Behaving according to protocol: How communicative projects are carried out differently in different settings

Ellen Breitholtz & Christine Howes
University of Gothenburg,
Department of Philosophy, Linguistics and Theory of Science; CLASP
ellen.breitholtz@ling.gu.se christine.howes@gu.se

Abstract

There are a number of theories and models for capturing the aspects of organisations that are systematically related to the modes and genres of communication taking place within them. In this paper we will consider the micro-level of organisations and present a model of how similar communicative projects are carried out differently within different activities. Central to our account is the notion of conversational games, which can be seen as strategies for realising communicative projects while assigning speaker roles to dialogue participants.

1 Introduction

It is well established in the literature on organisational communication that the type of organisation affects the type of communication occurring in it, and vice versa (Baker, 2007; Brown and Starkey, 1994; Yates and Orlikowski, 1992). This has lead to a number of theories and models for capturing the aspects of organisations that are systematically related to the modes and genres of communication that take place within them.

In this paper we will consider the micro-level of organisations and present a model of how particular interactions play out within social activities and communicative projects of certain types. In section two we will discuss some categories that have been used to analyse interactions and to define different types of interaction. In Section 3 we will show how these categories can be implemented in a formal theory of dialogue. In Section 4 we consider the dialogue moves involved in a particular conversational game of the type in which a suggestion is made. We consider two different dialogues where this game is being realised in two different ways depending on contextual parameters. Finally, in Section 5, we draw some conclusions

2 Defining interactions

There are several areas of research which aim to categorise interactions in ways that are predictive of their communicative (including linguistic) features. These theories are based on a variety of concepts such as (social) (communicative) *activity* (Allwood, 2000), (communicative) *project*, *frame* (Levin and Moore, 1977; Carlson, 1982), (language) (dialogue) *game* (Lewis, 1979; Ginzburg, 2012), *genre*, etc. In this section we provide a brief overview of some of these concepts and theories.

When defining genres, types of language use etc. a frequently used concept is that of *activity*, as in the activity in the context of which language occurs. A communicative activity can be described as a comprehensive communicative project tied to a socio-cultural situation type reminiscent of the Wittgensteinian concept "form of life" (Allwood, 2000; Malcolm, 1954). On Allwood's account an activity type is characterised by the *goals*, *roles*, *artefacts* and *environment* that are associated with it. The carrying out of an activity consists of a number of sub-goals being completed. These may be more or less communicative in nature. For example, instances of the activity type "Buying/selling coffee in a café" are made up of sub-goals such as "conveying which product one wants to order", "conveying how much the costumer should pay", and, finally, "paying/receiving money".

Linell (2009) also draws on activity types for analysing interactions, but he emphasises the *communicative projects* that make up activity types. Communicative projects are often strongly associated with the sub-goals of particular activities. Thus in the café-interaction, the goal "conveying which product one wants to order" is linked to a project like "establishing an order". Another notion that has been used to define different classes of interactions is the concept of *genre* (see for example Ginzburg, 2016).

Proceedings of the 28th Workshop on the Semantics and Pragmatics of Dialogue, September, 11–12, 2024, Trento, Italy.

In the context of literature the concept of genre attempts to capture notions of subject matter, content and vocabulary as well as style, and it is used in a similar way in linguistic theory. Thus the genre "conversation in a bakery" is a monolithic type that involves specific vocabulary as well as grammatical constructions and dialogue moves. This way of thinking about interactions is intuitive and can be useful, and offers a "blueprint" of the characteristics of particular types of interactions. However, what is missing is a way of capturing differences and similarities between interactions that are directly related to specific contextual parameters. For example, an interaction in any shop or institution where a customer or client talks to a sales assistant or receptionist is likely to involve a lot of greetings and good byes, but the vocabulary and level of formality might vary depending on what kind of shop or other institution we consider. Thus we treat 'activity type' and 'project type' as independent categories, where one project type may be embedded in many different activity types – although not necessarily carried out by means of identical conversational strategies. This last fact prompts us, following Breitholtz (2020), to introduce an additional category by means of which to define interactions – that of *conversational game*, reminiscent of Wittgenstein's language game.

We think of a conversational game as a procedure for carrying out a particular project, and depending on the context different games may be available to do that. In addition, a particular game may have different possible moves available at each point in time. For example, making a decision might in some contexts play out in terms of one person suggesting something, another dialogue participant asking a follow up question, receiving an answer and then accepting, or it could play out as one dialogue participant simply telling the other(s) what to do. Which of these is most likely depends to a great extent on the activity the exchange is part of. Consider for example the dialogues below:

(1) BNC HM6 189-192

U I propose that Sir Simon [last or full name], a director retiring by rotation <pause> be and if hereby re-elected a director of the company.

M Put the resolution to the meeting.
Those in favour <pause> any against <pause> thank you.
I declare the resolution carried.

(2) BNC FM2 167-187

A I was thinking of asking Monica if she could record something from the the Model Car Club and

W Yeah.

A their club meetings.

W Oh that'd be okay I think.

V Is that <pause> Monica?

A Yes.

V Erm she's alread already asked her Dad but

A Right.

V but they don't have <pause> meetings. They they meetings take place during the evening er as very sort of <pause> at the <unclear>

A Right. Okay.

V <unclear> it wouldn't work. And we've

A Right.

V missed the A G M.

A Okay.

V It was earlier this year.

A Fair enough.

(1) is an excerpt from a formal meeting where proposals are formally made according to particular rules, while the dialogue in (2) is more informal and several participants are contributing acceptances, clarification requests, etc. However, a game type revolving around a suggestion or proposal is relevant to both of these dialogues. We will refer to this type of game as the *suggestion game*. Basic requirements on such a game are that there are at least two participants, one of whom makes a suggestion regarding some project that is believed to be shared. The other agent (or agents) responds to the suggestion, for example by *accepting* or by *rejecting* the proposal. We refer to whoever makes the first move (that is makes a suggestion) as player 1'. This move may optionally be followed by a motivation for the suggestion, again by player 1. Another player (player 2) may respond to the suggestion by accepting or rejecting the move. Note that this move does not necessarily have to be an actual response. Depending on the level of grounding we find acceptable in the context, abstaining from protesting might be enough to signal acceptance of a given suggestion (as in (1) where agreement and disagreement is signalled by non-verbal means). '

This way of thinking about acceptable moves in a dialogue is analogous to the way syntax is

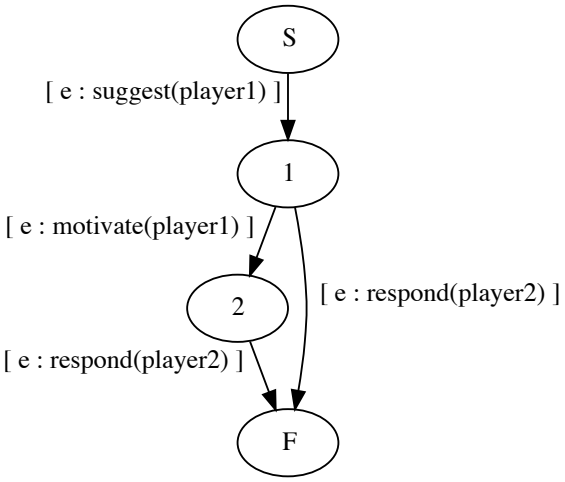

Figure 1: The Suggestion game

viewed in Dynamic Syntax (DS; Kempson et al., 2001; Cann et al., 2005), which is an action centred approach that takes the processes of building up interpretations in interaction as central to how language is used. In recent formulations of DS, the possibilities for future actions are represented as a Directed Acyclic Graph (DAG; see e.g. Howes and Eshghi, 2021), which constrains the space of possible transitions (moves) both within an utterance and between utterances, by restricting the range of (probabilistically) predictable next words or actions. Our formulation of specific games which become relevant at certain points within a dialogue – such as the suggestion game shown in Figure 1 serves the same purpose in making certain follow-up moves more likely in a given context. The types of games available may be further specified according to particular conventions of the organisations in which they occur. For example (1) occurs in the formal business setting of an Annual General Meeting, in which certain conventions apply, such as assenting to (or dissenting from) suggestions by raising your hand, with the majority vote prevailing. In contrast, (2) occurs in a less formal meeting between transcribers of the British National Corpus, and as such is less conventionally structured. This also means that the same sorts of actions (such as a suggestion being made and then either taken up or rejected) play out in different ways, despite being underpinned, in some sense, by the same game.

The moves in Figure 1 would suffice to account for an interaction where Player 1 makes a suggestion – optionally followed by a motivation – to player 2, who responds by either accepting or rejecting it. However, we would also like to allow for less straightforward rounds of the suggestion game, involving things like clarification requests, questions regarding other aspects of the context, etc, as in (2). For example, player 2 might ask for a reason for suggesting φ. This would be perfectly acceptable dialogue behaviour, and players must be allowed, within the suggestion game, to move into games of other types like the *clarification game*.

The ability to move between games reflects the expectations we have when engaging in dialogue – if you ask someone a question, you know that it is likely that you will get a response. However, we can still account for dialogue behaviour which does not conform to one particular game, since we allow dialogue participants to introduce new games – and even new projects.

We therefore want rules which allow for the suggestion game to be played in a number of different ways, including detours into other games. But let us leave that aside for the moment, and just consider the possibilities realised in (2).

3 Updating the Dialogue Game Board

For our model we will use TTR, a type theory with records (Cooper, 2005, 2014, 2023). TTR is a rich type theory, which can account for a range of linguistic phenomena, including many which are particular to dialogue (Cooper, 2005; Ginzburg, 2012; Cooper and Ginzburg, 2015; Lücking, 2016) Two key notions in TTR for dialogue are *Information State Update* (ISU), introduced by (Larsson and Traum, 2000) and (Larsson, 2002), and *Dialogue Gameboard* (DGB) Ginzburg (1994, 1996, 2012). The ISU approach considers the information states of each dialogue participant and how these are updated based on moves in the dialogue. Following Ginzburg, we will model information states as DGBs keeping the "score" of the dialogue in terms of sets of moves, presuppositions, questions, commitments, and other linguistic features which are relevant in the interaction.[1]

TTR is based on the capability in humans (and animals) to perceive and classify the world in terms of categories or *types*.

[1]The notion of game as a metaphor for conversation is not uncommon, see for example Wittgenstein (1953) and (Lewis, 1979).

Formally, the judgement that a particular object, a, is of a certain type T, can be described as $a : T$. The basic type of objects such as humans, animals and things in TTR is *Ind*, the type of individuals.[2]

In TTR *record types* are used in order to represent complex situations which involving many ptypes and individuals. A record type is a structure of pairs of labels and types. Often, the same letters that are used as those used to represent individual variables in other systems – 'x', 'y', etc.– are used as labels associated with the type *Ind*. For *constraints* on the type of situation represented by the record type we use the label 'c' with different subscripts. In (3) we see a record type representing a type of situation where a cat purrs.

(3)
$$\begin{bmatrix} \text{x} : \textit{Ind} \\ \text{c}_{\text{cat}} : \text{cat(x)} \\ \text{c}_{\text{purr}} : \text{purr(x)} \end{bmatrix}$$

The label 'x' in (3) points to an object of type *Ind*, and there are two constraints on the type of situation, namely that this individual is a cat (c_{cat}:cat(x)) and it purrs (c_{purr}:purr(x)).

In order to account for dialogue in TTR we use a version of the DGB which largely follows previous work (Ginzburg, 2012)[3] Following Cooper (2023) we treat the type of the information state of a conversational participant (the DGB of that participant) as a record type.

We think of the information state of an agent engaged in dialogue as comprising two types of information – that taken to be shared by the dialogue participants (similar to common ground Stalnaker, 1978; Clark et al., 1991) in the situation at hand, and the information taken to be private. As an example, let us assume that the type of an agent's shared information state is T_s and the type of the same agent's private information state is T_p. We see the type of that agent's information state in (4).

(4)
$$\begin{bmatrix} \text{private} : T_p \\ \text{shared} : T_s \end{bmatrix}$$

If we want to represent a "suggestion dialogue" in terms of updates of information states, we need rules handling not only the explicit moves represented in 1, but also *tacit* updates of the DGB. Tacit moves within a game represent inferences and other internal processes. We will now have a look at some of the updates of the DGBs of some of the participants throughout (2).

4 Analysing a suggestion dialogue

4.1 Initial tacit moves

In order to account for communicative games on the DGB we introduce the field *games*.

It is not until the first move is made, and thus considered shared by the conversational participants, that which game is being played can be expected to be shared, and thus appear on the shared game board.

At the beginning of the interaction in (2) the DGB of dialogue participant A is empty apart from the *project*, which we assume to be shared since the necessity of finding some data to record is obvious to both A and their interlocutors (who we shall refer to collectively as B in what follows for the sake of simplicity) in the context of the meeting. We represent a project as a type of event to be brought about by a number of agents. In (5) we see the type of a decision project, $T_{DecisionProject}$. $A_1, \ldots, A_n$ are dialogue participants and *Issue* the thing that is to be decided upon.

(5) $T_{DecisionProject}$ =
$$\left[\text{e} : \text{decide}\left(\{A_1, \ldots, A_n\}, \textit{Issue}\right)\right]$$

To allow representation of sequences of projects, fulfilling some complex goal (linguistic or other), the type *project* on the DGB is list(*RecType*). We would also be able to account for projects suddenly appearing in the information states of dialogue participants due to sudden events, such as "find shelter from the rain".

(6)
$$\begin{bmatrix} \text{private} : \textit{RecType} \\ \text{shared} : \left[\text{project} = \left[\left[\text{e} : \text{decide}(\{A, B\}, \text{data})\right]\right] : \text{list}(\textit{RecType})\right] \end{bmatrix}$$

(6) shows the type of the speaker, A's, information state at the beginning of the interaction in (2). For now we are interested only in the information state of dialogue participant A, not that of the listener, dialogue participant B. "Data" represents the issue of which data to collect, in (2).

The first update of the dialogue gameboard is an update of 'private games', that is the repository of conversational games which are salient with respect to a dialogue participant in a given context. Before

[2] *Ind* corresponds to *entity*, *e*, in Montague semantics (Montague, 1973).

[3] For a background on gameboard semantics in TTR the interested reader is referred to Ginzburg (2012) and Cooper (2023).

we move on to how we want to represent this update in TTR, let us have a look at the nature of projects and games in terms of types.

As illustrated in (5) we perceive a project as a record type representing the type of an event where a number of individuals (in this case A and B), jointly perform some action (in this case making a decision) regarding some non-decided-upon issue.

We may think of the development of a conversation as a finite state automaton where the arrows leading from one state to another correspond to the linguistic moves of the conversation, as represented in Figure (1). Instead of focusing on the states between the moves, we could focus on the sequence of moves themselves when defining a conversational game. We would then get a string of move types. The type in (7) for example, is of strings of moves comprising the type of *suggestion game*, $T_{SuggestionGame}$, – a suggestion by player 1 followed by an optional *motivation* by player 1, followed by a *response* (acceptance or rejection) by player 2. We represent move types as record types. A game of the type in (7) is made up of a suggestion, followed by an optional motivation by the dialogue participant who made the suggestion, followed by a response (either an *accept*- or a *reject* move) by the other player.

(7) $T_{SuggestionGame}$=$\left[\text{e : suggest(player1)}\right]^\frown$
$\left[\text{e : motivate(player1)}\right]^{\leqslant 1\frown}$ $\left[\text{e : respond(player2)}\right]$

The notation $\left[\text{e : motivate(player1)}\right]^{\leqslant 1}$ means that the suggestion move is followed by at most one motivation move ($^{\leqslant 1}$). [4] The string in (7) represents the type of a suggestion game on an abstract level – from this type we learn the sequence of move types involved and the relation between the *roles* that are necessary to play the game. However, in order for the game to work as a motor in the dialogue driving the updates, we need to assign the roles of the game to the individuals present in the context. For example, the player who initiates the game by making a suggestion has to be distinct from the player who acknowledges that suggestion.

4.2 Rules for updating private games

There are at least two different scenarios which would lead to an update of private games. First, there is the type of situation where the presence of a project on the DGB causes an agent to search his long term memory for a strategy by which to carry out that project, and load it onto the DGB. The second is when there is already a game on private games that would suffice to carry out the project. Assume for example that A has been thinking since he got out of bed in the morning that he wants to ask Monica to record some meetings from the Model Car Club. He has been meaning to suggest it for a while (or maybe hoping that B will suggest it), thus the suggestion game is activated on his private DGB. When A and B reach the point at which the issue of which data to collect becomes necessary to address, the project appears on the shared DGB. In this case the only update necessary on A's DGB is to place $T_{SuggestionGame}$ first in the list of games, while B has to retrieve the game from long term memory and load it onto private games. The idea here is that the update rules are combined with a control algorithm selecting which rule to apply in a given context. In Figure (2) we see a visualisation of the algorithm controlling the update of private games.

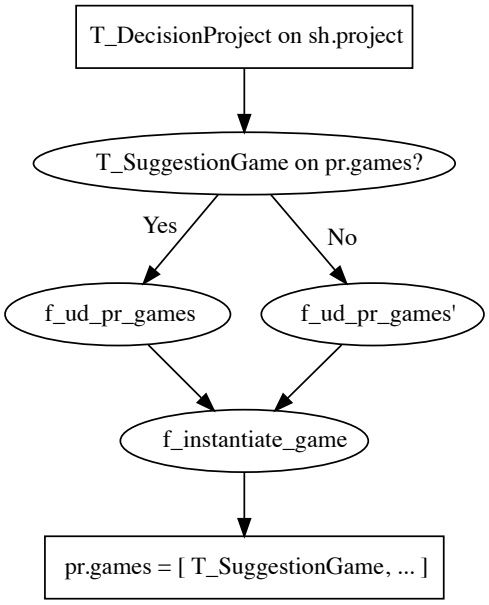

Figure 2: Update of private games

4.2.1 Update private games

We want the first rule $f_{ud\_pr\_games}$ to apply in a context where an agent has a project on her gameboard,

[4] One could argue that a suggestion might be followed by more than one move motivating the suggestion, and it would of course be possible to alter $^{\leqslant 1}$ to $^{\leqslant 2}$ or $^{\leqslant 3}$ or even $^{\leqslant +}$ using the kleene plus to mean one or more (with a corresponding loop in Figure 1) depending on how many motivation moves the model should allow.

but the game first on the list of private games is not relevant to carry out the project. The agent is then licensed to either reraise a game already on private games (but not first on the list) or to load a relevant game from resources onto private games. Now, one question that arises here is what it means to be a relevant game in relation to a particular project. One way of describing this would be in terms of licences in an agent's resources. If an agent has in her resources a link between a type of project T_P and a type of game T_G, she has a licence to carry out a communicative project of type T_P by means of T_G, and may load it onto 'private.games' on her DGB.

Which types of games are relevant to carry out particular types of projects is an empirical question. We think of the update rules licensing the carrying out of a project by means of a particular type of game as reflecting the pragmatic norms of a community. One way of modelling how an agent selects a strategy – for example choosing between an indirect and a direct speech act – would be to extend the model with a probabilistic component (see for example Eshghi and Lemon, 2014). However, in the limited model we are focusing on here, we assume that we have access to only one type of game which is relevant to the project at hand. Moreover, it seems to us that a limited set of project types and game types would suffice to account for a large number of dialogue situations. Thus, for each project type we would introduce a set of postulates defining which games could be relevant to carry out a project of that type. We use the notation relevant_to(T_1, T_2) to represent relevance of T_1 in relation to T_2.

When a communicative project appears on an agent's DGB and the agent initiates carrying out the project there are, as mentioned above, two possibilities. Either there is a game present in the private games field of the DGB by means of which the project can be carried out, or there is not. In the first case we want to make sure that the appropriate game is moved up to the first slot on the list of private games. In the second case, we want to pick an appropriate game from the agent's long term memory, and place it first on the list of private games. The update of 'private.games' thus consists of three rules: $f_{ud\_pr\_games}$ for reraising a game, $f_{ud\_pr\_games'}$ for uploading a game from resources, and – to complete the update – f_{inst_game}. In an instantiated game the roles (player1, player2, etc.) are assigned to dialogue participants from the point of view of the participant on whose gameboard the instantiated game appears. This means in the case of the suggestion game, that when A starts carrying out a decision making project by initiating a game of type $T_{SuggestionGame}$, she has also taken on the role of 'player 1' in that game. In every move type of the instantiated game on her DGB the move to be carried out by player 1 will be assigned to SELF, the ones by player 2 to OTHER.

Even though instantiated games involve assignments of roles to dialogue participants, we still want to be able to treat them as types. For this reason, the type of games is a *join type*. A join type is a disjunction such that, for any two types T_1 and T_2 you can form the join $T_1 \vee T_2$. a : $T_1 \vee T_2$ just in case either a : T_1 or a : T_2. This means that the type of games, T_{Game}, in our theory is a join of the types non-instantiated game, $T_{NonInstGame}$ and instantiated game, $T_{InstGame}$ as defined in (8):

(8) $a : T_{Game}$ iff $a : T_{NonInstGame}$ or $a : T_{InstGame}$

By defining the type of game as a join, we make sure that we can handle situations where, for example, something sudden and unexpected happens, and dialogue participant needs to postpone the initiation of a game already on the DGB. We will look at the instantiation process in more detail further on in this section.

(9) $f_{ud\_pr\_games}$ =

$$\lambda r{:}\begin{bmatrix}\text{pr} : \begin{bmatrix}\text{games} : \text{list}(T_{Game})\end{bmatrix} \\ \text{sh} : \begin{bmatrix}\text{project} = [\ T_{DecisionProject}\ ] : \text{list}(\textit{RecType})\end{bmatrix}\end{bmatrix} .$$
$$\lambda e{:}\begin{bmatrix}\text{g} : T_{SuggestionGame} \\ \text{c}_1 : \text{in(g, } r.\text{pr.games)}\end{bmatrix} .$$
$$\begin{bmatrix}\text{pr} : \begin{bmatrix}\text{games} = [\mu(e.\text{g}, r.\text{pr.games})\ ] : \text{list}(T_{Game})\end{bmatrix}\end{bmatrix}$$

In (9), $f_{ud\_pr\_games}$ takes a situation of the type where there is a decision project on 'shared.project' and, if there is a game of type $T_{SuggestionGame}$ on private games in that record, the function returns a type of situation where that game type is first on 'private.games'.

We think of the update rule $f_{ud\_pr\_games'}$, as seen in (10) as a function from an information state where an agent has a decision project on her gameboard but no game of type $T_{SuggestionGame}$ on the list of games on 'private.games'[5], to an information state where the agent has a decision project on 'shared.project' and a suggestion game first on 'private.games'. In this case the game $T_{SuggestionGame}$

[5]There may be other games on the list of private games, just not the game *suggestion game*.

has to be retrieved from parts of the agent's resources which are external to the DGB.

(10) $f_{ud\_pr\_games'} =$
$\lambda r{:}\begin{bmatrix}\text{pr} : \begin{bmatrix}\text{games} : \text{list}(T_{Game})\end{bmatrix} \\ \text{sh} : \begin{bmatrix}\text{project} = [T_{DecisionProject}\ ] : \text{list}(\textit{RecType})\end{bmatrix}\end{bmatrix}.$
$\lambda e{:}\begin{bmatrix}\text{g} : T_{SuggestionGame} \\ \text{c}_1 : \neg\text{in}(\text{g}, r.\text{pr.games})\end{bmatrix}.$
$\begin{bmatrix}\text{pr} : \begin{bmatrix}\text{games} = [e.\text{g} \mid r.\text{pr.games}] : \text{list}(T_{Game})\end{bmatrix}\end{bmatrix}$

The functions in (9) and (10) are similar to the update functions discussed by Cooper (2023). In order to obtain the required update of such a function we need to apply it to the *current information state* – that is the information state at the start of the update – of the agent whose information state we seek to capture. Let us consider a scenario where agent A has previously considered suggesting Walnut Street, but was distracted by an event which the agent has just observed. This caused another conversational game, T_{G_X}, to appear on the DGB. His initial information state is thus of the type in (11), which we refer to as $T_{current}$.

(11) $T_{current} =$
$\begin{bmatrix}\text{pr} : \begin{bmatrix}\text{games} = [T_{G_X}, T_{SuggestionGame}] : \text{list}(T_{Game}\ )\end{bmatrix} \\ \text{sh} : \begin{bmatrix}\text{project} = [T_{DecisionProject}\ ] : \text{list}(\textit{RecType})\end{bmatrix}\end{bmatrix}$
$s_{current} : T_{current}$

Before we apply the function we need to make sure that the type of the current information state is a subtype of the domain type of $f_{ud\_pr\_games}$. We should point out here that the type of the current information state might very well have other fields such as a shared game, a latest utterance, shared beliefs, etc., and still be a subtype of the domain type of $f_{ud\_pr\_games}$.

In (12) we see the application of $f_{ud\_pr\_games}$ to $s_{current}$, followed by an asymmetric merge of the result of that function application and the type $T_{current}$ of $s_{current}$ (as well as e_1 witnessing the condition that $T_{SuggestionGame}$ is in $s_{current}$.pr.games).

(12) a. $f_{ud\_pr\_games}\ (s_{current})(e_1)=$
$\begin{bmatrix}\text{pr} : \begin{bmatrix}\text{games} = [T_{SuggestionGame}, T_{G_X}] : \text{list}(T_{Game})\end{bmatrix}\end{bmatrix}$

b. $T_{current}\ \boxed{\wedge\!\!\cdot}$
$\begin{bmatrix}\text{pr} : \begin{bmatrix}\text{games} = [T_{SuggestionGame}, T_{G_X}] : \text{list}(T_{Game})\end{bmatrix}\end{bmatrix} =$
$\begin{bmatrix}\text{pr} : \begin{bmatrix}\text{games} = [T_{SuggestionGame}, T_{G_X}] : \text{list}(T_{Game})\end{bmatrix} \\ \text{sh} : \begin{bmatrix}\text{project} = [\begin{bmatrix}\text{e} : \text{decide}(\{A_1, A_2\}, \textit{Issue})\end{bmatrix}] : \text{list}(\textit{RecType})\end{bmatrix}\end{bmatrix}$

4.2.2 Instantiation of game

After an update putting a game which is a subtype of $T_{SuggestionGame}$ first on the list of private games (either by $f_{ud\_pr\_games}$ or $f_{ud\_pr\_games'}$), we need to instantiate the game, that is associate the roles of the game with the players in this particular situation. To do this we apply the function $f_{inst\_T_{SuggestionGame}}$ to a record assigning the values 'SELF' and 'OTHER' to the roles of the suggestion game.

(13) $f_{inst\_T_{SuggestionGame}} =$
$\lambda r{:}\begin{bmatrix}\text{player1} : \textit{Ind} \\ \text{player2} : \textit{Ind}\end{bmatrix}.\ [\text{e} : \text{suggest}(r.\text{player1})]^\frown$
$[\text{e} : \text{motivate}(r.\text{player1})]^{\leqslant 1\frown}\ [\text{e} : \text{respond}(r.\text{player2})]$

For dialogue participant A in our current example this assignment would be that in (14.)

(14) $r = \begin{bmatrix}\text{player1=SELF} \\ \text{player2=OTHER}\end{bmatrix}$

In (15) we see the application of $f_{inst\_T_{SuggestionGame}}$ to r.

(15) $f_{inst\_T_{SuggestionGame}}(r) =$
$\begin{bmatrix}\text{e} : \text{suggest}(\begin{bmatrix}\text{player1=SELF} \\ \text{player2=OTHER}\end{bmatrix}.\text{player1})\end{bmatrix}^\frown$
$\begin{bmatrix}\text{e} : \text{motivate}(\begin{bmatrix}\text{player1=SELF} \\ \text{player2=OTHER}\end{bmatrix}.\text{player1})\end{bmatrix}^{\leqslant 1\frown}$
$\begin{bmatrix}\text{e} : \text{respond}(\begin{bmatrix}\text{player1=SELF} \\ \text{player2=OTHER}\end{bmatrix}.\text{player2})\end{bmatrix} =$
$[\text{e} : \text{suggest(SELF)}]^\frown[\text{e} : \text{motivate(SELF)}]^{\leqslant 1\frown}$
$[\text{e} : \text{respond(OTHER)}]$

The instantiated suggestion game would in this situation thus be $T_{SuggestionGameInst}$, as seen in (16):

(16) $T_{SuggestionGameInst} = [\text{e} : \text{suggest(SELF)}]^\frown$
$[\text{e} : \text{motivate(SELF)}]^{\leqslant 1\frown}[\text{e} : \text{respond(OTHER)}]$

4.3 Updating the agenda

An important aspect of the notion of conversational game is that players (conversational participants), by identifying an utterance as being part of a particular game, get an idea of which moves are likely to follow and what part they should expect to play over the next few turns of the dialogue. In this sense conversational games may be seen as engines driving dialogues forward. Once a game is loaded onto the gameboard and roles are assigned to individuals in the context, an agent involved in a conversation can at any stage of the game look at her gameboard

and know what options are available if she wants to keep playing the game. Before the update of the agenda, agent A – if playing the suggestion game – has on her private games the instantiated game $T_{SuggestionGameInst}$ which we see in (17).

Now, we want an update rule that would load the first available move of the game which is to be carried out by SELF, onto the agenda. We have a set of rules pertaining to the suggestion game that govern the dynamics of the agenda, which is inherent in the suggestion game in (16). The agenda is part of the 'private'-field of an agent's DGB, and is represented as a record type (move type). Each move type has a label 'e' paired with one of a set of speech act types, e.g. *Suggest*.There are a number of constraints on such move types having to do with the roles of the agents involved in dialogue, c_{actor}. Further, there is a label 'ctnt' for content, which – after the first update of the agenda – will not yet be associated with a specified content.

The first rule to be employed of the rules of the suggestion game is a "starting rule" in (17), stating that if a player has an empty agenda and a suggestion game on his private DGB, he may push a suggestion onto the agenda. We refer to this rule as $f_{update\_agenda\_suggestion}$.

(17) $f_{update\_agenda\_suggestion} =$
$\lambda r{:}\left[\text{pr} : \begin{bmatrix}\text{agenda} = [\,] : \text{list}(\textit{RecType}) \\ \text{games} = [T_{SuggestionGameInst}] : \text{list}(T_{Game})\end{bmatrix}\right].$
$\left[\text{pr} : \left[\text{agenda} = [\ \begin{bmatrix}\text{e} : \text{suggest(SELF)} \\ \text{ctnt} : \textit{RecType}\end{bmatrix}\ ] : \text{list}(\textit{RecType})\right]\right]$

The content of the move type that ends up on the agenda is unspecified. $f_{update\_agenda\_suggestion}$ is applied to a record of the type in (18):

(18)
$$\begin{bmatrix}\text{pr} : \begin{bmatrix}\text{agenda} = [\,] : \text{list}(\textit{RecType}) \\ \text{games} = [T_{SuggestionGameInst}] : \text{list}(T_{Game})\end{bmatrix} \\ \text{sh} : \left[\text{project} = [\,[\text{e} : \text{decide}(\{A, B\}, \text{data})]\,] : \text{list}(\textit{RecType})\right]\end{bmatrix}$$

We apply the function in (17) to the current information state of the type in (18), and asymmetrically merge the current state type with the result of function application. In (19) we see the type of A's information state after the rule has been applied.

(19)
$$\begin{bmatrix}\text{pr} : \begin{bmatrix}\text{agenda} = [\ \begin{bmatrix}\text{e} : \text{suggest(SELF)} \\ \text{ctnt} : \textit{RecType}\end{bmatrix}\ ] : \text{list}(\textit{RecType}) \\ \text{games} = [T_{SuggestionGameInst}\,] : \text{list}(T_{Game})\,)\end{bmatrix} \\ \text{sh} : \left[\text{project} = [\,[\text{e} : \text{decide}(\{A, B\}, \text{data})]\,] : \text{list}(\textit{RecType})\right]\end{bmatrix}$$

The next update rule provided by the conversational game (although this rule is actually general and applicable to any conversational game) is a rule saying that if we have an item on the agenda which is to be performed by SELF and whose content is specified, that is the label 'ctnt' has one specific value ([ctnt=T:*RecType*]), then the agent is allowed to make that move and push the next move onto the agenda. However, at the moment the item on the agenda is not specified in terms of content – the label is just typed *RecType* (ctnt:*RecType*). In order to add a content specific move to the agenda, the agent needs to search her resources for relevant facts and ways of reasoning about the situation and the project at hand.

5 Conclusions

In this paper we consider different approaches to categorisation of interaction according to contextual factors, such as activity type and communicative project. We also consider the micro-level of organisations and present a model of how particular interactions play out within social activities and communicative projects of certain types. Our model can be integrated in a general formal model of dialogue such as Ginzburg (2012). We recognise three categories by which to define interactions – (communicative) activity, (communicative) project and conversational game, which we argue are associated with different aspects of the interaction and to some extent interdependent. These categories are also linked in a principled way to particular fields on the DGB. We illustrated by means of two examples of group decision making how the conversational game a project is associated with can be realised in different ways depending on the activity in which it is embedded. We may think of this as the possibilities afforded by the project being modified by the activity. We showed how the process of identifying a conversational game based on a project at hand, taking on or identifying one's role in the game and carrying out the appropriate moves can be modelled using DGBs modelled in TTR.

Our approach has the advantage that roles in a conversational game are analysed independently from the participants that carry them out in a particular interaction. This enables us to model for example anticipation of contributions by others and co-created utterances.

Acknowledgments

The research in this paper was supported by a grant from the Swedish Research Council (VR project 2014-39) for the establishment of the Cen-

tre for Linguistic Theory and Studies in Probability (CLASP) at the University of Gothenburg and ERC Starting Grant DivCon: Divergence and Convergence in Dialogue: The Dynamic Management of Mismatches (101077927)

References

Jens Allwood. 2000. An activity based approach to pragmatics. In Harry C. Bunt & William Black, editor, *Abduction, belief, and context in dialogue: studies in computational pragmatics*, pages 47–78. John Benjamins, Amsterdam.

Kathryn A Baker. 2007. Organizational communication. *Management Benchmark Study*, 1(1):1–3.

Ellen Breitholtz. 2020. *Enthymemes and Topoi in Dialogue: The Use of Common Sense Reasoning in Conversation*. Brill, Leiden, The Netherlands.

Andrew D Brown and Ken Starkey. 1994. The effect of organizational culture on communication and information. *Journal of Management studies*, 31(6):807–828.

Ronnie Cann, Ruth Kempson, and Lutz Marten. 2005. *The Dynamics of Language*. Elsevier, Oxford.

Lauri Henrik Carlson. 1982. *Dialogue games: An approach to discourse analysis*. Ph.D. thesis, Massachusetts Institute of Technology.

Herbert H Clark, Susan E Brennan, et al. 1991. Grounding in communication. In *Perspectives on Socially Shared Cognition*, pages 127–149. American Psychological Association, Washington DC.

Robin Cooper. 2005. Records and record types in semantic theory. *Journal of Logic and Computation*, 15(2):99–112.

Robin Cooper. 2014. How to do things with types. In *Joint proceedings of the second workshop on Natural Language and Computer Science (NLCS 2014) & 1st international workshop on Natural Language Services for Reasoners (NLSR 2014) July*, pages 149–158.

Robin Cooper. 2023. *Type theory and language - From perception to linguistic communication*. Oxford University Press, Oxford, UK.

Robin Cooper and Jonathan Ginzburg. 2015. Type theory with records for natural language semantics. In Shalom Lappin and Chris Fox, editors, *The Handbook of Contemporary Semantic Theory*, pages 375–407. Wiley Blackwell, Oxford.

Arash Eshghi and Oliver Lemon. 2014. How domain-general can we be? Learning incremental dialogue systems without dialogue acts. In *Proceedings of the 18th Workshop on the Semantics and Pragmatics of Dialogue - Full Papers*, pages 53–61. SEMDIAL.

Jonathan Ginzburg. 1994. An update semantics for dialogue. In *Proceedings of the Tilburg International Workshop on Computational Semantics*. ITK Tilburg.

Jonathan Ginzburg. 1996. Dynamics and the semantics of dialogue. In *Logic, language and computation*, volume 1. CSLI publications, Stanford, CA.

Jonathan Ginzburg. 2012. *The interactive stance: Meaning for conversation*. Oxford University Press, Oxford.

Jonathan Ginzburg. 2016. The semantics of dialogue.

Christine Howes and Arash Eshghi. 2021. Feedback relevance spaces: Interactional constraints on processing contexts in Dynamic Syntax. *Journal of Logic, Language and Information*, 30(2):331–362.

Ruth Kempson, Wilfried Meyer-Viol, and Dov Gabbay. 2001. *Dynamic Syntax: The Flow of Language Understanding*. Blackwell, Oxford.

Staffan Larsson. 2002. *Issue Based Dialogue Management*. Ph.D. thesis, University of Gothenburg.

Staffan Larsson and David Traum. 2000. Information state and dialogue management in trindi dialogue move engine tool kit. *Natural Language Engineering*, 6:323–240.

James A. Levin and James A. Moore. 1977. Dialogue-games: Metacommunication structures for natural language interaction. *Cognitive Science*, 1(4):395–420.

David Lewis. 1979. Scorekeeping in a language game. *Journal of Philosophical Logic*, 8(1):339–359.

Per Linell. 2009. *Rethinking Language, Mind, and World Dialogically: Interactional and Contextual Theories of Human Sense-Making*. Advances in Cultural Psychology: Constructing Human Development. Information Age Publishing.

Andy Lücking. 2016. Modeling co-verbal gesture perception in type theory with records. In *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 383–392. IEEE.

Norman Malcolm. 1954. Wittgenstein's philosophical investigations. *The Philosophical Review*, 63(4):530–559.

Richard Montague. 1973. The proper treatment of quantification in ordinary english. In Jaakko Hintikka, Julius Moravcsik, and Patrik Suppes, editors, *Approaches to Natural Language*. D. Reidel, Dordrecht.

Robert Stalnaker. 1978. Assertion. In P. Cole, editor, *Pragmatics*, volume 9 of *Syntax and Semantics*. New York Academic Press, New York, NY.

Ludwig Wittgenstein. 1953. *Philosophical investigations. Philosophische Untersuchungen.* Macmillan, London. Translated by G. E. M. Anscombe.

JoAnne Yates and Wanda J Orlikowski. 1992. Genres of organizational communication: A structurational approach to studying communication and media. *Academy of management review*, 17(2):299–326.