Formal Analysis of Meeting Protocols

Catholijn Jonker, Martijn Schut, Jan Treur, and Pınar Yolum

Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081a, 1081HV Amsterdam, The Netherlands
{jonker, schut, treur, pyolum}@few.vu.nl

Abstract. Organizations depend on regular meetings to carry out their everyday tasks. When carried out successfully, meetings offer a common medium for participants to exchange ideas and make decisions. However, many meetings suffer from unfocused discussions or irrelevant dialogues. Within Social Science sometimes general, informal meeting guidelines are formulated. To study meetings in detail, we first formalize general properties for meetings and a generic meeting protocol for the role interactions in meetings that is coherent with such guidelines. In the context of a case study, an example meeting is simulated based in this protocol. The properties are formally verified in this trace. These properties are also verified formally against empirical data of a real meeting in the same context. A comparison of the two traces reveals that a real meeting is more robust in the sense that by exception violations of the protocol may occur, and these exceptions are handled effectively without damaging the successfulness of the meeting. Given this observation, a more refined protocol is specified that includes exception handling strategies. Based on this refined protocol a meeting is simulated that closely resembles the real meeting.

1 Introduction

Meetings are an integral part of every day life. Meetings are important tools in most organizations to structure decision processes and to disseminate information throughout the organization. Typically the members of a group come together on a regular basis to inform each other of new developments, to discuss problems, and propose solutions. While many organizations depend on face-to-face meetings, it is notoriously difficult to hold a focused and effective meeting. There is an abundant literature on guidelines on how to carry a successful meeting [Wolf 2002, Creighton 2002]. These guidelines are rather informal, which makes it hard to put into practice and hard to evaluate.

This paper formalizes a domain-independent meeting protocol that can be used in various meetings. The formalization captures many intuitive ideas that are also mentioned in meeting guidelines, hence is compatible with most meeting guidelines. The formalization captures actions that need to be carried out by participants as well as constraints that each participant has to satisfy. The main aim of this work is to understand how meeting protocols are carried out, by understanding the different flows that take place in meetings. To achieve this, we study the meeting protocol with an empirical trace as well as with a simulated trace and analyze various properties. The empirical trace is based on observations of a real meeting. The simulated trace is generated in a simulation environment where agents are assumed to follow the meeting protocol strictly. We compare the two traces in terms of desirable properties.

The rest of this paper is organized as follows. Section 2 gives a technical background on the AGR methodology. Section 3 develops the formal generic meeting protocol using AGR. Section 4 introduces the empirical trace as well as the simulated trace. Chapter 4 analyzes both meeting in terms of desired properties. Chapter 5 discusses the relevant literature in comparison to this work.

2 Technical Background

To formally specify dynamic properties characterising dynamics of meetings, an expressive language is needed. To this end the Temporal Trace Language TTL is used (cf. [Jonker and Treur, 2002]), this language is briefly defined as follows. Examples of properties expressed in TTL can be found in later sections.

A *state ontology* is a specification (in order-sorted logic) of a vocabulary, i.e., a signature. A state for ontology Ont is an assignment of truth-values {true, false} to the set At(Ont) of ground atoms expressed in terms of Ont. The *set of all possible states* for state ontology Ont is denoted by

STATES(Ont). The set of *state properties* STATPROP(Ont) for state ontology Ont is the set of all propositions over ground atoms from At(Ont). A fixed *time frame* T is assumed which is linearly ordered. A *trace* or *trajectory* γ over a state ontology Ont and time frame T is a mapping $\gamma : T \rightarrow$ STATES(Ont), i.e., a sequence of states γ_t ($t \in T$) in STATES(Ont). The set of all traces over state ontology Ont is denoted by TRACES(Ont). Depending on the application, the time frame T may be dense (e.g., the real numbers), or discrete (e.g., the set of integers or natural numbers or a finite initial segment of the natural numbers), or any other form, as long as it has a linear ordering. The set of *dynamic properties* DYNPROPEXP(Σ) is the set of temporal statements that can be formulated with respect to traces based on the state ontology Ont in the following manner (for an organization or part thereof, Ont is the union of all input, output and internal state ontologies of the roles in the organization (part).

Given a trace γ over state ontology Ont, the input state of a role at time point t is denoted by state(γ, t, input(r)); analogously, state(γ, t, output(r)), and state(γ, t, internal(r)) denote the output state and internal state of the role. These states can be related to state properties via the formally defined satisfaction relation |=, comparable to the Holds-predicate in the Situation Calculus: state(γ, t, output(r)) |= p denotes that state property p holds in trace γ at time t in the output state of the organism. Based on these statements, dynamic properties can be formulated in a formal manner in a sorted first-order predicate logic with sorts T for time points, Trace for traces and F for state formulae, using quantifiers over time and the usual first-order logical connectives such as $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\forall$, $\exists$. Within TTL abstractions can be made by introducing additional terms (e.g., predicates) which are definable in terms of the existing terms.

3 Meetings Formalized

In this section a formalisation of the organisation of a meeting is presented: organisational structure, dynamic properties for the overall process, and a protocol for role interactions.

3.1 Organizational Structure

The American Heritage® Dictionary of the English Language provides the following definitions of a meeting: The act or process or an instance of coming together; an encounter. An assembly or gathering of people, as for a business, social, or religious purpose. For the study of organizations only the second is relevant. A meeting is a gathering of people with a common purpose. The common purpose, especially in a business setting, entails that the meeting is convened to discuss issues related to that common purpose, and often with the aim to make some decisions pertinent to the common purpose.

Experience learns that a meeting without some way to structure the discussions takes much longer to reach a conclusion than a meeting in which the discussion is structured. In fact, in any group of more than 2 people, discussion might lead to different sub-groups discussing the same or different arguments. How can you be sure that everyone hears all the arguments made? How can you be sure that everyone who has something relevant to add to the discussion gets a chance to present his views? How can it be achieved that all people involved (also those who were absent) have access to the results of the meeting afterwards?

Therefore, a common form to structure meetings is the following. A Chairperson chairs every meeting. The Secretary takes minutes of the meeting. Given the chosen meaning of the concept meeting, the dictionary says the following about the Chairperson and Secretary roles:

- Chairperson
  The presiding officer of an assembly, meeting, committee, or board.
- Secretary
  An officer who keeps records, takes minutes of the meetings, and answers correspondence, as for a company.

Depending on the type of meeting, also a Treasurer role may be distinguished; for simplicity this role is not considered in this paper. Taking minutes means writing down the arguments presented by the Participants of the meeting, as well as the decisions made. Chairing a meeting means opening and closing a meeting, making sure that people are talking one at a time, and that only the current issue is discussed. The decision process differs according to the customs and/or agreements in the group. Common decision procedures are decision by consensus, decision by majority, and decision by the Chairperson. In graphical form, the general structure of a meeting is as depicted in Figure 1. A question to be addressed is how dynamic properties describing such a protocol can be identified.

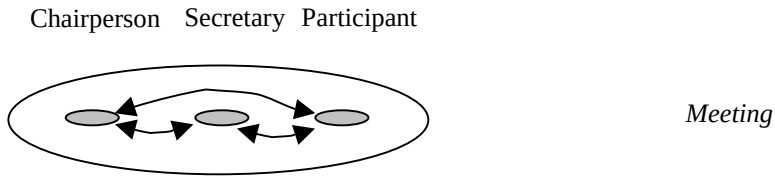


Fig. 1. Generic meeting organization structure

3.2 Organizational Behavior

Dynamic properties characterizing an organizational behavior can be specified at different levels: at the level of the organization as a whole, at the level of interactions between roles (interaction protocol), and at the level of roles.

3.2.1 Overall Organizational Behavior Properties

At the level of the overall organization (which in this case is the meeting as a whole) a number of organization properties can be identified. As an example the following property expresses that no two particpants speak at the same time. In this and the following properties, communicates_from_to(p, q, x, y) denotes that p communicates to q the communicative act x with the content y. For this paper, we consider two types of communicative acts, mainly inform and declare. Only when the communicative act x is a declare, then the receiver q is droped meaning that the message is sent to everyone. For the sake of simplicity, we assume that messages always reach their destination.

OP1
Informal
During the meeting only one Participant is speaking at a time.
Semiformal
At any point in time,
if any participant is speaking,
then all other participants are not speaking
Formal
$\forall t, p, p' :PARTICIPANT, q, q' :ROLE, x, x', y, y'$
$p \neq p' \ \& \ state(\gamma, t, output(p)) \models communicates\_from\_to(p, q, x, y) \Rightarrow$
$state(\gamma, t, output(p')) \not\models communicates\_from\_to(p', q', x', y')$

For an overview of all organisation properties specified, see Appendix A. To express the properties the following abstractions have been introduced for agenda item, current agenda item and addressed agenda item.

Abstraction: agenda item
Informal
An *agenda item* is an item that was declared to be an ageda item and not retracted since then
Semiformal
Item i is an agenda item if at some point in time it was declared to be so,
and since then it was not declared that it is no agenda item
Formal
$agenda\_item\_at(\gamma, i, t) =$
$\exists t' \leq t$
$state(\gamma, t', output(m)) \models communicates\_from\_to(m, declare, agenda\_item(i)) \ \&$
$\forall t'' \ \ t' < t'' < t \ state(\gamma, t'', output(m)) \not\models communicates\_from\_to(m, declare, not\_agenda\_item(i))$

Abstraction: current agenda item
Informal
A *current agenda item* is one that was opened but not yet closed.
Semiformal
An agenda item is a current item if and only if
Some time ago the Chairperson declared that item to be the current item

And since then the Chairperson did not declare the item closed.

Formal

current_agenda_item_at(γ, i, t) =
∃∀m:CHAIR, t' ≤ t
state(γ, t', output(m)) |= communicates_from_to(m, declare, opened(i)) &
∀t" [t' < t"< t ⇒ state(γ, t", output(m)) |≠ communicates_from_to(m, declare, closed(i))]

Abstraction: addressed agenda item

Informal

An agenda item has been *addressed* if it was opened and closed during the meeting.

Semiformal

An agenda item has been addressed if and only if
for every time point that the chairperson has opened the item, at a later time point she declared the item closed

Formal

addressed_agenda_item_at(γ, i, t) =
∃t1≤t state(γ, t1, output(m)) |= communicates_from_to(m, declare, opened(i)) &
∀t2≤t state(γ, t2, output(m)) |= communicates_from_to(m, declare, opened(i)) ⇒ ∃t3 t2≤t3≤t &
state(γ, t3, output(m)) |= communicates_from_to(m, declare, closed(i))

3.2.2 Role Interaction Properties: the Generic Meeting Protocol

A number of role interaction properties have been specified to define a generic interaction protocol for a meeting. For an overview, see Appendix B. Two examples are the following.

RI1 If the Chairperson generates a question (which implies a permission to speak) to a Participant, then a little time later the Participant generates an answer.

Formal

∀m:CHAIR, p:PARTICIPANT ∀t
[state(γ, t, output(m)) |= communicates_from_to(m, p, request, q)) &
not ∃ x state(γ, t', output(p)) |= communicates_from_to(p, m, inform, x)]
⇒ ∃t' > t state(γ, t', output(p)) |= communicates_from_to(p, m, inform, answer_on(a, q)))

RI2 If a Participant requests to add an item to the agenda,
then the Chairperson communicates this to all Participants.

Formal

∀m:CHAIR, p:PARTICIPANT ∀t
state(γ, t, output(p)) |= communicates_from_to(p, m, request, agenda_item(i)))
⇒ ∃t' > t state(γ, t', output(m)) |= communicates_from_to(m, declare, agenda_item(i)))

Notice that it is not difficult to express in these properties within how many seconds a reaction should be given. For simplicity this has been left out.

4 Example Trace of a Simulated Meeting based on the Generic Meeting Protocol

The simulations of interest are generated using a logic-based simulation environment. Using this environment, executable temporal rules are specified so that the simulation environment can generate a trace. These executable temporal rules consist of rules that fire based on the current status of the world, without regard to the past.This trace describes which properties related to the protocol hold at each time point. The generated traces can then be analyzed with an automated logic-based checker. This checker takes as input a property of interest about the trace and logically validates the property by the trace. If the property holds in the trace, the checker outputs success otherwise it outputs fail.

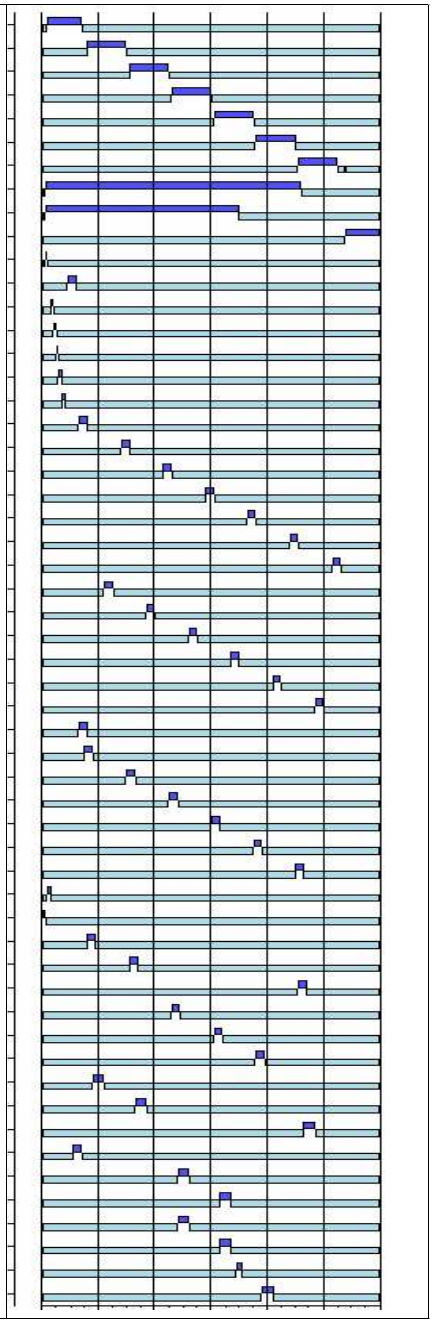
We consider a simulation of an example meeting on the topic of study groups. These simulations consist of one chairperson (referred to as chair) and three participants (referred to as p1, p2, and p3). The agenda items are about particular study groups, hence named as group_1, group_2, and so on. For each of the agenda items one of the participants is the contact person, who is asked to speak if the agenda item is opened.

The simulation discussed here is based on the formal specification of the generic meeting protocol, which was developed based on the meeting guidelines discussed above. The simulation follows the protocol but here we give a brief overview of the trace that is shown in Table 1. The left side of the table

lists the predicates used. The right side of the table shows the timeline. A predicate is true over a period of time if there is a bar above the time line; otherwise the predicate is false. The simulation starts by the chairperson declaring the desired end time (`proposed_end_time`) for the meeting. Next, the chairperson announces the agenda items one by one (`agenda_item`). Next, the chairperson asks for further additions to the agenda. Participant p1 suggests a new item (`schedule`), which is also added to the agenda. Once the agenda is finalized, the chair opens the first item (`group_1`) for discussions. The chairperson requests information from the participant who is likely to have input on the current agenda item. After this participant is done speaking, the chairperson asks the other participants to see if they have further information for the topic (`last_comments`). Since no participant has further input on the agenda item (`group_1`), the chairperson closes the agenda item and opens the second item. This procedure repeats itself until the agenda item is group_4. On this agenda item, when the chairperson asks for other comments from the participants, participant p3 provides additional comments. Later the meeting is continued as before. After the last agenda item is discussed, the chairperson declares the meeting closed.

Table 1 Simulation trace of a meeting that follows the generic protocol

internal(chair)|agenda_being_discussed
internal(chair)|being_discussed(group_1)
internal(chair)|being_discussed(group_2)
internal(chair)|being_discussed(group_3)
internal(chair)|being_discussed(group_4)
internal(chair)|being_discussed(group_5)
internal(chair)|being_discussed(schedule)
internal(p1)|wants_to_speak_about(p1, schedule)
internal(p3)|has_input_for(p3, group_4)
output(chair)|communicates_from_to(chair, declare, meeting_closed)
output(chair)|communicates_from_to(chair, declare, meeting_opened)
output(chair)|communicates_from_to(chair, declare, other_items)
output(chair)|communicates_from_to(chair, declare, agenda_item(group_1))
output(chair)|communicates_from_to(chair, declare, agenda_item(group_2))
output(chair)|communicates_from_to(chair, declare, agenda_item(group_3))
output(chair)|communicates_from_to(chair, declare, agenda_item(group_4))
output(chair)|communicates_from_to(chair, declare, agenda_item(group_5))
output(chair)|communicates_from_to(chair, declare, agenda_item(schedule))
output(chair)|communicates_from_to(chair, declare, closed(group_1))
output(chair)|communicates_from_to(chair, declare, closed(group_2))
output(chair)|communicates_from_to(chair, declare, closed(group_3))
output(chair)|communicates_from_to(chair, declare, closed(group_4))
output(chair)|communicates_from_to(chair, declare, closed(group_5))
output(chair)|communicates_from_to(chair, declare, closed(schedule))
output(chair)|communicates_from_to(chair, declare, last_comments(group_1))
output(chair)|communicates_from_to(chair, declare, last_comments(group_2))
output(chair)|communicates_from_to(chair, declare, last_comments(group_3))
output(chair)|communicates_from_to(chair, declare, last_comments(group_4))
output(chair)|communicates_from_to(chair, declare, last_comments(group_5))
output(chair)|communicates_from_to(chair, declare, last_comments(schedule))
output(chair)|communicates_from_to(chair, declare, newitem(schedule))
output(chair)|communicates_from_to(chair, declare, opened(group_1))
output(chair)|communicates_from_to(chair, declare, opened(group_2))
output(chair)|communicates_from_to(chair, declare, opened(group_3))
output(chair)|communicates_from_to(chair, declare, opened(group_4))
output(chair)|communicates_from_to(chair, declare, opened(group_5))
output(chair)|communicates_from_to(chair, declare, opened(schedule))
output(chair)|communicates_from_to(chair, declare, planned_end_time(120))
output(chair)|communicates_from_to(chair, declare, planned_start_time(1))
output(chair)|communicates_from_to(chair, p1, request, group_1)
output(chair)|communicates_from_to(chair, p1, request, group_2)
output(chair)|communicates_from_to(chair, p1, request, schedule)
output(chair)|communicates_from_to(chair, p2, request, group_3)
output(chair)|communicates_from_to(chair, p2, request, group_4)
output(chair)|communicates_from_to(chair, p3, request, group_5)
output(p1)|communicates_from_to(p1, chair, inform, group_1)
output(p1)|communicates_from_to(p1, chair, inform, group_2)
output(p1)|communicates_from_to(p1, chair, inform, schedule)
output(p1)|communicates_from_to(p1, chair, request, schedule)
output(p2)|communicates_from_to(p2, chair, inform, group_3)
output(p2)|communicates_from_to(p2, chair, inform, group_4)
output(p2)|communicates_from_to(p2, chair, stammer, group_3)
output(p2)|communicates_from_to(p2, chair, stammer, group_4)
output(p3)|communicates_from_to(p3, chair, inform, group_4)
output(p3)|communicates_from_to(p3, chair, inform, group_5)



	Time	0	20	40	60	80	100	120

From a broad overview, the simulation described above has some differences from our observations of real meetings. For this reason, we observed a real meeting and obtained data on how it was carried out. These data were analyzed in some depth.

5 An Empirical Trace of a Real Meeting

An important part of the work presented here is based on empirical data. This data was obtained through carefully observing a meeting in the Artificial Intelligence Department of the Vrije Universiteit Amsterdam. Similar to the observation techniques explained elsewhere [Serman and Basili, 1998], the observer sat apart from the meeting participants and the chair. Two of the participants and the chair knew why the observant was present, while a third participant did not.

	Informal Description	**Formal State**
0	C: Let's begin in a minute	proposed_begin_time(1)
1	C: Let's try to finish by 5'o clock	communicates_from_to(chair, declare, meeting_opened) proposed_end_time(120)
2	C: We will talk about the regular agenda	communicates_from_to(chair, declare, agenda_item(group_1) communicates_from_to(chair, declare, agenda_item(group_2) communicates_from_to(chair, declare, agenda_item(group_3) communicates_from_to(chair, declare, agenda_item(group_4) communicates_from_to(chair, declare, agenda_item(group_5)
3	C: First, talk about group_1	communicates_from_to(chair, declare, open(group_1))
4	C: Mark, any inputs for group_1	communicates_from_to(chair, p1, request, group_1)
5	Mark gives an explanation on group_1	communicates_from_to(p1, chair, inform, group_1)
6	C: OK, fine.	communicates_from_to(chair, declare, close(group_1))
7	C: Group_2	communicates_from_to(chair, declare, open(group_2))
8	C: Mark, any inputs for group_2	communicates_from_to(chair, p1, request, group_2)
9	Mark gives an explanation on group_2	communicates_from_to(p1, chair, inform, group_2)
10	Mark complains about lecture notes	communicates_from_to(p1, chair, inform, notes)
11	C: This is not the right time for that.	communicates_from_to(chair, p1, revoke, notes)
12	C: Let's move on	communicates_from_to(chair, declare, close(group_2))
13	C: Group_3	communicates_from_to(chair, declare, open(group_3))
14	C: Lieke, any inputs for group_3	communicates_from_to(chair, p2, request, group_3)
15	Lieke explains with frequent stops.	communicates_from_to(p2, chair, inform, group_3) communicates_from_to(p2, chair, stammer, group_3) communicates_from_to(p3, p2, complete, group_3)
16	C: Let's move on	communicates_from_to(chair, declare, close(group_3))
17	C: Group_4	communicates_from_to(chair, declare, open(group_4))
18	C: Lieke, any inputs for group_4	communicates_from_to(chair, p2, request, group_4)
19	Lieke explains with frequent stops.	communicates_from_to(p2, chair, inform, group_4) communicates_from_to(p2, chair, stammer, group_4) communicates_from_to(p3, p2, complete, group_4)
20	C: Let's move on	communicates_from_to(chair, declare, close(group_4))
21	C: Group_5	communicates_from_to(chair, declare, open(group_5))
22	C: Tibor, any inputs for group_5	communicates_from_to(chair, p2, request, group_5)
23	Tibor speaks more on group_4	communicates_from_to(p3, chair, inform, group_4) communicates_from_to(chair, declare, open(group_4))
24	C: We talked enough on group_4	communicates_from_to(chair, p3, revoke, group_4) communicates_from_to(chair, declare, close(group_4))
25	C: Group_5	communicates_from_to(chair, declare, open(group_5))
26	C: Tibor, any inputs for group_5	communicates_from_to(chair, p2, request, group_5)
27	Tibor speaks on group_5	communicates_from_to(p3, chair, inform, group_5)
28	C: We are done with the agenda	communicates_from_to(chair, declare, close(group_5))
29	M: I have a comment on schedule	communicates_from_to(p1, chair, request, schedule)

30	C: OK	communicates_from_to(chair, declare, agenda_item(schedule)) communicates_from_to(chair, p1, request, schedule)
31	Mark makes a comment	communicates_from_to(p1, chair, inform, schedule)
32	C: OK, we are done now.	communicates_from_to(chair, declare, close(schedule))
33	C: Same time, next week	communicates_from_to(chair, declare, meeting_closed)

Table 2 The transition from informal statements to formal states

The observer wrote down the conversations of the meeting in an informal language. Later these informal texts were formalized to analyze and reason about the meeting. Table 2 gives an overview of these conversations. The left column in the table provides the informal text and the right column gives the formalized states.

We briefly explain the differences from the simulated meeting trace in Section 5. The trace again starts with the chairperson announcing a desired end time for the meeting (`proposed_end_time`). The chairperson announces the agenda items but does not explicitly ask for additions to the agenda. After the chair opens an agenda item and receives input on the item, she closes the item when she sees fit. Compared to the generic meeting protocol described in Section 3.3.2, the difference here is that the chair does not explicitly ask for further input from the participants. Complementing this is a change in the role behavior of participants. Whereas in the meeting simulated according to the generic protocol (Section 5), a participant speaks only when permission is given, in the real meeting participants take the initiative to speak up without being asked. The interesting question then is how these different behaviors affect the outcome of the meetings? Do the desired properties of interest hold for both cases? Does one trace have advantages over the other one? We discuss these questions next.

6 Formal Analysis of Simulated Trace and Emipirical Trace

We analyze the traces generated by these simulations in terms of the organization properties defined above. To do so, the first eight organization properties have been entered into the checker and automatically checked against each trace.

6.1 Analysis of the Simulated Meeting

The meeting simulated according to the generic protocol (Section 5) satisfies the first organization property (OP1) which states that no two participants speak at the same time. This is intuitive since participants speak only when given permission. In this simulation, the chair ensures that only one participant has the permission to speak. Hence, the property holds. The second property (OP2) is on the agenda items that were talked. The role interaction RI8 specifies that once an agenda item is closed, then the chair chooses a new item from the agenda. Hence, it is always the case that the chairperson will open an existing agenda item. This explains why OP2 holds for this trace as well.

OP3 is satisfied for this trace because before closing each topic the chairperson asks for further comments from the participants. Hence, anyone who declares an intent to speak then will get a change to speak. Organization property OP4 states that the meeting is eventually closed. This will always hold for a meeting based on the generic meeting protocol as long as the number of items on the agenda as well as the duration of comments on the items are finite. OP5 ensures that no meeting ends prematurely; that is if the meeting ends, then all agenda items have been discussed. In the specification of the meeting, the only way to close a meeting is when the meeting items have been discussed. OP6 states that no two items are open at the same time. This holds for this trace again due to role interaction RI8. A chairperson will open a new agenda item only if the previous item is closed. Organization property OP7 states that if a participant is speaking then she is speaking on the current item. This follows from the fact that the chairperson will only allow a participant to speak on the current item (RI3). Organization property (OP8) states that meeting start and end on time. This property holds for this trace since the first thing in the traces there is a declaration of intended start and end times of the meeting and that the meeting takes place between these time points. However, in general this property may have conflicts with OP3.

6.2 Analysis of the Empirical Data of the Real Meeting

While the generic meeting protocol obediently obeys the organization properties, the real meeting trace violates some of them. To avoid repetition, only the properties that are violated are discussed here.

The first interesting situation happens during the discussion of item group_3 (see lines 13-16). The chairperson requests information from p2 on the item. The participant p2 speaks with short breaks (`stammer`), which influences one of the other participants (p3) to help p2 with his speech (`complete`). Notice that this is not part of the generic protocol and in general no participant has to help other participants. To be able to generate this behavior, we added an extra role interaction property to the simulation so that participant p3 would help p2. Participant p3's helping p2 is constructive in that it allows p2 to formulate his thoughts. Ironically, this situation disobeys one of the desired organization properties of meetings; namely OP1 which states that no two participants at a meeting should speak at the same time.

After a chairperson requests information from a participant, the participant provides the required information. In some cases, it could also be the case that the participant provides information that is not relevant to the request of the chairperson. One such example happens during the discussion of item group_2 (see lines 7-12). After giving feedback on group_2, participant p1 starts speaking on a topic (`notes`) that is out of the scope of group_2. This is an example of impromptu interruption from participants that sometimes happen. This behavior of p1 causes the violation of the organization property OP7, which says that participant speak on current agenda items only. While this behavior of the participant is not part of the generic interaction protocol, a method for recovering from such a situation is followed in the meeting. Hence, the chair person can first revoke the permission from participant p1 and then continue with the protocol.

Contrary to the generic protocol, in this simulation the chairperson does not request further input from other participants before closing an agenda item. One interesting consequence is that after the discussion of item group_4, the chairperson closes the agenda item (line 20). However, there is still a participant who is willing to speak more on the item. Hence, this participant (participant p3) continues speaking about group_4, even though the item has been closed and a new item has been open (line 23). This point in time is interesting because in reality both agenda items are current. Item group_5 is current because it has been declared as open and not closed by the chairperson. While group_4 is also current, since one participant is talking about this item. Hence, another organization property, property OP6 is violated since there are two current items at the same time. However, this failing of this property does not halt the system. The meeting handles this exception in the sense that the chairperson in this case lets the participant finish and then re-closes the item group_4 and reopens the item group_5 (in lines 24 and 25).

7 Refined Protocol and Simulation

As shown in the analysis in Section 7, a real meeting (such as the one described in Section 6) may deviate from a meeting correctly following the protocol (such as the simulated meeting in Section 5) in the following ways:

- sometimes, by exception, protocol properties are violated by one of the members
- strategies are employed to handle these exceptions and get the meeting on the right track again

One of the reasons that these exceptions occur are the fact that human agents are not ideal and may forget things. In practice members are able to accept these shortcomings and to recover from them. To this end a number of exception handling strategies are used. This can be considered a more sophisticated way of working than just by following the protocol. An interesting question is whether the generic meeting protocol can be refined by including such exception handling strategies to provide a more robust protocol. This question is discussed in the current section.

To experiment with a refined protocol, using the formal states given for the empirical trace, a second simulation was developed, where a number of the rules for the simulation (as used in Section 5) were adapted to reconstruct the empirical trace as precisely as possible. The generated trace indeed closely resembles our observations of the real meeting described in Section 6. For example, the exception of the participant speaking on notes while the current agenda item is group_2, is now handled realistically in the simulation: the chairperson first revokes the permission from participant p1 and then continues with the protocol. Moreover, now also the simulated meeting can handle the exception that during an item i1 a participant wants to add to an already closed agenda item i2. The strategy was added that for such an exception the chairperson returns to the earlier agenda item i2, lets the participant finish and then re-closes the item i2 and reopens the item i1. The following rules, that can be considered part of such a refined protocol, were used to obtain this:

RI1 If after a new agenda item was opened and not yet closed, a Participant speaks on an earlier addressed agenda item,

then the Chairperson closes the current agenda item and reopens the earlier item.

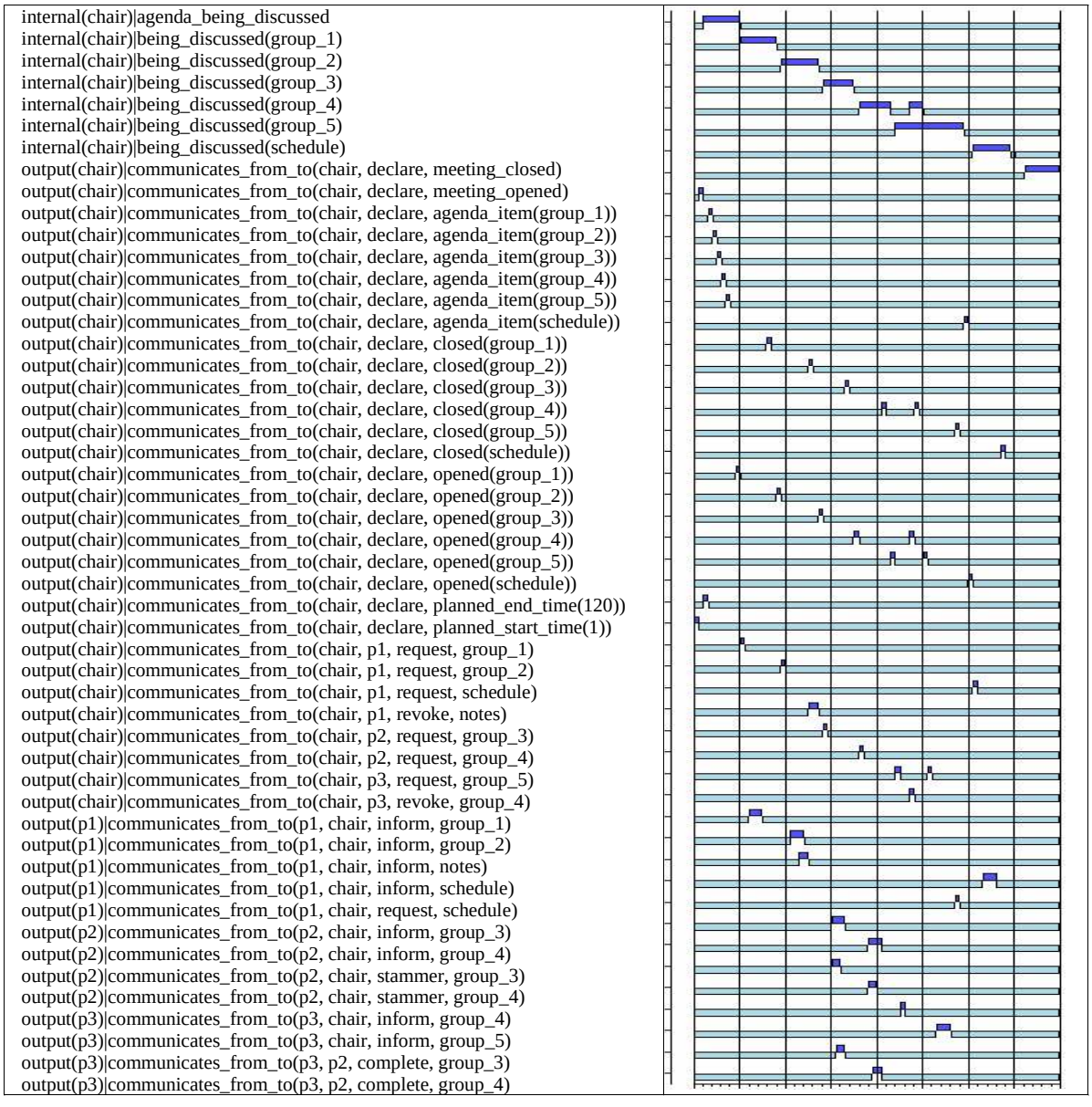
Formal

∀t, i1,i2 ∀m:CHAIR, p:PARTICIPANT
[current_agenda_item_at((γ, i2, t) &
addressed_agenda_item_at(γ, i1, t) &
state(γ, t, output(p)) |= communicates_from_to(p, m, inform, y) & in_context_of(i1)]
⇒ ∃ t"≥t state(γ, t", output(m)) |= communicates_from_to(m, declare, closed(i2)) &
state(γ, t", output(m)) |= communicates_from_to(m, declare, opened(i1))

RI2 If a Participant speaks on an item other than the current agenda item or any earlier addressed agenda item,
then the Chairperson revokes the Participant and asks for additional comments on the current agenda item from the other participants.

Formal

∀t, i2 ∀m:CHAIR, p:PARTICIPANT, y
[state(γ, t, output(p)) |= communicates_from_to(p, m, inform, y) &
not ∃i1 addressed_agenda_item_at(γ, i1, t) & in_context_of(y, i1)]
⇒ ∃ t"≥t state(γ, t", output(m)) |= communicates_from_to(m, p, permission, revoke) &
∀q state(γ, t", output(m)) |= communicates_from_to(m, q, request, info_on(i1))



	time	0	20	40	60	80

Table 3 Simulation trace based on the refined protocol

8 Discussion

In this paper a generic role interaction protocol for meetings that adhere to several guidelines on holding meetings was formalized, using the logical language TTL; cf. [Jonker and Treur, 2002]. Moreover, desirable overall properties for a meeting were formally specified. In a case study in terms of the desirable overall properties of a meeting, an empirical trace was compared with a simulated trace generated from the given meeting protocol. Based on deviations revealed in this comparison, a more human-like refined protocol was specified and used as a basis for another simulation, closely resembling the empirical data.

Croston and Goulding present one of the earlier empirical works on meeting effectiveness [1966]. Croston and Goulding develop a meeting analysis kit that is used in different departments of a company by the participants of the meeting. The kit enables the participants to reevaluate a past meeting by analyzing the topics discussed, the time spent on each topic, and so on. Based on the analysis from different meetings, Croston and Goulding observe that the starting a meeting with a formal agenda and better chairing of the meetings increase the effectiveness of meetings. The meeting protocol that we propose respects both of these observations. Further, we explicitly formalize the notion of better chairing a meeting.

Serman and Basili study various properties of software inspection meetings in a software development project [1998]. Similar to the generation of the empirical trace here, Serman and Basili collect data by attending inspection meetings as an observant. They later analyze their data statistically to uncover causal relations between various properties of the meeting, such as effectiveness, efficiency, or meeting length. While Serman and Basili discover interesting relations, they do not provide a formal protocol of how the meetings should be carried out as we have done here. Since our study uses simulations, we can easily adjust different behaviors of participants to see the effect of (local) properties of participants of a meeting on the (global) properties of the meeting as a whole.

Generally, the group-support systems help participants share data, improve communication, and reach decisions. Hence, group-support systems can help increase the efficiency of meetings. Niederman *et al.* study the meetings in organizations with group-support systems [1996]. Their primary focus is to show how the use of group-support systems by facilitators affects meeting performances. Through interviews with facilitators, Niederman et al. observe that different facilitators have different ideas on measuring performance. However, no formal rules for identifying or bringing out successful meetings are identified.

Given the informal literature as discussed, the work reported in the current paper contributes some first steps in formal analysis of meetings. It is shown how meeting simulations following widely accepted guidelines in a rigid manner, do not resemble human meetings, which exploit more sophisticated strategies. It is pointed out how this discrepancy can be overcome by allowing by exception violations of the protocol, and by including exception handling strategies within the protocol. Future research will address this theme further.

Acknowledgment

We thank the anonymous referees for useful comments.

References

Katleen M. Carley and Les Gasser. Computational Organization Theory in Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. Chapter 7. Gerhard Weiss, editor. MIT Press, 1999.

James L. Creighton. Using Group Process Techniques to Improve Meeting Effectiveness.
URL: http://www.effectivemeetings.com/teams/teamwork/creighton.asp

J. D. Croston and H. B. Goulding. The Effectiveness of Communication at Meetings: A Case Study. Operational Research Quarterly, Vol.17. No.1, pp. 47-57, March 1966.

Catholijn M. Jonker and Jan Treur. Compositional Verification of Multi-Agent Systems: a Formal Analysis of Proactiveness and Reactiveness. International Journal of Cooperative Information Systems, vol. 11, 2002, pp. 51-92.

Fred Niederman, Catherine M. Beise, and Peggy M. Beranek. Issues and Concerns about Computer-Supported Meetings: The Facilitator's Perspective. MIS Quarterly, Vol.20, No. 1, pp. 1-22, March 1996.

Carolyn B. Serman and Victor R. Basili. Communication and Organization: An Empirical Study of Discussion in Inspection Meetings. In IEEE Transactions on Software Engineering, Vol.24, No. 6, pp. 559-572, July 1998.

Kevin Wolf. The Makings of a Good Meeting. October, 2002. Available at: http://members.dcn.org/kjwolf

Appendix A Overall Organizational Behavior Properties

At the level of the overall organization (which in this case is the group as a whole) a number of organization properties have been identified.

OP1
Informal
During the meeting only one Participant is speaking at a time.
Semiformal
At any point in time,
if any participant is speaking,
then all other participants are not speaking
Formal
$\forall t, p, p'$:PARTICIPANT, q, q' :ROLE, x, x', y, y'
$p \neq p'$ & state(γ, t, output(p)) |= communicates_from_to(p, q, x, y) $\Rightarrow$
state(γ, t, output(p')) |≠ communicates_from_to(p', q', x', y')

OP2
Informal
During the meeting only agenda items are addressed.
Semiformal
At any point in time t,
if the item i is opened
then i is an agenda item
Formal
$\forall t, i, p, q, x, y$ $\forall$m:CHAIR
state(γ, t, output(m)) |= communicates_from_to(m, declare, opened(i))
$\Rightarrow$ agenda_item_at(γ, i, t)

OP3
Informal
Every Participant who indicates that he has something to say on the current agenda item will have the opportunity to speak.
Semiformal
At any point in time t,
if a participant communicates that he has something to say about the current agenda item i
then before the item was closed a later time point exists such that at t' the participant communicates something in the context of i
Formal
$\forall t, I, p$:PARTICIPANT, q:ROLE current_agenda_item_at(γ, i, t)
&
state(γ, t, output(p)) |= communicates_from_to(p, q, inform, has_input_for(p, i)) $\Rightarrow$
$\exists t' \geq t, x$
state(γ, t', output(p)) |= communicates_from_to(p, q, inform,x)
& is_in_context_of(x, i)

The notion of being in context of is assumed a given notion.

OP4
Informal
Eventually the meeting is closed.
Semiformal
At some point in time the chairperson declares the meeting closed
Formal
$\forall$m:CHAIR $\exists t$ state(γ, t, output(m)) |= communicates_from_to(m, declare, meeting_closed)

OP5
Informal
If the meeting is closed, all agenda items have been addressed.

Semiformal
At any point in time,
if the meeting is declared closed,
then for any item i that was on the agenda there are earlier time points at which item i was declared opened and closed
Formal
∀t, i, m:CHAIR
state(γ, t, output(m)) |= communicates_from_to(m, declare, meeting_closed) & agenda_item_at(γ, i, t) ⇒
addressed_agenda_item_at(γ, i, t)

OP6
Informal
No two items are current at the same time.
Semiformal
At any point in time t,
if item i is current at t,
and item i' is current at t,
then i = i'
Formal
∀t, i, i'
current_agenda_item_at(γ, i, t) & current_agenda_item_at(γ, i', t) ⇒ i = i'

OP7
Informal
If a participant is speaking, then she is speaking on the current item.
Semiformal
At any point in time t,
if at t the item i is current agenda item
and at t any participant is communicating X,
then X fits in item i
Formal
∀t, i, p, q :ROLE, x, y
current_agenda_item_at(γ, i, t) & state(γ, t, output(p)) |= communicates_from_to(p, q, x, y)
⇒ is_in_context(y, i)

OP8
Informal
The meeting starts and ends in time.
Semiformal
The meeting starts at the planned starting time
and ends before the planned end time
Formal
∀m:CHAIR ∀t1 [planned_starting_time(t) ⇒
state(γ, t1, output(m)) |= communicates_from_to(m, declare, meeting_opened))] &
∀t2, t3 state(γ, t2, output(m)) |= communicates_from_to(m, declare, planned_end_time(t3)) ⇒
∃t4 ≤ t3 state(γ, t4, output(m)) |= communicates_from_to(m, declare, meeting_closed))

OP9
Informal
Every communication in the meeting is received by everyone
Semiformal
At any point in time,
if a participant communicates something to another one,
then this communication will be received by everyone
Formal
∀t, p, q , q':ROLE, x, y
state(γ, t, output(p)) |= communicates_from_to(p, q, x, y) ⇒
∃t'≥t state(γ, t, input(q')) |= communicates_from_to(p, q', x, y)

OP10
Informal

The secretary will make minutes of the meeting

Semiformal

if an agenda item is closed,
then notes for the minutes on this item have been made by the Secretary

Formal

$\forall t, i \; \forall m$:CHAIR
state(γ, t, output(m)) |= communicates_from_to(m, declare, closed(i)) $\Rightarrow$
state(γ, t, EW) |= notes_present_for_by(i, Secretary)

OP11

Informal

The internal state property of the chairperson indicating that i is being discussed holds precisely then when i is a current agenda item

Semiformal

if an agenda item is closed,
then notes for the minutes on this item have been made by the Secretary

Formal

$\forall t, i \; \forall m$:CHAIR
current_agenda_item_at(γ, i, t) $\Leftrightarrow$
state(γ, t, internal(m)) |= being_discussed(i)

Appendix B Role Interaction Properties: the Generic Meeting Protocol

The following role interaction properties define an interaction protocol for the meeting.

RI1 If the Chairperson generates a question (which implies a permission to speak) to a Participant, then a little time later the Participant generates an answer.

Formal

∀m:CHAIR, p:PARTICIPANT ∀t
[state(γ, t, output(m)) |= communicates_from_to(m, p, request, q)) &
not ∃ x state(γ, t', output(p)) |= communicates_from_to(p, m, inform, x)]
⇒
∃t' > t state(γ, t', output(p)) |= communicates_from_to(p, m, inform, answer_on(a, q)))

RI2 If a Participant requests to add an item to the agenda,
then the Chairperson ommunicates this to all Participants.

Formal

∀m:CHAIR, p:PARTICIPANT ∀t
state(γ, t, output(p)) |= communicates_from_to(p, m, request, agenda_item(i)))
⇒
∃t' > t state(γ, t', output(m)) |= communicates_from_to(m, declare, agenda_item(i)))

RI3 If the Chairperson generates a permission to speak for a Participant,
then that Participant will begin speaking on the current agenda item.

Formal

∀m:CHAIR, p:PARTICIPANT ∀t
state(γ, t, output(m)) |= communicates_from_to(m, p, permission, speak)) &
current_agenda_item(i)
⇒ ∃t' > t, y
state(γ, t', output(p)) |= communicates_from_to(p, m, inform, y) & is_in_context_of(y, i)

RI4 If a Participant has finished speaking on the current agenda item, (observation result) or 3 minutes have passed since the permission was given (observation result), then the Chairperson revokes the permission to speak of that Participant.

RI5 If the Chairperson revokes the permission to speak from a Participant while that Participant is still speaking, then that Participant will stop speaking immediately.

Formal

∀t, i ∀m:CHAIR, p:PARTICIPANT
state(γ, t, output(m)) |= communicates_from_to(m, p, permission, no_permission_to_speak)
& state(γ, t, output(p)) |= communicates_from_to(p, x, y, z)
⇒ ∀x', y', z' state(γ, t+1, output(p)) |≠ communicates_from_to(p, x, y, z)

RI6 If all Participants who at an earlier point in time have indicated that they have information or a question regarding the current item, have put forward their information,
then the Chairperson asks each Participant in turn whether he has further information on the current item.

Formal

∀t, i ∀m:CHAIR
[∀p:PARTICIPANT [∃t"≤t state(γ, t", output(p)) |= communicates_from_to(p, m, inform, has_input_for(i)) ⇒
∃t"'≤t, y state(γ, t"', output(p)) |= communicates_from_to(p, m, inform, y) & is_in_context _of(y, i)]]
⇒ ∃ t'≥t state(γ, t, output(m)) |= communicates_from_to(m, request, further_info_on(i))

If all Participants have answered that they have no further information on the current item, then the Chairperson summarizes the discussion on the item and declares the item closed.

Formal

∀t, i ∀m:CHAIR
[∀p:PARTICIPANT state(γ, t, output(p)) |= communicates_from_to(p, m, inform, no_further_info_on(i))]
⇒ ∃ t'≥t state(γ, t, output(m)) |= communicates_from_to(m, declare, summary(i)) &
state(γ, t, output(m)) |= communicates_from_to(m, declare, closed(i))

RI7 If the Chairperson has declared an agenda item closed, and not all items have been treated, then the Chairperson will announce one of the remaining items as the current item.

Formal

∀t, i ∀m:CHAIR
state(γ, t, output(m)) |= communicates_from_to(m, declare, meeting_closed)
& ∃i agenda_item_at(γ, i, t) & not addressed_agenda_item_at(γ, i, t)
⇒ ∃i, t'≥t agenda_item_at(γ, i, t) & not addressed_agenda_item_at(γ, i, t) &
state(γ, t', output(m)) |= communicates_from_to(m, declare, opened(i))

RI8 If the Chairperson has declared the meeting opened,
then the Chairperson will announce the proposed end time.

Formal

∀t, i ∀m:CHAIR
state(γ, t, output(m)) |= communicates_from_to(m, declare, meeting_opened)
⇒ ∃t'≥t, t" state(γ, t', output(m)) |= communicates_from_to(m, declare, planned_end_time(t"))

RI9 If the Chairperson has proposed an end time,
then the Chairperson will announce the agenda items.

∀t, t", i ∀m:CHAIR
state(γ, t, output(m)) |= communicates_from_to(m, declare, planned_end_time(t")) &
agenda_item_at(γ, i, t)
⇒ ∃t'≥t state(γ, t, output(m)) |= communicates_from_to(m, declare, agenda_item(i))

RI10 If the Chairperson has announced all agenda items,
then the Chairperson will ask if any Participant has another agenda item.

∀t, t", i ∀m:CHAIR
∀i [agenda_item_at(γ, i, t) ⇒
∃t'< t state(γ, t', output(m)) |= communicates_from_to(m, declare, agenda_item(i))]
⇒ ∃t"≥t state(γ, t", output(m)) |= communicates_from_to(m, request, other_items)) &

RI11 If the Chairperson has declared the last agenda item closed and 10sec has passed,
then the Chairperson will close the meeting.

∀t, t", i ∀m:CHAIR
∀i [agenda_item_at(γ, i, t) ⇒
∃t'≤ t state(γ, t', output(m)) |= communicates_from_to(m, declare, closed(i))] &
∃i [agenda_item_at(γ, i, t) &
state(γ, t, output(m)) |= communicates_from_to(m, declare, closed(i))] &
⇒ ∃t"≥t t" < t + 10
state(γ, t", output(m)) |= communicates_from_to(m, declare, meeting_closed))