Formal Analysis of Meeting Protocols

Catholijn M. Jonker[a]   Martijn Schut[a]   Jan Treur[a,b]   Pınar Yolum[a]

[a] Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{jonker, schut, treur, pyolum}@few.vu.nl
[b] Universiteit Utrecht, Department of Philosophy
Heidelberglaan 8, 3584 CS Utrecht, The Netherlands

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 success 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 [7, 1]. 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 develops the formal generic meeting protocol. Section 3 studies a generic meeting trace based on the formalized protocols. Section 4 introduces an empirical trace of a real meeting. Section 5 analyzes both traces formally in terms of desired properties. Section 6 provides a revised protocol and a simulation of the enhanced protocol. Section 7 discusses the relevant literature in comparison to this work.

2 Meetings Formalized

In this section a formalization of the organization of a meeting is presented: organizational structure, dynamic properties for the overall process, and a protocol for role interactions.

2.1 Organizational Structure

Consider a typical meeting that contains a chairperson, a secretary, and a number of participants. A common form to structure meetings is the following. A Chairperson chairs every meeting. The Secretary takes minutes of the meeting. 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. A question to be addressed is how dynamic properties describing such a protocol can be identified.

2.2 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.

2.2.1 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 participants 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” act, then the receiver q is dropped meaning that the message is sent to everyone. For the sake of simplicity, we assume that messages always reach their destination. For an explanation of the formal language TTL used, see [3,4].

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

$$\forall t, p, p' : \mathrm{PARTICIPANT}, q, q' : \mathrm{ROLE}, x, x', y, y'$$
$$p \neq p' \;\&\; \mathrm{state}(\gamma, t, \mathrm{output}(p)) \models \mathrm{communicates\_from\_to}(p, q, x, y) \Rightarrow$$
$$\mathrm{state}(\gamma, t, \mathrm{output}(p')) \not\models \mathrm{communicates\_from\_to}(p', q', x', y')$$

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 agenda 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

$$\mathrm{agenda\_item\_at}(\gamma, i, t) =$$
$$\exists\, m{:}\mathrm{CHAIR}, t' \leq t$$
$$\mathrm{state}(\gamma, t', \mathrm{output}(m)) \models \mathrm{communicates\_from\_to}(m, \mathrm{declare}, \mathrm{agenda\_item}(i)) \;\&$$
$$\forall t''\quad t' < t'' < t \; \mathrm{state}(\gamma, t'', \mathrm{output}(m)) \not\models \mathrm{communicates\_from\_to}(m, \mathrm{declare}, \mathrm{not\_agenda\_item}(i))$$

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

$$\mathrm{current\_agenda\_item\_at}(\gamma, i, t) =$$
$$\exists m{:}\mathrm{CHAIR}, t' \leq t$$
$$\mathrm{state}(\gamma, t', \mathrm{output}(m)) \models \mathrm{communicates\_from\_to}(m, \mathrm{declare}, \mathrm{opened}(i)) \;\&$$
$$\forall t''\; [\, t' < t'' < t \Rightarrow \; \mathrm{state}(\gamma, t'', \mathrm{output}(m)) \not\models \mathrm{communicates\_from\_to}(m, \mathrm{declare}, \mathrm{closed}(i))\,]$$

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) =
∃ m:CHAIR, 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))

2.2.2 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.

3 Simulating a 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, for more details see [3]. These executable temporal rules are executed based on the current status of the world, without regard to the past. A generated trace describes which state 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.

Simulation as 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. 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. A complete trace can be found in [3].

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.

4 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 [6], 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**
…		

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)

…		
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))
…		
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))
…		
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 1 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 1 gives brief snapshots from this, for a complete formalized trace see [3]. 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 2.2.2, 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 3), 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.

5 Formal Analysis of Simulated Trace and Empirical Trace

We analyzed the traces generated by these simulations in terms of the organization properties defined above, for more properties see [3]. To do so, the organization properties of Section 2 (and more) have been entered into the checker and automatically checked against each trace.

5.1 Analysis of the Simulated Meeting

The meeting simulated according to the generic protocol (Sections 2 and 3) 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 intention to speak 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 is 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.

5.2 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 chair person 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 chair person 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).

6 Refined Protocol and Simulation

As shown in the analysis in Section 5, a real meeting (such as the one described in Section 3) may deviate from a meeting correctly following the protocol (such as the simulated meeting in Section 4) 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 is 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 3) 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 4. 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))

Using these rules, a new trace was generated that shows how participants can accommodate these exceptions. More information on this trace can be found in [3].

7 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. [3]. 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 [2]. 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 [6]. 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 [5]. 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

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

[2]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.

[3]Catholijn M. Jonker, Martijn Schut, Jan Treur, and Pinar Yolum. Formal Analysis of Meeting Protocols. Vrije Universiteit Amsterdam, Department of Artificial Intelligence. Technical Report, 2004. Available at: http://www.few.vu.nl/~wai/Papers/TR2004meeting.pdf

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

[5]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.

[6]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.

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