; the TRAINS project co-working dialogue corpora, human-human conversations about managing the shipment of goods and movement of trains around a rail network (Allen et al., 1995; Heeman and Allen, 1995); and the AMI corpus of dialogues arising from, primarily, design team meetings (Carletta, 2007). In all these cases the corpora possess one or more of the shortcomings noted in Section 1: the task is artificial; the setting is static; in the case of AMI, dialogue arising from collective deliberation in meetings is very different from the sort of co-working dialogue that is the focus of work here, i.e. dialogue in settings where agents strive in real time to bring about a state of affairs in the physical world.

5.2 Analytical Frameworks for Dialogue

Task-based Dialogue Segmentation Grosz and Sidner (1986) propose segmenting dialogues ac-

cording to their *intentional* structure. In the examples they give, segments are recognised and labelled with intentions by human analysts. By contrast, in our case segments are determined by reference to an external task specification or plan. However, by adopting a plan an agent can be seen as forming an intention to execute each of the steps in the plan. Isard and Carletta (1995) segmented the Map Task dialogues into transactions by identifying sequences of dialogue that corresponded to the communication of a particular section of the route (i.e. a sub-task of the high level map task). While this work is similar to our approach, the resulting dialogue segments correspond to a single, artificial task type; we address multiple tasks as specified in a real world plan. Finally, in the AMI corpus dialogues are segmented by topic[10], using a set of domain-specific topics pre-specified by the corpus designers. This contrasts strongly with our task-based segmentation, where the tasks underlying the segmentation are provided in a plan devised not by corpus designers but originating in the real world context in which the dialogues occur.

Games and Moves Starting with Power (1979), there is a tradition of analysing dialogues in terms of *games* and *moves* (Kowtko et al., 1997; Lewin, 2000; Mann, 2002). Kowtko et al. (1997), for example, present a framework for analysing task-oriented dialogues which involves a two-level analysis in terms of *conversational games*, sequences of turns required to achieve a conversational sub-goal, and, at a lower level, *moves*, which are single, multiple or partial utterances that convey a single, specific intent. In annotation of the Map Task dialogues, they used six "initiating moves": *Instruct, Check, Align, Query y-n, Query-w, Explain* and six "response and feedback" moves: *Clarify, Acknowledge, Ready, Reply-Y, Reply-N, Reply-W*. There are some similarities between this work and our own, e.g. their *Instruct* move – "a direct or indirect request or instruction, to be done immediately or shortly" – is similar to our *Execute* interaction type. However, our moves are more grounded in the task – the arguments in our *Execute* serve to link to an external task model; our *Ready* interaction type is about communicating readiness to start a task while their *Ready* move is about conversants signalling readiness to take part in a conversation or game; their *Check* is to check a participants understanding of the communication, our *Checkstatus* is about checking that something in the external world is as it should be.

Dialogue Acts Much previous work has focussed on defining a set or hierarchy of dialogue acts, which are like moves as discussed above, but express a more finely nuanced descriptive framework for characterising different functional aspects of elements of dialogue (see, e.g., the Damsl dialogue act markup scheme (Allen and Core, 1997), the Switchboard dialogue act tagset (Stolcke et al., 2000) and the ISO 24617-2 dialogue annotation standard (Bunt et al., 2010, 2012)).

This work, particularly the ISO 24617-2 standard, proposes a rich, multi-dimensional approach to functional segment classification in dialogue The co-working interaction types we propose fall within the task dimension in the ISO standard. However our types provide a more detailed view of the communicative function of dialogue units, capturing the task semantics in a way that would allow an agent to interpret them in relation to an externally supplied model of the task and domain.

6 Conclusion and Future Work

In this paper we have taken initial steps towards defining a novel two level framework for analysing and annotating co-working dialogues. Key aspects of the framework are (1) the identification and annotation of task-specific threads within extended real world dialogues, which can be linked to external task specifications, and (2) the definition of a set of "interaction types", which recur across co-working dialogues and serve to identify both the communicative function of the linguistic unit in the co-working context and the elements within it which refer to objects, entities and activities in the task world. We illustrated both levels of the framework by reference to dialogues in the Apollo 11 air-to-ground mission transcripts, an invaluable source of real world co-working dialogues.

Going forward we intend first to validate the generality of our framework by applying it to a co-working corpus in another domain. We then plan to manually annotate a sufficient quantity of dialogue to train automatic annotators. Starting by modelling basic maintenance, repair and overhaul tasks in limited domains, we also intend to implement a co-working dialogue agent based on the framework put forward in this paper.

[10] http://groups.inf.ed.ac.uk/ami/corpus/annotation.shtml

References

James Allen and Mark Core. 1997. Draft of damsl: Dialog act markup in several layers contents.

James F. Allen, Lenhart Schubert, George Ferguson, Peter Heeman, Chung Hwang, Tsuneaki Kato, Marc Light, Nathaniel Martin, Bradford Miller, Massimo Poesio, and David Traum. 1995. The trains project: a case study in building a conversational planning agent. *Journal of Experimental & Theoretical Artificial Intelligence*, 7(1):7–48.

Anne H. Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, Catherine Sotillo, Henry S. Thompson, and Regina Weinert. 1991. The hcrc map task corpus. *Language and Speech*, 34(4):351–366.

Harry Bunt, Jan Alexandersson, Jean Carletta, Jae-Woong Choe, Alex Chengyu Fang, Koiti Hasida, Kiyong Lee, Volha Petukhova, Andrei Popescu-Belis, Laurent Romary, Claudia Soria, and David Traum. 2010. Towards an ISO standard for dialogue act annotation. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Languages Resources Association (ELRA).

Harry Bunt, Jan Alexandersson, Jae-Woong Choe, Alex Chengyu Fang, Koiti Hasida, Andrei Petukhova, Volha an Popescu-Belis, and David Traum. 2012. ISO 24617-2: A semantically-based standard for dialogue annotation. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 430–437, Istanbul, Turkey. European Language Resources Association (ELRA).

Jean Carletta. 2007. Unleashing the killer corpus: experiences in creating the multi-everything ami meeting corpus. *Language Resources and Evaluation*, 41(2):181–190.

W. J. Clancey. 2004. Roles for agent assistants in field science: Understanding personal projects and collaboration. *Trans. Sys. Man Cyber Part C*, 34(2):125–137.

Richard E. Fikes and Nils J. Nilsson. 1971. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3):189 – 208.

Maria Fox and Derek Long. 2003. Pddl2.1: An extension to pddl for expressing temporal planning domains. *J. Artif. Int. Res.*, 20(1):61–124.

Malik Ghallab, Dana Nau, and Paolo Traverso. 2016. *Automated Planning and Acting*. Cambridge University Press.

Yolanda Gil. 2005. Description logics and planning. *AI Magazine*, 26(2):73–84.

B. J. Grosz and C. L. Sidner. 1986. Attention, intentions and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

Michael Grüninger and Christopher Menzel. 2003. The process specification language (psl) theory and applications. *AI Mag.*, 24(3):63–74.

Peter A. Heeman and James Allen. 1995. The trains 93 dialogues. Technical Report 94-2, Computer Science Dept., University of Rochester.

M. Hermann, T. Pentek, and B. Otto. 2016. Design principles for industrie 4.0 scenarios. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pages 3928–3937.

Amy Isard and Jean Carletta. 1995. Replicability of transaction and action coding in the map task corpus. In *In AAAI 1995 Spring Symposium Series: Empirical Methods in Dicourse Interpretation and Generation*, pages 60–66.

Jacqueline C. Kowtko, Stephen D. Isard, and Gwyneth M. Doherty. 1997. Conversational games within dialogue. Technical Report HCRC/RP-31, University of Edinburgh.

Oliver Lemon, Alexander Gruenstein, Alexis Battle, and Stanley Peters. 2002. Multi-tasking and collaborative activities in dialogue systems. In *Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue - Volume 2*, SIGDIAL '02, pages 113–124, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ian Lewin. 2000. A formal model of conversational game theory. In *Proc. Gotalog-00, 4th Workshop on the Semantics and Pragmatics of Dialogue*.

William C. Mann. 2002. Dialogue macrogame theory. In *Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue - Volume 2*, SIGDIAL '02, pages 129–141, Stroudsburg, PA, USA. Association for Computational Linguistics.

Richard Power. 1979. The organisation of purposeful dialogues. *Linguistics*, 17:107–152.

Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. 2015. A survey of available corpora for building data-driven dialogue systems. *CoRR*, abs/1512.05742.

Andreas Stolcke, Noah Coccaro, Rebecca Bates, Paul Taylor, Carol VanEss-Dykema, Klaus Ries, Elizabeth Shriberg, Daniel Jurafsky, Rachel Martin, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Comput. Linguist.*, 26(3):339–373.