Agent-Enhanced Collaborative Activity in Organized Settings

Ioannis Partsakoulakis and George Vouros
Department of Information and Communication Systems Engineering
University of the Aegean
Samos, 83200, Greece

For groups of agents to act collaboratively, they need to recognize the need for collaboration, decide on the method to be followed for achieving goal states, assign responsibilities to subgroups and individuals, and so on, until responsibilities that can be fulfilled by individuals are reached. Aiming to support collaborative activity of humans within organized settings, this paper introduces a set of constructs for specifying organizational structures and introduces an explicit representation of individual and collaborative responsibilities within organizations. We conjecture that group members create common awareness towards recognizing the need for collaboration by forming group acceptances. Acceptances are formed by means of shared practices and beliefs of individual agents. The paper introduces state recognition recipes that drive group members within organizations to form acceptances, and thoroughly explains the exploitation of these recipes in conjunction to state achievement recipes for achieving goal states and fulfilling responsibilities collaboratively.

1 Introduction

The concept of personal agents, i.e. of software entities that help their users and act on behalf of them, is one of the first application paradigms of agent technology. Today, it still remains an important paradigm for realizing systems in many important research areas such as e-Commerce, Ambient Intelligence, and the Semantic Web. However, until now personal agents specialize in some tasks ignoring most facets of human activity and the context in which these activities take place[1]. The social context of humans comprises the groups in which humans participate or they have a special interest in.

The social context constantly changes, affecting human activity and the ways tasks are accomplished. For example, a personal agent for email filtering may have learned to give priority to email messages from the boss of its user. However, if the user changes position in the company, then his priorities concerning email messages change, too; the personal agent must possess knowledge on user’s social context and must adapt its behavior according to it.

To support collaborative human activity in well-organized settings we consider that multi-agent systems form the digital analogues of human organizations. Agents represent humans within the digital organization and must be aware of the social context in which their users act. In these digital analogues of human organizations, representative agents play the roles that their human partners play within the human organization and have the same responsibilities, helping humans to fulfill their responsibilities individually or in collaboration with their colleagues. Therefore, the representatives engage in the same activities in which their respective users do, and keep track of these activities in order to provide helpful behavior. It is assumed that although, like their human partners, agents may decide with selfish criteria to enter or leave an organization, when they act in the context of an organization, their aim is not to deceive their colleagues but to act as members of the organization in the core sense\textsuperscript{2}. Subsequently, referring to individual agents or groups of agents playing specific roles within organizations, we mean representatives or groups of representatives that play the corresponding roles within digital organizations.

The research described in this paper is based on the conjecture that personal agents that are aware of and exploit the social context of their users have a better potential to be deployed successfully towards helping humans achieving their goals\textsuperscript{3}. Specifically, this research aims to support humans in collaborative activities that take place in well-organized groups by means of personal agents that are aware of the social context of their users. During collaborative activity, group members act like a single entity by sharing knowledge, creating common awareness, sharing practices and preferences, and building and maintaining models of their peers. Towards their common endeavor, group members must pursue their group and individual responsibilities, contributing to the goals of the organization. Major questions to be answered include:

- What is the impact of agents’ social context in group decision making (i.e. on deciding what responsibilities to pursue)?
- How groups create and maintain group awareness in organized settings?
- How groups plan towards fulfilling their collaborative responsibilities in well-organized settings in conjunction to creating common awareness?
- What is the impact of agents’ organized setting in the fulfillment of their responsibilities?

To answer these questions, the paper distinguishes between individual beliefs and group acceptances, as well as between individual and commonly agreed goals. Specifically, in order a group to act as a single entity, it must accept a common view that
depends on the individual beliefs of its members and commit to the collaborative performance of agreed goals. The paper introduces an explicit representation of individual and group responsibilities, and focuses on the collaborative recognition of acceptances and on the collaborative achievement of goal states in well-organized settings.

Most theoretical models of collaborative decision making in MAS literature[4, 5, 6, 7] adopt a summative account of group belief to cope with group awareness. According to this, a group of agents collectively believe a state $\alpha$, if every individual in the group believes $\alpha$ and everyone in the group believes that everyone believes $\alpha$, and so on. This definition is grounded on individual views (or beliefs) of group members and it requires that all group members have the same view. However, in settings where information is inherently distributed and access restrictions to information sources apply, not all group members have the possibility to form a personal view. Furthermore, each group member, according to its position, may have a “view” which can be different (or “complementary”) to the views of the others in the organization. For example, only some employees in the financial department of a specific organization, as well as some executive members can be aware of some critical economic figures. In this case, not all company members are aware of a particular situation. In these cases, even if group members have different views of a situation, the group must form a decision and be prepared to act as a single entity. To do so, group members, acting as such, are not required to have a collective view, but they need to possess policies for reaching a view that all members shall accept. Accepting the view of the more experienced group member or the view of the majority are examples of such policies. Finally, the inherent infinite nature of collective belief, as it has been specified above, causes difficulties to its implementation. To overcome these difficulties collective beliefs are computed by means of approximations. For instance, one such approximation is that for a group to collectively believe $\alpha$, it must hold that all agents believe $\alpha$ and all agents believe that all agents believe $\alpha$. The problem here is that such approximations may differ from system to system and are often implicit in the implemented systems, affecting their openness and their ability for dynamic re-structuring.

Besides the summative accounts of group beliefs there is also the non-summative ones[8, 9, 10, 11]. According to the non-summative approaches it is not necessary that all or most of group members to believe a state $\alpha$, in order for the group to believe $\alpha$ [9]. Therefore, the non-summative account of group belief is more appropriate for organized groups where some of the group members decide the beliefs of the group as a whole. However, there is not yet a framework that explains how group beliefs are being formed in well-organized groups by means of group policies.

This paper adopts a non-summative account of group belief and presents a framework for their computation based on group policies that group members share. Specifically, in this paper, according to Tuomela[11], we consider that a well-organized group believes a state $\alpha$ if the following conditions hold: (a) Some members of the group collaboratively decide to accept $\alpha$, which is a view that all group members shall accept, (b) the other members of the group accept $\alpha$, and (c) all the members of the group know that $\alpha$ has been accepted by the group. Furthermore we specify that the group accepts $\alpha$, or that there is a group acceptance concerning $\alpha$, if there is a group belief concerning $\alpha$. 
The above introduce the problem of representing and exploiting policies for collaboratively deciding when a state shall be accepted by a group. Such policies are important for the formation and maintenance of group acceptances. For instance, a policy may state that group members shall accept something if the more experienced members of the group believe it: In other words, the more experienced members of the group must exploit this policy to form a decision that all other members shall accept. More than policies, this example reveals the necessity for agents to clearly distinguish between their individual beliefs, and group acceptances. Based on this distinction, we assume that group members may accept a state, even if there are individuals that do not believe it.

The concept of acceptance has lately received the attention of philosophers\cite{8, 12, 13, 14, 15}. According to Engel\cite{14}, acceptances differ substantially from beliefs mainly in the following points:

(a) Accepting a state is a matter of holding that state as being true. This is a matter of decision in contrast to beliefs that are a kind of disposition.

(b) Acceptances in contrast to beliefs are context dependent. Therefore, an agent can hold different acceptances as a member of different groups.

(c) The set of acceptances (in contrast to the set of beliefs) is not subject to an ideal of integration and agglomeration. This means that acceptances of different groups can be conflicting.

As already stated, this paper proposes the use of group policies that enable some of the members of a group to reach decisions concerning acceptances. Dealing with well-organized settings, we consider that policies represent organization practices that all members of a group think that are appropriate to reach decisions on a common view. Since different groups may have different policies, a single agent participating in different groups may hold different acceptances, even if the beliefs upon which these acceptances depend are the same. Finally, assuming that an agent can hold conflicting acceptances, we do not claim that consistency is not an important issue. An agent should seek to participate in groups that are “close” to its personal beliefs and goals. Therefore, we assume that once an agent is a member of a group then it does not “select” the group attitudes that fit best to its personality, but is a group member in the core sense. However, we assume that a single group cannot hold conflicting acceptances.

Concluding the above, based solely on a summative account of group belief, agents cannot tolerate different beliefs towards common group awareness, leaving outside many interesting real-world group settings. This paper distinguishes between group and individual beliefs\cite{13, 16} and proposes group policies towards the formation of acceptances. Acceptances enable groups of agents to share a common view and pursue their responsibilities collectively, acting like a single entity. The paper focuses on a non-summative account of group belief, and on the collaborative achievement of goal states in well-organized settings. Finally, it introduces state recognition recipes (r-recipes) and goal achievement recipes (a-recipes), and thoroughly shows the exploitation of these recipes in such settings.
The rest of the paper is structured as follows: Section 2 presents a real-life application scenario that motivates our work and reveals the requirements for personal agents. Section 3 presents the constructs for building organizational models and organizational structures. Section 4 presents in rough details the fulfillment of individual and collaborative responsibilities, providing an overview of the proposed approach. Section 5 presents state recognition recipes and their exploitation for the collaborative formation of acceptances. Acceptances of the conditions of collaborative responsibilities drive groups to recognize the need for collaboration. Section 6 presents state achievement recipes and their exploitation for the achievement of collaborative responsibilities’ goal states. Section 7 presents the communication cost for forming acceptances. Section 8 presents related work and finally Section 9 concludes the paper.

2 Motivation and requirements

This section presents a real-life scenario that motivates our work and reveals most of our considerations for building multi-agent systems that are able to support humans to fulfill their responsibilities either individually or in collaboration with colleagues in well-organized settings.

Mike is a newcomer engineer in a large company and he is employed in the engineering department. His work consists of getting customer orders, designing custom products by gathering the best available parts according to regulations and standards, and providing instructions to the manufacturing unit for getting the final product. The company, as well as the department in which Mike belongs has certain responsibilities. Some of these responsibilities are addressed to individuals while others are responsibilities of his department or of the company as a whole. The latter are called collaborative responsibilities and, in contrast to the individual responsibilities, they require group members to collaborate towards their fulfillment.

Being a newcomer, Mike is not aware of his colleagues’ regular activities, has not hands-on experience on the tasks he must perform and he does not possess proper knowledge on how problems are handled when they arise. Mike’s capabilities (e.g. designing and engineering products), permissions (e.g. access to information sources), physical and organization context (e.g. individual responsibilities and group responsibilities that he shares with colleagues, organizational structures and groups he participates in) and intentional context [17] (individual and group activities, constraints and permissions related to these activities) drive the way Mike fulfils his responsibilities.

The digital representative of Mike is an agent that holds knowledge concerning the capabilities of Mike, as well as his organizational and physical context. This agent represents Mike in the digital counterpart of the company, which is a multi-agent system comprising the digital representatives of company members. When Mike enters or leaves the company, then his digital representative enters or leaves this multi-agent system as well. It is this agent that presents to Mike all the necessary information about the organization (e.g. the organizational structure, roles and responsibilities of associated roles, methods for achieving these responsibilities etc.) and helps him achieve his goals.
individually or in collaboration with his colleagues.

Due to security restrictions or to constraints imposed by the organization, information is distributed to company members. This requires organizational members to be able to combine different pieces of information in order to agree about the need to achieve certain goals. For instance, let us assume that according to organization policies, Mike, as well as all the members of the engineering department, need different and complementary pieces of information to reach a decision about the design of a product. When group members in the engineering department receive these pieces of information, then they must form a commonly accepted view about the design of this product. To do so, group members exploit certain policies that reflect organizational practices and that allow information to be combined towards a commonly accepted view of the world. As already pointed, this is an important prerequisite for the group to act as a single entity, especially when group members possess conflicting pieces of information. Having an accepted view of the world, representative agents must help humans to pursue their responsibilities, exploiting the organizational structure and practices.

We distinguish between three modes of helpful behavior that a personal agent can offer to its user towards the fulfillment of a specific responsibility. In the first mode, the user decides to fulfill the responsibility without the help of his personal agent. In this case, the personal agent keeps the responsibility in the agenda of the user and presents this responsibility if such a need arises; however, it does not act on behalf of the user. To achieve this behavior, a personal agent must be knowledgeable about the organizational and intentional context of the user.

In the second mode, the user delegates the responsibility to the agent. In this case, the personal agent must not only be knowledgeable about the organizational context of the user, but it must also have the appropriate capabilities and knowledge to fulfill the delegated responsibility. In case the delegated responsibility is a collaborative one, then the agent must have collaboration abilities and must be able to cope with the distribution of knowledge.

Finally, in the third mode, the user collaborates with the agent for the fulfillment of the responsibility. This collaboration is independent on whether the responsibility is collaborative, as it concerns the relation between the user and his representative. In this case the user and the agent commit themselves to proceed to the fulfillment of the responsibility together. The agent must be able to track the activity of the user and to form decisions with him about the fulfillment of the responsibility.

In this paper we focus on the capabilities of agents to fulfill their responsibilities without collaborating with their human partners. We believe that this is a necessary step towards an integrated form of helpful behavior that realizes all three help modes mentioned above.

### 3 Roles, positions, groups, and responsibilities

Dealing with collaborative activity in organized settings we need to specify the structure of organizations. This section specifies the modeling constructs of roles, positions
and groups, which are necessary for specifying the collective aspects of group activity. Capturing every aspect of organizational models is out of the scope of this paper. Fig. 1 shows the modeling constructs that are thoroughly explained in subsequent paragraphs.

The central construct of the organizational model is that of role. A role serves as a prototype that specifies the behavior of an individual or of a set of individuals that form a group. Examples of roles are “company”, “department” and “seller”. Roles have several constituents for the specification of individuals’ and groups’ behavior and can be related to each other with various relations[18, 19, 20, 21]. In this paper, as Fig. 1 shows, we assume that each role has been assigned specific responsibilities and comprises recipes for pursuing these responsibilities.

Roles are interrelated via the transitive relation “contains”. Through the relation “contains” between roles, a role aggregation hierarchy is created. Roles are distinguished into atomic roles, which do not contain other roles, and composite roles that contain at least one role. Fig. 2 shows a small example of a role aggregation hierarchy. The roles “customer”, “seller” and “engineer” are atomic roles, while the roles “company”, “customer department”, “engineering department” and “manufacturing department” are composite roles. According to this specification, in any company we can distinguish customer departments, engineering departments and manufacturing departments. In

Figure 1: Reference meta-model for organizations.
any customer department we can distinguish sellers and customers, in any engineering
department there are engineers and in any manufacturing department there are manu-
facturers.

As Fig. 1 shows, the association between agents or groups of agents and roles is done
through (organizational) positions. We distinguish between roles, that are abstract
entities, and positions, that are instantiations of roles. This is motivated by the need to
have organizations where more than one group/individual plays the same role, and by
the need to distinguish the members of each group. For example, if an organization has
two engineering departments and each such department has two engineers, then we must
distinguish between these two departments by means of specific positions that correspond
to the role “engineering department”. Furthermore, we must distinguish engineers by
means of distinguished positions that correspond to the role “engineer” and by relating
these positions to the specific departments that these engineers belong. Conclusively,
positions are place-holders for individuals or groups of agents. They correspond to roles
and are related via the “contains” relation, as well. Positions that correspond to atomic
roles are called atomic positions while those that correspond to composite roles are called
composite positions. Any containment relation between positions that has not been
anticipated in the roles’ model cannot be allowed. Positions inherit the responsibilities
and the recipes of the corresponding roles; however, they can have additional individual
responsibilities and recipes that differentiate each position from the other positions of
the same role. Additional responsibilities, as it will be explained in Section 6, can be
assigned to positions during the collaborative achievement of a group state. For example,
in order a customer department to achieve the goal of sending a product to a customer,
a specific seller must find the address of the customer and arrange the shipment. This
responsibility is assigned to a specific seller and not to the role “seller”. In the latter
case, the responsibility would be assigned to all the sellers. As far as the dynamic
assignment of recipes to positions, we do not deal with this case in this paper.

Fig. 3 shows the positions that correspond to the roles in the hierarchy of Fig. 2. As Fig. 3 shows, the “contains” relation between positions is in accordance to the
“contains” relation between roles. For example, the position “Engineer2”, which corre-
sponds to the role “engineer”, is contained in the position “E-dept”, which corresponds
to the role “engineering department”. “E-dept” is further contained in the position “The Company”, which corresponds to the role “company”. Generally, the “positions layer” is referred to as the organizational structure, while the “roles layer” is referred to as the organizational model.

In a given organization there is at most one individual for each atomic position, as there may be atomic positions that have not been occupied by any individual. The set of agents in a composite position $p$, i.e. the set of agents that occupy atomic positions that are contained in $p$, is considered to be a group that occupies $p$. We refer to this group with the name of the position that it occupies. In this sense, we can say that groups of agents are assigned to composite positions. We represent groups of agents with the capital letter $G$. When a set of agents is assigned to a composite position, then we say that the group plays the corresponding role. The relation “plays” holds between an individual agent and an atomic role, when the individual occupies a position that corresponds to that role.

Given a role aggregation hierarchy, such as the one shown in Fig. 3, the designer must assign responsibilities to the roles and the positions$^1$.

As Fig. 1 shows, a responsibility comprises a goal state and a condition, and is denoted as a rule of the form $c \Rightarrow g$ where $g$ is the goal state and $c$ the corresponding condition. Generally, a state is an atomic first-order predicate-logic formula that may

$^1$Although there can be agents within an organization whose task is to change the organizational structure and the assignment of responsibilities to positions dynamically, this paper does not deal with this case.
contain free variables. Goal states and conditions are states. The application of a list of substitutions \( \sigma \) to a state \( \alpha \) is denoted by \( \alpha|_{\sigma} \). The resulting state can be a ground instance that contains no free variables or a partially instantiated state.

In case a responsibility has variables, then it is actually a responsibility template and at a specific time point it can have several instantiations. For example, the responsibility, “if there is a customer order, then it must be carried out” is a responsibility template that has one instantiation for each customer order known at a specific time point. A responsibility is denoted by the small letter \( r \), and a set of responsibilities by the capital letter \( R \).

When the designer of the organization assigns a responsibility to a group, then he may consider this group as a list of individuals or as a meta-agent[22, 23, 24]. For example, when the designer assigns the preparation of a presentation to a department, then he may want each member of the department to achieve this goal independently from the others, or he may want them to collaborate to prepare a single presentation. Obviously, considering a group as a list of individuals or as a meta-agent, does not have to do with the properties of the group but with the intention of the designer. Therefore, referring to the same group, the designer may wish to consider it as a list of individuals or as a meta-agent. We use the notation \( R\{G\}r \) to denote the fact that group \( G \) is responsible to act towards fulfilling the responsibility \( r \) as a meta-agent, and the notation \( R\langle G\rangle r \) to denote the fact that members of \( G \) are responsible to act towards fulfilling \( r \) independently from each other\(^2\). In the former case the responsibility is a collaborative responsibility, while in the latter case the responsibility is an individual responsibility undertaken by each agent in the group. It must be pointed that atomic roles have only individual responsibilities. In addition to these types of responsibilities, we consider hybrid responsibilities. For a hybrid responsibility \( r \), we use the notation \( R\{G\}r \) to denote the fact that members of \( G \) must act as a meta-agent for recognizing instances of \( r \) and as a list of individuals for achieving the corresponding goal states of these instances.

If \( c \Rightarrow g \) is an individual responsibility of a role that a group plays, then each group member that recognizes instances \( c|_{\sigma_i}, i = 1, 2, \ldots \) of \( c \) individually (i.e. each agent recognizes its own instances), must individually see to it that the corresponding instances \( g|_{\sigma_i}, i = 1, 2, \ldots \) of \( g \) hold. On the contrary, in case \( c \Rightarrow g \) is a collaborative responsibility of a role that a group plays, then each group member must contribute to the recognition of instances \( c|_{\sigma_i}, i = 1, 2, \ldots \) of \( c \). In case such an instance \( c|_{\sigma_i} \) is accepted by the group members, then the group must collaboratively see to it that the corresponding instance \( g|_{\sigma_i} \) of \( g \) hold. The processes of forming acceptances and collaboratively seeing to it that a goal state holds are explained in detail in Sections 5 and 6, respectively. Finally, for a hybrid responsibility \( c \Rightarrow g \) of a role that a group plays, each group member must contribute to the recognition of every instance \( c|_{\sigma_i}, i = 1, 2, \ldots \) of \( c \), and in case such an instance \( c|_{\sigma_i} \) is accepted by the group members, then every group member must individually see to it[25] that the corresponding instance \( g|_{\sigma_i} \) of \( g \) holds.

Let us show the differences between these three types of responsibilities with an example. Assume that the responsibility “if the order is pending, then the order is prepared”

\(^2\)The notation is consistent to that used by Normal et al[24].
is a collaborative responsibility for the role “customer department”. In this case, members of the “C-dept” must collectively recognize that specific orders are pending. This is done by forming group acceptances for the instances of the condition state. If the members of the group “C-dept” accept that an order is pending, then they shall try together to find a plan to prepare this order. On the other hand, in case the same responsibility is specified to be an individual one for the role “customer department”, then each member of the “C-dept” that will believe that an order is pending shall try to prepare this order individually, without even trying to inform the other members of the “C-dept” about its belief. Finally, if the same responsibility is a hybrid responsibility of the “C-dept”, then each member of this group that will come to believe that an order is pending shall communicate this belief to the others in the group, aiming for the group members to accept this state. Having accepted this state, each member of the “C-dept” group shall try to prepare the order independently from the others.

If $i_{\text{resp}}(r_1, \rho)$, $c_{\text{resp}}(r_2, \rho)$, and $h_{\text{resp}}(r_3, \rho)$ denote that the responsibilities $r_1$, $r_2$, and $r_3$ of the role $\rho$ are individual, collaborative and hybrid responsibilities respectively, and $\text{plays}(G, \rho)$ denotes the fact that the group $G$ plays the role $\rho$, then we can state the following axioms:

**Axiom 1.** $i_{\text{resp}}(r_1, \rho) \land \text{plays}(G, \rho) \Rightarrow R_G(r_1)$.

**Axiom 2.** $c_{\text{resp}}(r_2, \rho) \land \text{plays}(G, \rho) \Rightarrow R_{\{G\}}(r_2)$.

**Axiom 3.** $h_{\text{resp}}(r_2, \rho) \land \text{plays}(G, \rho) \Rightarrow R_{\{G\}}(r_2)$.

Subsequent sections thoroughly explain how responsibilities are fulfilled through the processes of recognizing instances of condition states (i.e. through recognizing group beliefs and forming acceptances about these states) and of achieving the corresponding goal states.

## 4 The fulfillment of responsibilities

The operation cycle of an agent is a fairly simple one: The agent (a) perceives the environment, (b) communicates i.e. it checks for incoming messages and sends messages that are in its outbox, and (c) selects a responsibility of a position whose condition is accepted or believed, and sees to it that the corresponding goal state holds.

The pursuit of a responsibility $c \Rightarrow g$ comprises two main tasks: the formation of group acceptances concerning $c$’s instances and the achievement of the corresponding instances of $g$. The execution of each task depends on whether the group is considered as a list of individuals (i.e. the responsibility is an individual one) or as a meta-agent (i.e. the responsibility is a collaborative one). Fig. 4 illustrates how the different kinds of responsibilities are handled by an agent: There are distinct processes for the recognition of the instances of $c$ and for the achievement of the corresponding instances of the goal state: one for individual and one for the collaborative case. Specifically, individual responsibilities $c \Rightarrow g$ are routed to the individual recognition of instances $c_{|\sigma}$ of the condition $c$. The corresponding instances $g_{|\sigma}$ of the goal state are routed to the individual achievement process. Collaborative responsibilities are routed to the collaborative recognition of instances $c_{|\sigma}$ of the condition $c$, and then, the corresponding instances
$g|_{\sigma}$ of the goal state are routed to the collaborative achievement process. Finally, as far as hybrid responsibilities are concerned, these are routed to the collaborative recognition of the condition state instances, and then, to the individual achievement of the corresponding goal states.

### 4.1 Individual responsibilities

For an agent to recognize the instances of a responsibility $c \Rightarrow g$ individually, it must be based solely on its own individual beliefs. We consider that each agent owns a set of state recognition actions (shortly r-actions) that it can use to generate individual beliefs concerning instances $c|_{\sigma}$ of the state $c$. Each r-action is associated with a state that is called the relevant state of the r-action. These actions may exploit knowledge about the physical as well as about the social environment. The specific language for specifying r-actions depends on the platform in which agents are being developed.

Given the individual responsibility $c \Rightarrow g$ and for every recognized instance $c|_{\sigma}$ of $c$, the agent proceeds to see to it individually that the state $g|_{\sigma}$ holds. Seeing to it that a goal state holds comprises different kind of intentional activities. For example, according to Tuomela[16], taking as an example the goal state “the window is open” the following cases of intentional activity fall under an agent’s seeing to it that this goal holds: the agent opens the window (by his own direct actions), if it is closed; he keeps it open, if some other agent tries to close it; the agent gets some other agent to open the window, if it is closed; or he refrains from preventing another agent from opening the window.
In this paper we assume that the achievement of goal states in an individual manner is
done by agent’s own direct actions.

For an agent to individually see to it that an instance of a goal state holds, it must
be able to check whether the goal state already holds or it is unattainable. If it does
not hold and it is attainable, then the agent attempts to achieve it. For individuals,
the check whether a goal state holds or it is unattainable is done by means of r-actions.
The achievement of a goal state is done by means of state achievement actions (shortly
a-actions). These actions, similarly to r-actions, have a relevant state that is being
achieved when they are executed successfully.

Although the fulfillment of individual responsibilities is an important issue, this paper
focuses on collaborative responsibilities. These responsibilities drive groups to form
acceptances and to collaborate towards the achievement of specific goal states.

4.2 Collaborative responsibilities

The elaboration of collaborative responsibilities requires collaborative activity that com-
prises the following four distinct stages: Recognition of the need for collaboration, group
formation, recipe selection, and responsibility allocation. We briefly describe each of
these stages below.

4.2.1 Recognition of the need for collaboration

This is an acceptance formation task that results to the acceptance that an instance
\( c|_\sigma \) of a state \( c \) holds. Given a collaborative responsibility \( c \Rightarrow g \) that a group \( G \) has,
the acceptance of \( c|_\sigma \) equals to \( G \)'s recognition of the need to collaborate towards the
achievement of \( g|_\sigma \). The acceptance of an instance of a condition state can be a tedious

In this paper we assume that the achievement of goal states in an individual manner is
done by agent’s own direct actions.

For an agent to individually see to it that an instance of a goal state holds, it must
be able to check whether the goal state already holds or it is unattainable. If it does
not hold and it is attainable, then the agent attempts to achieve it. For individuals,
the check whether a goal state holds or it is unattainable is done by means of r-actions.
The achievement of a goal state is done by means of state achievement actions (shortly
a-actions). These actions, similarly to r-actions, have a relevant state that is being
achieved when they are executed successfully.

Although the fulfillment of individual responsibilities is an important issue, this paper
focuses on collaborative responsibilities. These responsibilities drive groups to form
acceptances and to collaborate towards the achievement of specific goal states.

4.2 Collaborative responsibilities

The elaboration of collaborative responsibilities requires collaborative activity that com-
prises the following four distinct stages: Recognition of the need for collaboration, group
formation, recipe selection, and responsibility allocation. We briefly describe each of
these stages below.

4.2.1 Recognition of the need for collaboration

This is an acceptance formation task that results to the acceptance that an instance
\( c|_\sigma \) of a state \( c \) holds. Given a collaborative responsibility \( c \Rightarrow g \) that a group \( G \) has,
the acceptance of \( c|_\sigma \) equals to \( G \)'s recognition of the need to collaborate towards the
achievement of \( g|_\sigma \). The acceptance of an instance of a condition state can be a tedious
task when information is highly distributed to group members i.e. when group members
have limited reasoning capabilities and restricted access to information sources. Distin-
guishing between group acceptances and individual beliefs[13, 26], acceptances about a
state \( \alpha \) are formed even if there are individuals that do not believe that this state holds.
For the formation of acceptances, group members need specific organization policies.
Such policies are being specified by means of state recognition recipes (shortly r-recipes).
R-recipes specify the required beliefs of individuals and the required acceptances of sub-
groups that drive group members to recognize group beliefs. Section 5 describes the
form and use of r-recipes.

4.2.2 Group formation

As already being said, collaboration of a group \( G \) of agents begins when group members
accept that an instance of the condition \( c \) of a collaborative responsibility \( c \Rightarrow g \) of \( G \)
holds. In this case, agents in \( G \) do not need to engage in a group formation session, as
all group members accept an instance \( c|_\sigma \) of the responsibility condition. This accep-
tance drives the group members to assume that there is a commonly agreed goal state,
specifically \( g|_\sigma \) that the group shall achieve collaboratively. Each group member must
check and verify that the commonly agreed goal does not contradict the other goals it
current has. This may cause some agents to declare unavailability due to consistency reasons. In this paper, we do not deal with goals’ consistency checking.

Of course, according to the above, there may be cases where there are not enough group members to fulfill a specific responsibility. This may result to a failure of the group to achieve the commonly agreed goal and therefore it may result to organization’s reduced performance. To tackle this we can impose special restrictions to the assignment of agents to roles in order to minimize conflicts between goals[27, 28], or we can introduce special responsibilities to group members that drive them to monitor and/or evaluate other group members. Assuming that there are always enough group members for the achievement of a commonly agreed goal, this paper does not deal with these cases.

4.2.3 Recipe selection

This stage is initiated when the group $G$ has a commonly agreed goal state that does not hold and is attainable. Upon completion of this stage the group members have a commonly accepted recipe for achieving this goal. In organized settings there are common practices according to which a group can achieve its goals. Usually, such practices distribute responsibilities to group members or subgroups. To represent these practices we introduce state achievement recipes (shortly a-recipes). The specification and use of a-recipes is presented in Section 6.

4.2.4 Responsibility allocation

An a-recipe specifies the quantity and quality of subgroups or individuals that shall undertake specific responsibilities towards the goal state. However, the group must decide which are the specific subgroups or individuals that shall undertake the corresponding responsibilities. This gives to the group the opportunity to self-manage, at least partially, responsibility allocation within its members. Therefore, when a group commits to a certain a-recipe, then the responsibilities that this recipe specifies must be allocated to subgroups or to individuals that are members of this group. Then the selected individuals or subgroups must add the corresponding responsibilities to their agenda.

5 Collaborative formation of acceptances

As already pointed in Section 1, since agents act as group members in the core sense, it is assumed that when group members recognize a group belief about a state $\alpha$, then they accept that this state holds. Therefore, in organized settings an acceptance is formed by means of state recognition recipes (r-recipes) and is based on the individual beliefs of group members. This section specifies r-recipes in detail and presents the mechanism for the formation of acceptances.
Figure 5: The constituents of an r-recipe. An r-recipe serves as a policy for the formation of group acceptances towards instances of the recipe state.

5.1 R-recipes

The way pieces of information contribute to the acceptance of certain states is specified by r-recipes. An r-recipe belongs to a specific role, which is called the recipe’s “relevant role”. Any member of a group that plays this role knows this recipe: the recipe is a policy for the group members to form acceptances towards the recipe state. Each group that plays the recipe’s relevant role is a “relevant group” for this recipe. The constituents of an r-recipe are shown in Fig. 5.

The body of an r-recipe comprises the recipe state and elements of the form $\rho_{\text{ind}} : S$, where $\rho$ is an internal role, $\text{ind}$ an indicator, and $S$ is a template that denotes subsidiary states of recipe’s state. The indicator is a quantifier for the players of $\rho$ and in this paper we restrict its value to be “all” or “one”, indicating all the players of $\rho$, or at least one of them, respectively. Each internal-role of the r-recipe must be contained (via the relation “contains”) to the recipe’s relevant role. The following is an example of an r-recipe:

$$\langle \text{pending}(P, C), \{\text{customer}_{\text{one}} : \text{want–product}(P, C), \text{seller}_{\text{all}} : \text{pending}(P, C)\} \rangle$$

This recipe belongs to the role “customer department” with recipe state $\text{pending}(P, C)$ and states that the members of a customer department shall form an acceptance that an order concerning a specific product $P$ ordered by a customer $C$ is pending, when it is known that the customer wants product $P$ and that all sellers believe that the product order is a pending one. Fig. 6 shows a graphical representation of that recipe. We assume that every agent that undertakes the role “customer” in the company is able to determine (has appropriate r-actions) whether it wants a product. Furthermore, every agent that undertakes the role “seller” in the company is able to recognize the orders that are pending.

Using the above recipe, a group that plays the role “customer department”, e.g. “C-Dept”, knows that to accept an instance of the state $\text{pending}(P, C)$ it must gather information (beliefs) from at least one customer and from all the sellers. This is a policy.
for any group that plays the role “customer department” for the acceptance of the state $pending(P, C)$. Indicators are important here because they specify the number of agents of each atomic role that must contribute information, in order for the group members to accept an instance of the recipe state.

Since recipes have only one-level constituents, in order to define (complete) policies of groups that are high in the role containment hierarchy to be specified, we need more than one r-recipe. Let us clarify this with an example: To specify whether a state shall be accepted by all company members of our example organization specified in Fig. 3 using an r-recipe, we must specify what pieces of information subsidiary roles of the “company” role must contribute. This is achieved by creating an r-recipe with relevant role “company” and associating (in the body of that r-recipe) subsidiary states with roles that the role “company” contains. Then, for each such composite role new r-recipes are needed for the specification of how subsidiary states can be accepted by the members of the relevant subgroups of the company. This must be done recursively for the subgroups, until states that individuals in the company can recognize in a direct manner (i.e. via r-actions) are reached.

Generally, for a complete group policy towards the acceptance of a state, agents need a number of r-recipes that form one or more r-trees. R-trees are constructed gradually by combining r-recipes towards the recognition of states. Specifically, for the construction of an r-tree using an r-recipe $rec$, each subsidiary state $S$ of an element $\rho_{ind}: S$ of $rec$ must be unified either with the recipe state of an r-recipe of the role $\rho$, if $\rho$ is composite role, or, in case $\rho$ is an atomic role, with the state of an r-action of $\rho$. In case $\rho$ is a composite role, this process continues until states that can be directly recognized by individuals playing atomic roles (through their r-actions) are reached. Therefore, leaf nodes of r-trees are states that correspond to atomic roles.

According to the above, we can state that r-recipes specify the individuals in a group that are authorized to disseminate beliefs/information and specify how the combination of these beliefs yields to acceptances for the members of the group. Fig. 7 shows two r-recipes that define a policy for the role "company": The policy concerns the acceptance of instances of the state $pending-order(P, C)$.

Although r-recipes are simple in structure, they have enough expressive power to specify several real-life situations. Also, the expressive power of r-recipes can be increased if indicators specify a percentage of group members, or constraints that individuals/groups of recipe’s internal roles must satisfy. In this case, we can specify for instance that the department must accept that a product needs re-design if an engineer with extensive
experience in designing such products believes so.

Fig. 9 shows four policies (r-trees) that the members of a group that plays the role “company” may use to accept a state $S$. These policies result from the combination of the r-recipes in Fig. 8. Specifically, Fig. 9(a) specifies that the company shall accept $S|\sigma$ to hold, when (starting from the bottom) each engineer in each engineering department of the company believes $S_2|\sigma$. Assuming that the states in Fig. 9(b), (c) and (d) are ground (i.e. they have no variables), Fig. 9 (b) specifies that the company members accept $S$, when at least one engineer from each engineering department believes $S_3$. Fig. 9 (c) specifies that the company members accept $S$, when one engineer from one department believes $S_5$. Finally Fig. 9 (d) specifies that the company members accept $S$, when all engineers from a department believe $S_6$. 

Figure 7: Two r-recipes for a policy of the company for a state pending-order($P, C$).

Figure 8: Combinations of r-recipes for the acceptance of a state $S$. 

<table>
<thead>
<tr>
<th>Relevant Role</th>
<th>State-recognition recipes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company</td>
<td>$S$</td>
</tr>
<tr>
<td></td>
<td>engineering department$_{all}$: $S_1$</td>
</tr>
<tr>
<td>Engineering department</td>
<td>$S_1$</td>
</tr>
<tr>
<td></td>
<td>engineer$_{all}$: $S_2$</td>
</tr>
<tr>
<td></td>
<td>engineer$_{one}$: $S_4$</td>
</tr>
<tr>
<td></td>
<td>engineer$_{one}$: $S_6$</td>
</tr>
<tr>
<td></td>
<td>engineer$_{all}$: $S_5$</td>
</tr>
</tbody>
</table>
5.2 Forming acceptances

Each agent that is a member of a specific group with a collaborative responsibility \( c \Rightarrow g \) must exploit r-recipes to check whether it can contribute towards the acceptance of an instance \( c|_{\sigma} \) of the state \( c \). The acceptance of a state \( c|_{\sigma} \) requires communication between group members and knowledge about the position of each agent in the organization.

In our example, let us consider the state template \( \text{pending-order}(P,C) \), where \( C \) and \( P \) are variables denoting a customer and a product, that must be checked by the members of the group “The Company”. Towards the acceptance of one or more instances of this state, and according to the r-recipes specified in Fig. 7, each seller and one customer must “contribute” to the acceptance of such a state. Based on the contributions of customers and sellers, one or more instances of the state \( \text{pending-order}(P,C) \) may be accepted by the agents in “The Company”.

The following paragraphs formalize the generic notion of the “contribution” and specify how acceptances are being formed based on the contributions of individual agents.

Generally, considering an r-tree for a state \( S_{k+1} \), a path from the root \( S_{k+1} \) to a leaf state \( S_1 \) is a potential contribution for an agent that plays the atomic role that corresponds to this leaf state. A potential contribution is represented as a list

\[
(S_1, \langle \text{rec}_1, \text{ind}_1 \rangle, S_2, \ldots, S_k, \langle \text{rec}_k, \text{ind}_k \rangle, S_{k+1})
\]

where, each recipe \( \text{rec}_i \) has recipe state \( S_{i+1} \) and each recipe element \( \text{ind}_i \) specifies the set of agents (playing the role \( \rho^i \)) that must accept/believe \( S_i \), \( i = 1, 2, \ldots, k+1 \). It must be noticed that, since \( S_1 \) is a leaf node of an r-tree, \( \rho^1 \) is an atomic role and individuals that play this role can believe an instance of \( S_1 \) only if they recognize such a state via an r-action. Fig. 10 shows a potential contribution in an r-tree.

For example, from the r-recipes in Fig. 7 we can identify two potential contributions, one for the sellers and one for the customers that are the following:

For the sellers,

\[
(\text{pending}(P,C), \langle \text{rec}_2, \text{seller}_{all} \rangle, \text{pending}(P,C), \langle \text{rec}_1, \text{customer-department}_{one} \rangle, \text{pending-order}(P,C))
\]
and for the customers,

$$(\text{want–product}(P,C), \langle \text{rec}_2, \text{customer\_one} \rangle, \text{pending}(P,C), \\
\langle \text{rec}_1, \text{customer–department\_one}, \langle \text{pending–order}(P,C) \rangle)$$

Agents, acting as group members, must identify their potential contributions towards states depending on the roles they play and according to the r-recipes of these roles. Concerning the contribution shown in Fig. 10, when an agent plays $\rho^1$ and it forms a belief $S_1|_\sigma$, then the contribution $(S_1, \langle \text{rec}_1, \rho^{\text{ind}_1}, \langle \ldots, S_2, \langle \ldots, \langle \text{rec}_k, \rho^{\text{ind}_k}, S_k+1 \rangle, \ldots \rangle \rangle, \ldots \rangle, \ldots )|_\sigma$, which results from the application of $\sigma$ to a potential contribution, is a personal contribution of this agent. Instances of $S_1$ are recognized by means of r-actions.

Having clarified the notions of policy and contribution, we can specify how the acceptance of an instance of a state $S$ for a group $G$ is being actually computed. Each agent in $G$, based on the r-actions it holds, forms individual beliefs about ground states. Based on these beliefs and the r-recipes it knows, it computes all its personal contributions for the acceptance of an instance of $S$ considering only those elements of r-recipes that concern roles played by groups in which the agent participates. For example, when a seller searches for contributions for the state order–prepared$(P,C)$ using the recipe specified in Fig. 7, it shall consider only the element that corresponds to the role “customer department”, since sellers in our example organization participate in groups that play this role. We must notice that since all agents belong to the same group, and since all members of a group share the recipes of the role that the group plays, each individual agent knows an r-recipe with a recipe state that matches $S$. However, each agent has zero or more contributions to the acceptance of an instance of $S$.

Each agent identifies its own personal contributions without constructing the whole r-tree and without considering the contributions of the others. However, in a group $G$, agents that play the same role and have the same beliefs have identical personal contributions towards a state. The personal contributions of a set of agents that are identical constitute a contribution set. According to r-recipes’ specifications, there must be a sufficient number of contributions in a contribution set in order to affect the acceptances of the group. This is determined by the indicators of each r-recipe element in the contribution and the structure of the organization. For example, according to the
recipe shown in Fig. 7, the contribution set of sellers must include personal contributions from each seller in order for the company members to form an acceptance for the state \( \text{pending-order}(P,C) \). A contribution set that contains a sufficient number of personal contributions—according to recipes’ specifications—is called a group contribution.

An r-tree specifies the necessity of several potential contributions from different sites. Therefore, a single group contribution may not be enough for the members of a group to form an acceptance. For instance, according to the r-recipe specified in Fig. 7 the contributions of company sellers and customers are required: Sellers and customers are not the only members of “The Company”. However, according to this recipe, the agents that play these roles are necessary and sufficient to determine whether the state \( \text{pending-order}(P,C) \) holds for this company. Specifically, there must be a group contribution from all the sellers and a contribution from at least one customer. An acceptance for this state is formed via communication between the sellers and the customers. The other members of the company must be informed about any acceptance formed, but they need not be bothered for the formation of such an acceptance.

As far as communication is concerned, prima facie, the best solution to the problem of the formation of acceptances will be that who would minimize communication messages. This would require the introduction of a group facilitator agent that will receive contributions, compute group acceptances and disseminate these to the group members. However this is a centralized solution that reduces systems’ robustness. An alternative would be to delegate the task of recognizing and disseminating acceptances to specific and dynamically determined members of the organization. A totally decentralized solution is the one in which no group agent has a special role for the recognition of acceptances. Group agents send their contributions to all group members and each one computes the acceptances individually, according to the known group policies. In this latter case, although we must tolerate the communication burden, the system is more robust for open settings where agents join and leave the organization unexpectedly. In this paper we assume the totally decentralized setting described above.

5.3 Example: Computing acceptances

This section gives an example that illustrates how acceptances are formed within the organization shown in Fig. 11. Let us consider that company members need to form acceptances concerning the state template \( \text{pending-order}(P,C) \) and have the r-recipes shown in Fig 7. The combination of these recipes results in the group policy depicted in Fig. 11. Gray scales in Fig. 11 show the set of positions that share the same personal contributions (region 1), the set of positions that share the same group contributions (region 2) and the set of positions that share the same acceptances (region 3).

Table 1 shows the information that each agent has and communicates to the others in the company for accepting an instance of the state \( \text{pending-order}(P,C) \).

The first column of Table 1 shows the positions within the organization: “seller\(_1\)”, “seller\(_2\)”, “customer” as well as the other positions in the company: engineers and manufacturers. Engineers and manufacturers, according to the policy, do not participate in the formation of the respective acceptance.
Figure 11: The communication “regions” of (1) personal contributions of sellers, (2) group contributions and (3) acceptances.

Personal contributions:
\[ C_1 = (\text{want-product}(p_1,c_1), \langle \text{rec}_2, \text{customer}_{dept} \rangle, \text{pending}(p_1,c_1), \langle \text{rec}_1, \text{customer-department}_{dept} \rangle, \text{order-prepared}(p_1,c_1)) \]
\[ C_2 = (\text{pending}(p_1,c_1), \langle \text{rec}_2, \text{seller}_{all} \rangle, \text{pending}(p_1,c_1), \langle \text{rec}_2, \text{customer-department}_{dept} \rangle, \text{order-prepared}(p_1,c_1)) \]
\[ C_3 = (\text{pending}(p_2,c_2), \langle \text{rec}_2, \text{seller}_{all} \rangle, \text{pending}(p_2,c_2), \langle \text{rec}_1, \text{customer-department}_{dept} \rangle, \text{order-prepared}(p_2,c_2)) \]
\[ C_4 = (\text{pending}(p_3,c_3), \langle \text{rec}_2, \text{seller}_{all} \rangle, \text{pending}(p_3,c_3), \langle \text{rec}_1, \text{customer-department}_{dept} \rangle, \text{order-prepared}(p_3,c_3)) \]

Table 1: A snapshot of the information distribution in “The Company” group during the formation of acceptances.
The second column of Table 1 shows the beliefs of each agent in a position. These, according to the policy shown in Fig. 11, are the beliefs that are relevant to the formation of agents’ personal contributions towards the state $\text{order-prepared}(P,C)$. The third column shows the personal contributions of each agent and how these contributions are shared between agents that play the same role. Contributions typed in bold face show “own” contributions of the corresponding agent, while those typed in normal face show contributions sent by other agents that play the same role. Each contribution is being marked with the agent that has constructed it. So, for example, “seller$_1$” has sent its personal contributions to “seller$_2$” and vice versa. Shaded regions in Table 1 correspond to the shaded regions in the organizational structure of Fig. 11, indicating the set of agents that communicate during each acceptance-formation stage.

Each agent, having a set of personal contributions, forms contribution sets (i.e. sets of identical contributions) annotated with the set of agents that have sent them. These are shown in the fourth column of Table 1. During the formation of the contribution sets agents do not need to communicate, since all agents have all the necessary information to form these sets. Agents that have formed contribution sets must check whether contributions in these sets are sufficient for the formation of group contributions, i.e. they must check which contribution sets include contributions from a sufficient number of positions according to the r-recipe indicators. Group contributions are communicated between the agents that share the same policy. For the policy of Fig. 11 this is the set of sellers and customers. The set of group contributions is shown in the fifth column of the table. Similarly to personal contributions, group contributions in boldface are own contributions formed by each agent, by exploiting the contributions sets. Group contributions that have been sent to an agent are shown in normal face font.

Finally, having the set of group contributions, each agent can form acceptances individually. These acceptances are then sent to the agents in the group “The Company”. The last column of Table 1 shows the acceptances formed by each agent in the company. Again, acceptances in bold face are those that have been formed by each agent, while those that have been sent by other agents are shown in normal font face.

The maintenance of an acceptance in a dynamic environment requires regular checking and validation of the contributing acceptances and beliefs. For example, an agent may change its beliefs. This may result in changing the set of its personal contributions that, in its turn, can affect the group contributions and the acceptances of the group. Therefore we require that each agent shall keep the other agents informed about changes in its personal contributions, group contributions and acceptances. I.e. an agent that adds or retracts a personal contribution from the set of its personal contributions communicates this change to the agents that play the same role. Similarly, when an agent adds or retracts a group contribution from the set of its personal contributions, it communicates this change to the agents that follow the same policy. Finally, when an agent adds or retracts an acceptance it must inform the other agents in the group.
6 Collaborative achievement of goal states

When the members of a group accept an instance c|g of the condition of a collaborative responsibility c ⇒ g, then they form a commonly agreed goal for seeing to it that the corresponding instance g|σ of the responsibility goal state holds. The group must pursue this goal collaboratively. For each commonly agreed goal, the members of the group must check the consistency of that goal with the pre-existing goals they possess, they must agree on a specific state-achievement recipe (a-recipe) and finally they must allocate the subsidiary responsibilities of the selected recipe. A-recipes represent the organization's common practices. They specify how a group achieves goal states by distributing responsibilities to group members or subgroups. This section introduces a-recipes and presents the mechanisms for recipe selection and responsibility allocation, based on the constructs of the organizational model.

6.1 A-recipes

As Fig. 12 shows, the structure of an a-recipe is similar to that of an r-recipe. However, there are two important differences. The first concerns the presence of a condition, which is a state. When this condition is accepted by the group members, then they consider the a-recipe to be applicable. For r-recipes there is no need for such a condition because agents shall use any available r-recipe to recognize states. The second difference concerns the body of the recipe, where roles are associated with responsibilities towards states. These responsibilities can be collaborative, hybrid or individual responsibilities. Atomic roles can have only individual responsibilities, while a group can have any of the three types of responsibilities. However, for presentation reasons and without losing the generality, we assume that composite roles in an a-recipe are associated only with collaborative responsibilities.

Therefore, an a-recipe can be represented as a tuple ⟨α, θ, B⟩, where α is the state that the recipe achieves, θ is a condition for the recipe to be applicable, and B is the body of the recipe. The body of an a-recipe consists of elements of the form ρ inds : R, where ρ is an internal role, inds is an indicator as in r-recipes, and R is a set of collaborative responsibilities assigned to the internal role ρ. The goal state of each responsibility in an a-recipe is considered to be a subsidiary state of the recipe state α.

The following is an example of an a-recipe for the state m&s(P, C):

⟨m&s(P, C), true, \{engineering-department one : specs(P, C, S) ⇒ mready(P, S), manufacturing-department one : mready(P) ⇒ m(P), customer-department one : mp(P, C) ⇒ s(P, C)\}⟩

This recipe belongs to the role “company” and states that in order for the company to manufacture and ship a product P ordered by a customer C, i.e. to achieve the state m&s(P, C), each department must undertake one responsibility. More precisely, the responsibility of the engineering department states that in case the specifications S for the product P by the customer C are known (i.e. the state specs(P, C, S) holds) then the department must see to it that the manufacturing instructions exist (mready(P, S)).
Figure 12: The constituents of state-achievement recipes.

Figure 13: An example of an a-recipe.

The responsibility of the manufacturing department states that if the manufacturing instructions for a product \( P \) exist, then the group must manufacture this product (\( m(P) \)). Finally, the responsibility of the customer department states that if a product has been manufactured for a specific customer \( C \) (\( mp(P,C) \)), then it must be shipped (\( s(P,C) \)). These responsibilities are collaborative responsibilities of the corresponding departments. Fig. 13 shows a graphical representation of this recipe.

Using state achievement recipes, a group distributes responsibilities to subgroups and individuals in a task-dependent way. Depending on the commonly agreed goals, the recipe selection mechanism and the subgroup/individual selection mechanism adopted, responsibilities are dynamically distributed to groups or individuals.

Fig. 14 shows four possible cases for the company to achieve a goal state \( g \). These cases result from the combination of eight a-recipes. The recipe for achieving the commonly agreed goal \( g \) shown in Fig. 14(a) specifies that every engineering department must undertake the responsibility \( c_1 \Rightarrow g_1 \). Doing so, if the members of an engineering department accept that an instance \( c_1|_\sigma \) of \( c_1 \) holds then the group proceeds to achieve the commonly agreed goal state \( g_1|_\sigma \) of the instantiated responsibility. Towards this commonly agreed goal, all engineers of the department shall undertake the responsibility \( c_2|_\sigma \Rightarrow g_2|_\sigma \) which results from the responsibility \( c_2 \Rightarrow g_2 \) after the application of \( \sigma \). In Fig. 14(b), (c) and (d), states are considered to be ground. Fig. 14(b) specifies that to achieve \( g \), the company must select a department for fulfilling the responsibility \( c_3 \Rightarrow g_3 \). If the department succeeds to form a commonly agreed goal \( g_3 \), then it must select one engineer to adopt \( c_4 \Rightarrow g_4 \). In the case (c) all departments must adopt \( c_5 \Rightarrow g_5 \). In case a
6.2 Achieving goals collaboratively

For each goal state that shall be achieved collaboratively, group members pass through the stages of group formation, recipe selection and responsibility allocation. We explain each of these stages in the following paragraphs.

6.2.1 Group formation

Group members that form an acceptance on the condition $c$ of a collaborative responsibility $c \Rightarrow g$, they also form a commonly agreed goal $g$. In case $g$ is not in conflict with other goals that an agent has, it is assumed that the agent shall consider contributing to the achievement of the commonly agreed goal. This is conjectured because in organized settings agents must be active towards the achievement of group goals and this participation can be a measured by the quality of role enactment. Although monitoring and measuring agents’ participation in group activities is an important issue, it lies out of the scope of this paper.

For agents to start acting in a coordinated manner, they must identify the need to achieve the commonly agreed goal $g$. Using r-recipes agents must accept that the state $g$ is not unattainable and that it has not been established. If each agent forms such an acceptance, then it will individually proceed to the following two stages.

6.2.2 Recipe selection

During this stage agents shall try to select a recipe for the commonly agreed goal. Before the selection, agents must identify the applicable recipes i.e. the recipes for which the department forms the commonly agreed goal $g_5$, then it must select an engineer to adopt $c_6 \Rightarrow g_6$. Finally, in (d) the company must select a department, and if that department forms a commonly agreed goal $g_7$, then all engineers of the department adopt $c_8 \Rightarrow g_8$.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
\textbf{Relevant Role} & \textbf{State-achievement recipes} \\
\hline
Company (a) & $c_1 \Rightarrow g_1$ & $c_2 \Rightarrow g_2$ & $c_3 \Rightarrow g_3$ & $c_4 \Rightarrow g_4$ \\
\hline
Engineering department (b) & $c_1 \Rightarrow g_1$ & $c_4 \Rightarrow g_4$ & $c_6 \Rightarrow g_6$ & $c_7 \Rightarrow g_7$ \\
\hline
\end{tabular}
\end{table}

Figure 14: Several state achievement recipes for the stage $g$. 

\[ g \]
\[ \text{engineering department}_{all}: \quad c_1 \Rightarrow g_1 \]
\[ \text{engineering department}_{one}: \quad c_1 \Rightarrow g_1 \]
\[ \text{engineering department}_{all}: \quad c_1 \Rightarrow g_1 \]
\[ \text{engineering department}_{one}: \quad c_1 \Rightarrow g_1 \]

\[ g \]
\[ \text{engineering department}_{all}: \quad c_1 \Rightarrow g_1 \]
\[ \text{engineering department}_{one}: \quad c_1 \Rightarrow g_1 \]
\[ \text{engineering department}_{all}: \quad c_1 \Rightarrow g_1 \]
\[ \text{engineering department}_{one}: \quad c_1 \Rightarrow g_1 \]
applicability condition has been accepted by the members of the group. It must be emphasized that the group members must accept the recipe’s applicability condition for the recipe to be applicable. Currently, the selection of a recipe from the set of applicable recipes is done by each agent individually and it is based on specific preferences that reflect the organization’s preferences. These are specified in advance and are common to group members.

6.2.3 Responsibility allocation

A group that has selected an a-recipe shall allocate the responsibilities specified in the recipe to the corresponding roles. When the indicator of an element $\rho_{\text{ind}} : R$ in the body of a recipe has the value “all”, then there is no need for a group selection mechanism for $\rho$, since all agents/groups that play the internal role must undertake the specified responsibilities. If the indicator has the value “one”, then the group must select the “appropriate” group or agent, depending on whether $\rho$ is composite or atomic. Currently, the selection of the groups/agents that undertake the responsibilities in an a-recipe is done similarly to the recipe-selection mechanism and it is based on organization preferences. Future work concerns forming policies towards accepting specific role-playing groups or individuals for undertaking responsibilities.

6.3 Example: Achieving goal states collaboratively.

In order for agents to record the responsibilities and agreed goals of the groups in which they participate as well as their actions towards the fulfillment of a responsibility, they construct an intentional context. In case an agent participates in multiple groups with multiple responsibilities, then it has multiple intentional contexts—one for each responsibility.

The intentional context of an agent binds the groups in which the agent participates and motivates activities towards goal states in accordance to the responsibilities of the group(s) in which the agent participates. Fig. 15 shows the intentional context of an agent for fulfilling a collaborative responsibility of “The Company”. As already said, it is assumed that r-recipes contain only collaborative responsibilities.

An intentional context is constructed as follows: Given a collaborative responsibility of a group, the agent constructs a node for this responsibility template in its intentional context. This node is assigned to the corresponding group (the relation between agent’s intentional context and groups in which the agent participates is shown below). For every accepted instance of the responsibility, a goal is generated that is added in the context below the responsibility. Depending on whether the responsibility is a collaborative or a hybrid one, each group member assigns the corresponding goal state to the group (in this case we say that we have a commonly agreed goal) or to itself.

For each commonly agreed goal the group must select an applicable a-recipe and assign responsibilities to subgroups or individuals, depending on whether the internal roles are composite or atomic. When a subgroup is assigned a collaborative or a hybrid responsibility, then the above mentioned process is performed for this subgroup. When
a subgroup is assigned an individual responsibility, then each group member, using r-actions and a-actions, shall try to fulfill this responsibility. In this way, a tree-like structure is constructed incrementally.

As Fig. 15 shows, the intentional context is divided into several levels. Each level corresponds to groups in which the agent participates. The last level corresponds to the individual agent. Although members of a group have different intentional contexts, they are expected to share a common view of the group activity with respect to a responsibility.

An agent records in its intentional context only those aspects that affect its activity towards the fulfillment of a responsibility. Therefore, given a responsibility of the company, company members recognize several instances of the responsibility condition. For each instance of the condition, as it is shown in Fig. 15, there is a commonly agreed goal state to the achievement of which some of the group members will commit. These goal states form the children of the corresponding responsibility in the agent’s intentional context. For each of the goal states a subgroup of committed agents selects a commonly agreed recipe, selects the “appropriate” subgroups/individuals and distributes responsibilities to them. These responsibilities form the children of the corresponding goal state in agent’s intentional context. In case the agent has been assigned an individual responsibility, or in case it participates in a group that has been assigned a group responsibility, it continues to record the subsidiary responsibilities/activities towards the fulfillment of this responsibility in the next level of the intentional context.

As can be understood from the above, each member of a group holds an intentional context that records part of the group activity towards the fulfillment of a responsibility. More precisely, each group member is aware of the responsibilities and the goals of
each group it belongs, of the a-recipes that the group has selected for each goal state and of the assignment of responsibilities to groups or individuals. For example, the representative of Mike, representing an engineer, is aware of the responsibilities and goals of the engineering department “E-dept”, is aware of the recipes that the “E-dept” has selected for each agreed goal, and of the specific subgroups or individuals that have been assigned responsibilities towards achieving these goals.

When a group or individual has been assigned a responsibility, then each of it’s members is responsible to recognize the instances of this responsibility, the goal states that emerge from this responsibility, and therefore it is responsible to expand the intentional context towards the fulfilment of this responsibility. For example, although the representative of Mike knows the responsibilities of other representatives in the engineering department, it may not know what these representatives are actually doing, i.e. it may not know the goal states that the representatives of other engineers have agreed upon.

7 Communication requirements

As it can be understood from the above, the task that mostly requires communication is the formation of group acceptances: This is necessary for recognizing the need for collaboration, for selecting a-recipes towards the achievement of goal states, and for allocating subsidiary responsibilities to individuals and sub-groups.

To study the communication requirements for the formation of acceptances in a totally distributed setting, let us assume that a group has \( n \) agents and comprises \( m \) atomic roles. Also, let us assume that each role has \( k \) distinct atomic positions and that each agent occupies one position. Therefore \( n = k \cdot m \).

Given a policy with a required contribution from each atomic role (this is the most demanding case where all agents participate in deciding group acceptances) we can distinguish between two extreme cases: (a) The policy requires the contribution of all the agents that play the corresponding m atomic roles (i.e. indicator of each r-recipe element is “all”), and (b) The policy requires the contribution of one of the agents that play the corresponding atomic roles (i.e. the indicator of each r-recipe element is “one”).

In the first case, each agent must send its personal contribution to each of the \( k - 1 \) agents that play the same role. Since the policy requires the contribution of all agents, it requires \( n(k - 1) \) messages. Then, each group contribution identified by each agent must be sent to the rest \( n - 1 \) members of the group. In the worst case this requires \( n(n - 1) \) messages. After communicating their group contributions, agents will identify an acceptance in a distributed way. This acceptance must be communicated to the members of the group. In the worst case all agents communicate the recognized acceptance before they receive any messages. This yields to \( n(n - 1) \) messages. So, there is a total of \( n(k - 1) + 2n(n - 1) \) messages.

In the best case, only one agent from each of the \( m \) roles will communicate the group contribution to the other agents (requiring \( m(n - 1) \) messages) and one agent will communicate the acceptance to the others (requiring \( n - 1 \) messages). So, in the best case the formation of an acceptance requires \((k + m)n - m - 1\) messages. Since the product
$k \cdot m = n$ is constant, the quantity $k + m$ is minimized when $k = m = \sqrt{n}$. Therefore, the minimum total number of messages is of magnitude of $n\sqrt{n}$.

In the second case, agents do not need to communicate their personal contributions since each personal contribution is a group contribution. In this case, the worst case scenario requires $2n(n - 1)$ messages while the best case scenario requires $2(n - 1)$ messages for the formation of an acceptance.

To achieve the best-case in both of the above cases, agents may delay sending the recognition of group contributions, waiting for someone else to send it. The delay time is usually specified to be proportional to agents’ priority. In this case, we can achieve lower communication overhead, although we cannot guarantee that we will always achieve the number of messages encountered in the best-cases.

Counting the number of messages required, we have assumed a totally distributed setting: There is not a specific agent (e.g. a special seller or the manager of the selling department) that gathers all personal/group contributions, that decides and communicates the formed acceptances. Given such an agent exists, the required messages in the first case (where the contributions of all agents are needed), for the worst and the best scenario, drop to $2(n - 1)$. In the second case (where only one agent is needed), the worst scenario (which results for $m = n$) requires $2(n - 1)$ messages to be exchanged, while the best scenario (that results for $m = 1$) requires $n - 1$ messages.

8 Related work

The development of tools and systems that support humans to work together effectively has found much attention in the last few years: Workflow management systems (WFMS) as well as agent-based systems for supporting humans to accomplish common tasks are emerging paradigms. WFMS is a relatively recent technology[29]. WFMS research prototypes include METEOR[30], MOBILE[31], ADEPT[32], and EXOTICA[33]; while commercial products include IBM MQSeries Workflow, Staffware, TIBCO InConcert, and COSA Workflow. Such systems aim to coordinate the work of humans and not to actively help them during their collaboration. The research reported in this paper complements efforts for the development of WFMS, as in a greater extent than these systems it aims to cover collaborative activity in settings where flexibility to the fulfillment of responsibilities is important and where information is inherently distributed among the members of a group. Agents in our approach recognize the need for enacting collaborative processes in a totally distributed way, agree on the appropriate recipe to be used, and allocate responsibilities for the achievement of commonly accepted goal states dynamically. Doing so, they incrementally build a shared context that is distributed among individuals.

There are several multi-agent systems that aim to support humans on specific collaborative tasks[34, 35, 36, 37]. Unlike WFMS, multi-agent systems typically support ill-structured tasks, in which steps to be followed, coordination and information flow cannot be totally specified in advance. Such tasks include meeting scheduling[34] and visitor hosting[35, 36]. E-ELVES[37] is a sophisticated multi-agent system that integrates a range of technologies that can support a variety of tasks within a human organization:
scheduling meetings, arranging lunch, and locating other people. In E-ELVES each hu-
man owns a personal agent that interacts with its user and communicates with the other
agents. Agents in E-ELVES coordinate using TEAMCORE[38], a domain independent, de-
centralized, teamwork-based integration architecture. Although in E-ELVES agents are
organized using a role aggregation hierarchy like the one defined in our model, this or-
ganization is not exploited for the formation of acceptances. Furthermore, although E-ELVES agents collaborate to each other, the aim of the system is to support human
coordination and not collaborative problem solving activity.

Agent-based solutions have also been used to support humans to participate effectively
in network communities. Network communities are unstructured, either with an informal
or with no organizational structure and with relationships in networks often hidden or
unknown. Although collaborative activity in organized settings is quite different from
providing support for humans to participate in network communities, it is worthy to
mention the work of Hattori et al[39]. Their system supports humans via (a) linking them
with others and with communities that share similar interests, (b) supporting smooth
communication by providing support for visualizing and sharing common contexts, as
well as, for identifying the flow of conversations/discussions, and (c) finding relationships
between people including objectives/roles of communities and individuals. Hattori et
al[39] propose a solution that is similar to the approach proposed in this paper, to the
extent that it is based on the active support of humans using agent technology, on the
interaction between software agents and on the interaction between human and software
agents. More specifically, they associate to each human one or more personal agents
and they assume that all of these agents cooperate and act as a unit, with the user
being the central figure. These agents can be either domain specific (for example, an
information retrieval agent specialized for financial news) or more generic (for example,
an interface agent for navigating and reading documents). A human can participate in
multiple communities; in this case she may have multiple domain-specific agents in each
community. This approach, although it has several similarities to our approach, does
not deal with supporting collaborative activity in organized settings.

In order to build flexible collaborative systems and not just collaboration tools[17] we
must take into account a general model of collaboration. Collaboration is a complex type
of behavior and several general theories/models have been introduced for its specifica-
tion. The most dominant are the SHARED-PLANS model[17] and the JOINT-INTENTIONS
theory[4, 40]. These are general enough and provide some form of requirements or
guidelines for building collaborative agents. However, for the realization of collaborative
agents several important aspects of collaborative activity must be specified.

The stages of collaborative activity in well-organized settings proposed in this paper
strictly resemble the stages of the models of Wooldridge and Jennings[6] and Panzarasa
el al[5]; however, these have been adapted for agents acting in well-organized settings.
The model proposed by Wooldridge and Jennings[6] lacks any social aspect during col-
aboration. The model proposed by Panzarasa et al[5] captures the social processes in
each stage of collaboration based on roles and a social mental shaping process. Social
mental shaping refers to the phenomenon that the social nature of agents affects their
mental states, and thereby motivates their behavior.
In contrast to the above mentioned approaches, we propose r-recipes for forming acceptances and commonly agreed goal states in contrast to the notions of collective belief and joint/shared goal. This makes the representation of group policies explicit, affecting (in conjunction to the exploitation of the organizational structure) all stages of collaborative activity: The need for collaboration is recognized by the members of a group by accepting the condition state of an explicitly stated collaborative responsibility. This is a totally distributed method for recognizing the need for collaboration, in contrast to the other models where this need is identified by a single agent. The group of collaborators is formed by means of the agents that play specific organizational roles: These agents have accepted the responsibility condition and have checked that the commonly agreed goal state is consistent with the other goals they have. This is rather straightforward, as there is no need for identifying the agents that shall collaborate via a from-a-first-principles approach: The search space for the possible collaborators is limited significantly. Currently, recipes selection is rather straightforward by means of two assumptions: All agents in a group share the same recipes for achieving goals (these are the organizational practices) and share the preferences for these recipes. Then, the group in a totally distributed way can decide on the applicability of recipes by accepting their conditions. Future work concerns relaxing the assumption about recipes’ preferences. Finally, responsibility allocation is been done by exploiting a-recipes’ specifications on the subgroups or individuals that shall undertake specific responsibilities. Currently, the assumption during this stage is that all agents share preferences as to which subgroup or agent to select. It must be noticed that preferences on recipes as well as on subgroups and individuals can be formed by the group through a learning process based on success and failure stories, reliability, etc. However, the use of a-recipes enables the group to decide which are the subgroups or individuals that shall undertake the corresponding responsibilities. This gives to the group the opportunity to self-manage, at least partially, responsibility allocation within its members.

There are several efforts and approaches for building collaborative systems of agents (or teams of agents). Closer to our approach are systems like GRATE*[41], STEAM[42], RETSINA-MAS[43], and CAST[44], in which collaboration is based on the above mentioned general models of collaborative activity. Although STEAM, RETSINA-MAS, and CAST use the role concept in modeling teamwork, none of them offers an adequate organizational model for representing human organizations. Roles in these systems are not interconnected to form a concrete role model and are not used to build complex organization units. In this paper we specify complex organizational structures using composite roles and we provide a set of modeling constructs for building organizational models. Also, none of the above systems (including GRATE*) makes a distinction between beliefs and acceptances as we do here. This makes difficult, if not impossible, for these systems to consider several real-life situations where individuals accept certain states based on certain policies of the group in which they participate, or due to the policies that an organization imposes to its members.
9 Concluding remarks

Collaborative activity between agents is complex group behavior that requires sharing of knowledge and communication. Although there are generic models that concern several aspects of collaborative activity, building systems with generic mechanisms for collaborative problem-solving and action is quite complicated. In this paper we focus mainly on collaborative activity in well-organized, distributed and dynamic settings. In such settings, agents must not only coordinate their activities for the achievement of their shared goals, but must be able to (a) establish common views of the world, based on organization practices, (b) distribute responsibilities dynamically among members of the organization based on specific needs that arise during the course of action and (c) share a context of action that binds the members of groups together towards the achievement of their goals.

The objective of this research is to build systems that empower humans to reason about their responsibilities, create common awareness for every agent about important states of the world, form agreed goals, and achieve goal states. This paper, in a greater extent than existing approaches on collaborative activity, emphasizes on the exploitation of organizational structures. It describes how an explicitly stated and shared organization structure can enhance collaborative activity in a distributed and dynamic setting via the specification of role responsibilities and recipes for the recognition of the need for collaboration and for the achievement of goal states.

The paper proposes a specific set of constructs for building organizational models and organizational structures which is by no means complete. There are many aspects of a real-world organization that can not be modeled using the constructs proposed. For instance, different types of relations between roles, resources available to groups and agents, benefits and utilities of agents are not taken into account. The emphasis in this paper is on the role construct, roles’ responsibilities and policies of groups for pursuing their agreed goals. The enrichment of the organization model towards building systems that support more complex human organizations is an ongoing research.

Future work concerns increasing the expressivity of the language used for specifying recipes as well: This includes the addition of temporal constraints for the fulfillment of responsibilities towards subsidiary states in recipes, as well as the extension of the set of indicators. R-recipes may include indicators expressing a percentage of group members, their majority or minority, or even members with specific expertise/capabilities. A-recipes may also include indicators expressing members with specific capabilities and/or expertise to whom the task will be allocated.

We have implemented a prototype system in SWI-Prolog[45] for building organizations of collaborative agents. Our prototype system comprises all the required mechanisms for elaborating individual, hybrid, and collaborative responsibilities, as they are described in the previous sections of this paper. Agent communication is message-based and asynchronous. Our experimentation has been conducted in a single machine. However, using the JPL[46] java-prolog interface we have also managed to incorporate our agents into the JADE agent development framework[47]. In the current implementation we assume that the communication service is reliable and messages are delivered within
an acceptable time bound. Currently, group decision making mechanisms are based on knowledge that can be obtained by group members and is done individually by each group member. Further work concerns the implementation of different types of group decision-making mechanisms for the selection of recipes, groups and individuals, based on hybrid and collaborative responsibilities.

Acknowledgements

This work has been supported by GR Pythagoras grant number 1349 under the Operational Program for Education and Initial Training. We also thank the anonymous reviewers for their comments and suggestions: The helped us to improve and clarify the presentation of our work.

References


