Supporting Organizational Accountability inside Multiagent Systems

Matteo Baldoni[1], Cristina Baroglio[1], Katherine M. May[2],
Roberto Micalizio[1], and Stefano Tedeschi[2]

[1] Università degli Studi di Torino — Dipartimento di Informatica
firstname.lastname@unito.it
[2] Università degli Studi di Torino
firstname.lastname@edu.unito.it

Abstract. We present and analyze the problem of realizing an accountability-supporting system in multiagent systems technology. To this aim and to avoid ambiguities, we first characterize the concept of accountability, which we later work to realize computationally, particularly in relation to the partially overlapping notion of responsibility. Then, with the aim of achieving accountability as a design property, we provide a few principles that characterize an accountability-supporting multiagent system. We provide an accountability protocol to regulate the interaction between an agent, willing to play a role in an organization, and the organization itself. Finally, we show as a case study how the use of such a protocol allows multiagent systems, realized with JaCaMo, to support accountability.

Keywords: Computational Ethics, Accountability, Multiagent Systems, Sociotechnical Systems.

1 Introduction

In the field of multiagent systems (MAS), individual and organizational actions have social consequences, which require the development of tools to trace and evaluate principals' behaviors and to communicate good conduct. This concerns the value of accountability. The main contribution is to provide a notion of when an organization of agents supports accountability by investigating the process of construction of the organization itself as well as to give the definition of a protocol to be followed in order to design and build an organization that supports accountability. The core of the analysis is the notion of role and the action of role adoption (or enactment). Indeed, the concept of role has been studied in many different fields of computer science like software engineering, databases, security and, of course, artificial intelligence. Despite several proposals have been made about a characterization of such a concept, there is little consensus about its properties and purposes. A reason for this lack is the diversity of scopes and applications that such a general concept could have in different disciplines. If seen from an organizational point of view, roles should describe the way individuals can interact with and act upon an organization.

Continuing our work on accountability as presented in [3], in this study we wish to present a more nuanced view of the concept, particularly in relation to its close sibling, responsibility, and offer an initial exploration of platform requirements for a software-based, accountability-attribution process. Building on our previous characterizations of accountability, this study adds considerations that consequently impact implementation decisions on our chosen platform. We also attempt to tie our work more closely to work in the field of ethics so as not to diminish the rich meaning behind accountability and avoid cheapening the concept to the point of a definitional equivalence with traceability. We offer ethical dilemmas to force our hand into programming difficult decisions and form a more conceptually complete system of accountability. With some conceptual modifications, we believe JaCaMo [5] a particularly apt platform for building in an accountability mechanism.

2 A Characterization of Accountability and Responsibility

As a potentially wide-ranging concept, accountability can take on many different characteristics and meanings, especially depending on the discipline in which discussion takes place. Our own approach relies on a more mechanistic view in which accountability is a backward-looking, institutional process that permits one entity to be held to account by another. The accountability process activates when a negative (or positive) state is reached and an authoritative entity wishes to hold to account those behind said state. The process itself is divided into three primary phases: 1) an investigative entity (forum) receives all information regarding agents' actions, effects, permissions, obligations, etc. that led to the situation under scrutiny, 2) the forum contextualizes actions to understand their adequacy and legitimacy, and finally 3) the forum passes judgment on agents with sanctions or rewards [7]. Our goal consists in automating the entire process for use in a MAS, although we will presently leave out the sanctioning piece of the third phase due to its organization-specific application. In order to realize our goal, we begin by looking at the reasoning process behind accountability to identify points of adaptation into the software world.

Three conditions for accountability attribution are identified in [8]: agency, causal relevancy, and avoidance opportunity. The agency condition expresses not only an individual's independence and reasoning capacity, but also that entity's ability to distinguish right and wrong. Because software agents have no independent ethics, we maintain that their moral compass, so to speak, only emerges in and is unique to a given software organization. An agent's ”ethics” will be expressed through adherence to norms and commitments. The causal relevancy condition expresses the necessity of causation linking a given agent to a situation under scrutiny. The avoidance opportunity condition specifies that an “agent should have had a reasonable opportunity to have done otherwise” [8]. The condition does not imply that in absence of the agent's action, the situation would not have come to pass. That is, an agent can be held accountable for a causal contribution even in the presence of the inevitable. To illustrate, we can consider the example from [8] in which we find ourselves in the presence of two murderers who simultaneously shoot someone, their bullets piercing the victim's heart in the same moment. Should one of the murders have chosen not to shoot, the victim would still have died, yet we would naturally hold both accountable. They both had an opportunity to avoid causally contributing to the victim's death.

The accountability attribution process, due to its quality of determining the parties who causally contributed to a given situation, necessarily uses of a backward-looking approach. Its domain lies in the past, uncovering intrigue and revealing causal mystery. Responsibility attribution, as accountability's temporal mirror sibling, instead looks to the future and places its definitional weight on parties deemed capable and worthy of realizing a certain goal or goals. Both concepts, divided only by time, work towards the common purpose of situational deconstruction in order to distribute responsibility or accountability in a reality-reflecting fashion among a situation's atomic agents. Their common purpose gives rise to a strong working relationship between the concepts. Their complementary nature permits considerations and evaluations to flow between the two, which can, at the very least, inform their respective investigations. As expressed in [9], "When I am responsible for my actions now, I may be held accountable later."

Despite their close relationship, one concept cannot directly implicate the other. Even if a responsible entity's task was not completed as planned, that does not automatically indicate that fault lies with said entity. Likewise, if an entity is deemed accountable, that entity was not necessarily designated responsible in planning stages. As an example, should a person be named responsible for task *A* but then be coerced to realize *"not A"*, that person cannot be, at the very least, wholly accountable. We would rather name the coercer partially accountable if not solely accountable for her/his role in causing *"not A"*. However, even beyond the extreme example of coercion, other forces can also impact an individual's ability to realize a task. For example, say we have an organization consisting of two members: one to prepare a wall, *wall-preparer*, and another who paints the wall, *painter*. The encompassing organization would give both access rights and distribute responsibility to *wall-preparer* for prepping the wall, and to *painter* for painting the wall. Then come execution time, *wall-preparer* fulfills her/his task by spackling the wall but, having a whimsical side, unexpectedly paints a black stripe down the middle. Unfortunately, due to the unexpected action, *painter* has not the correct amount of materials and cannot fulfill her/his task. Though not coerced, *painter* cannot proceed due to the actions of another.

If we assume the complete absence of unplanned influences in a given context, responsibility directly maps to accountability. Though one might be hard-pressed to justify such an assumption in an organization with human actors, its absurdity diminishes in a world of software agents. Simply put, we must be able to adequately plan for the unexpected. To illustrate, we can take the previous example and imagine it unfolding in a multiagent setting. Before agreeing to be the organization's painter, *painter* would stipulate provisions for its role goals, in this case, *white wall*. An organization, accepting *painter*'s provisions, would then add *white wall* as a goal condition to the job description of *wall-preparer*. *Wall-preparer* would in turn accept its new goal condition. Come execution time, when *wall-preparer* adds the black stripe, not only will *painter*'s provisions not be met, but *wall-preparer* will also have violated its own goal conditions. Since *wall-preparer* knows what it did was wrong thanks to goal conditions, causally contributed to an adverse situation of work failure, and could have avoided causally contributing, it will be held accountable for the adverse state. A direct correlation between responsibility and accountability presents other advantages for accountability. As discussed in [9], when it comes time to assess an event, an investigative forum analyses who was involved, at what level, and whether those involved acted correctly or the presence of mitigating circumstances if not. If, however, we exclude the possibility of unplanned actions, any deliberations a forum would make would be already programmed in and accounted for in the very definitions of roles and groups. The deliberation would, thus, take the form of tracing violated pre and post conditions to their source.

An exclusion of unexpected actions fulfills both the positive and negative approaches to accountability. The positive approach means that an agent will be held accountable if it doesn't fulfill its job, while the negative approach means that it will not negatively affect the organization independently of its own designated job. While a realization of the positive approach implies a careful interaction pattern between an organization and its agents to keep an agent to its post conditions should its provisions hold, the negative approach implies a more temporally independent mechanism for the entire lifecycle of an organization to hold its members to correct behavior. This study concentrates firstly on the positive approach with the negative approach to follow in future studies.

Difficult ethics questions on moral responsibility and accountability present us with critical decision points wherein we must choose the "ethics" of our software agents. The famous Frankfurt example [16] confronts us with such a choice. Consider an individual who performs an action. Unknowingly, a mechanism in her/his brain would cause her/him to perform the action anyway, even should she/he choose not to act. The example complicates the assumption for moral responsibility of the alternate possibilities condition. As difficult a question the example raises, we find a comfortingly simpler scenario when we translate the example to software. Suppose an organization contains a certain agent who performs an action. That same agent by design cannot have a hidden mechanism because quite simply it knows its own plans. Plans it does not know it cannot execute. However, if the same agent finds itself in that troublesome state whatever action it performs, should the agent be held accountable for the adverse state when that state is inevitable? We adopt the incompatibilist position to moral responsibility and conclude in a software setting that an agent cannot be held accountable in causal determinism. That said, due to our distributed approach to accountability in which agents declare their provisions and goals, the burden of discovery of inevitable states lies with the agents. If an agent stipulates provisions for an inevitable adverse state, that same adversity would be represented to some degree in its provisions due to its inevitable nature. The agent would still be held accountable for the state, because it effectively declares responsibility for a goal through its accepted stipulated conditions.

Likewise as a consequence of building up provisions and goal conditions dynamically in part through agent declarations, an organization effectively excludes goal impossibilities. If an agent offers to realize a goal in the presence of a certain provisions, it is declaring for all intents and purposes that the goal is possible under certain conditions. Should that agent offer up an incorrect assessment and the goal be effectively impossible even with stipulations, the agent will nevertheless be held accountable, because the organization "believes" an agent's declaration. We therefore conclude, thanks to the distributed nature of our vision of accountability, that an organization can consider absent both impossibilities and inevitabilities.

Another ethical dilemma concerns knowledge of the consequences of one's actions. That is, can one be held accountable for an unforeseeable outcome? In software this particular ethical dilemma can take on various forms. As an initial observation, thanks to the principles of information hiding, a software agent by design cannot, and indeed should not, know all the effects of its actions in an organizational setting. In our case the modularity of software makes it possible for us to address the ethical dilemma. For instance, in the context of JaCaMo [5], an agent can see organizational goals and roles and, therefore, knows nothing beyond its own place in a chain of goals. An agent can, therefore, be accountable for interrupting its own goal, and consequently the executions of future-dependent goals, but cannot be accountable for causing an error that crashes the server. Only if an organization were to communicate to its agents that a certain state crashes the server would agents be accountable for a crash-causing state.

3 Organizational Accountability as a Design Property

As a consequence of our ethical dilemmas and general considerations of the intricacies surrounding responsibility and accountability, we conclude the concepts require the presence of a number of attributes. Because a direct mapping between responsibility and accountability requires the absence of unplanned actions, an agent must be aware of future expectations, that is, must be able to plan for all required future actions. Thus, if an organization requires an agent to realize a certain goal, the agent must be made aware before taking on its corresponding responsibility. Since goals only take on meaning in a given context, the assumption of responsibility also only meaningfully occurs within a specific context. A general acontextual responsibility assignment cannot indicate a later potential accountability due to lack of situatedness. To talk about accountability in a MAS we need to trace such a characterization back to the notions upon which MASes are traditionally built, like agents, roles, organization, environment, and interaction. To this end, we distill the following few founding principles at the basis of the way to achieve organizational accountability *as a design property*.

Principle 1 *All the collaborations and communications subject to considerations of accountability among the agents occur within a single scope that we call* organization.

This principle derives from the observation that accountability is a relationship between principals, and this is meaningful only within a specific context, i.e., the organization. The organization is a social structure in the sense given by Elder-Vass [14], i.e. an entity that is made up of parts (entities themselves), with which given relationships hold. Accountability in this perspective is a synchronically emergent property of the organization, that is a property that holds because of its parts and their relationships. The parts (or relationships) taken by themselves would not show it.

Placing an organizational based limit on accountability determinations serves multiple purposes. It isolates events and actors into more manageable pieces so that when searching for causes/effects, one need not consider all actions from the beginning of time nor actions from other organizations. Agents are reassured that only for actions within an organization will they potentially be held accountable. Actions, thanks to agent roles, also always happen in context.

Principle 2 *An agent can enroll an organization only by playing a* role *that is defined inside the organization.*

Related to the need to set a context to agent interaction, roles, which we see as an organizational and contextual aid to accountability, attribute social significance to an agent's actions and can, therefore, provide a guide to the severity of non-adherence.

Principle 3 *An agent willing to play a role in an organization must be aware of all the powers associated with such a role before adopting it.*

Following Hohfeld [18], a power is "one's affirmative 'control' over a given legal relation as against another." The relationship between powers and roles has long been studied in fields like social theory, artificial intelligence, and law. Here we invoke a knowledge condition for an organization's agents, and stipulate that an agent can only be accountable for exercising the powers that are publicly given to it by the roles it plays. Such powers are, indeed, the means by which agents affect their organizational setting. An agent cannot be held accountable for unknown effects of its actions but, rather, only for consequences related to an agent's known place in sequences of goals. By definition, accountability operates in the context of relationships between, borrowing terms from [11], an account-giver and an account-taker. To give it a computational form, it is necessary to start from the relationship-describing stipulations, for both account-giver and account-taker, of conditional expectations on account-giver's behavior. Thus, an agent will not be held accountable for an unknown goal that the organization attaches to its role, and this leads us to the next principle.

Principle 4 *An agent is only accountable, towards the organization or another agent, for those goals it has explicitly accepted to bring about.*

This means that a rational agent has the possibility to take on the responsibility for a goal only when it knows it possesses the capabilities for achieving the goal. In other words, not only is the autonomy of an agent is not constrained by accountability, it is even improved since the agent can apply forms of practical reasoning to determine whether to join an organization and under what conditions. Indeed, accountability's very domain lies in contextual dynamics and situational complication. In our own societies we are very familiar with the complexities within which accountability operates: certainly no person would agree to be held unconditionally accountable for any goal because, simply put, context matters. So too must the participating agents in a MAS have the opportunity to stipulate their own provisions, as expressed in the next principle.

Principle 5 *An agent must have the leeway for putting before the organization the provisions it needs for achieving the goal to which it is committing. The organization has the capability of reasoning on the requested provisions and can accept or reject them.*

Thinking about the use of accountability in the world of artificial agents, one might rightfully ponder accountability's potential role. From the extensive history of moral responsibility, we find two such justifications for the attribution of praise/blame: 1) because individuals deserve the feedback, and 2) in order to influence change in an individual (and indeed attribution is only appropriate when such a change occurs) [15]. Naturally, we find our own purpose in the latter. With our concept of computational accountability we strive towards a general goal of teaching agents correct behavior in the context of a particular organization. Our goal consists in realizing a fully automized system of accountability. We wish to directly program accountability in a MAS platform, specifically JaCaMo, while maintaining the most integral pieces of the concept, namely autonomy balanced with traceability and culpability.

4 An Accountability Protocol

We turn now to a MAS design-phase application of the above-mentioned accountability principles. Chopra and Singh explored a similar approach of design-phase accountability in [11]. In their work, Chopra and Singh suggest that an actor can legitimately depend on another to make a condition become true only when such a dependency is formalized in an *institutionalized expectation*, whose structure describes expectations one actor has of another and whose inherently public nature wields normative power. To tackle accountability as a design property, Chopra and Singh introduce the notion of *accountability requirement* as a special case of institutionalized expectation. An accountability requirement is a relation involving two principals, an account giver (a-giver) and an account taker (a-taker). The a-giver is accountable to the a-taker regarding some conditional expectation; namely, the expectation involves an antecedent condition and a consequent condition. Usually, the consequent condition is pursued only when the antecedent condition is true. In principle, if an accountability requirement is violated, the a-taker has a legitimate reason for complaint. The notion of accountability requirement can be further refined in terms of *commitments*, *authorizations*, *prohibitions*, and *empowerments* [11]. Each of these relations has specific implications in terms of who is accountable and for what reason. It is worth noting that an a-giver is normally accountable for a specific condition towards the whole group of agents in a MAS. That is, in an agent society, agents are accountable for their actions towards the society as a whole. Rather than creating an accountability requirement between each possible pairs of a-giver and a-taker, it is convenient to adopt the perspective by Chopra and Singh, i.e. to consider both the agents and the organization as principals, among which mutual expectations can be defined.

In other words, an organization is considered as a *persona iuris* [11], a legal person that, hence, can be the a-giver or a-taker of an accountability requirement, as any other principal represented by an agent. In addition, an organization will also be the conceptual means through which complex goals are articulated in terms of subgoals, and distributed among a set of *roles*. An organization is, therefore, a design element that allows one to specify: (1) what should be achieved by the MAS (i.e., the organizational goals), and (2) what roles are included in the organization and with what (sub)goals. Concerning accountability, an organization that shows the above features naturally satisfies principles 1 and 3.

Our intuition is that in order to obtain accountability as a design property of a MAS, the agents who are willing to be members of an organization enroll in the organization by following a precise *accountability protocol*. The organization provides the context in which accountability requirements are defined. To define such an accountability protocol, we rely on the broad literature about commitment-based protocols and focus our attention on the accountability requirements that can be expressed as (practical) commitments. Commitments have been studied at least since the seminal works by Castelfranchi [10] and Singh [22]. A social commitment is formally represented as $\mathsf{C}(x, y, p, q)$, where x is the debtor (a-giver, in our case), that commits to the creditor y (a-taker) to bring about the consequent condition q should the antecedent condition p hold. From the accountability point of view, the a-giver is accountable when the antecedent becomes true, but the consequent is false.

The gist of the accountability protocol is to make explicit the legal relationships between the agent and the organization. These are expressed as a set of (abstract) commitments, directed from organizational roles towards the organization itself, and vice versa. The first step captures the adoption of a role by an agent. Let $pwr_{i,1}, \ldots, pwr_{i,m}$ be the powers that agent Ag_i, willing to play role R_i, will get. Ag_i will commit towards the organization to exercise the powers, given to it by the role, when this will be requested by the legal relationships it will create towards other agents. In this way, the agent stipulates awareness of the powers it is endowed with, becoming accountable, not only towards some other agent in the same organization but also towards the organization itself, of its behavior:

$$\begin{aligned} cpwr_{i,1} &:: \ \mathsf{C}(Ag_i, Org, \mathsf{C}(Ag_i, Z_1, pwr_{i,1}), pwr_{i,1}) \\ &\quad \ldots \\ cpwr_{i,m} &:: \ \mathsf{C}(Ag_i, Org, \mathsf{C}(Ag_i, Z_m, pwr_{i,m}), pwr_{i,m}) \end{aligned}$$

above $Z_j, j = 1, \ldots, m$ represent some roles or some (not necessarily different) agents in the organization. These commitments represent the fact that, from an accountability-based point of view, an agent, when exercising a power because of a social relationship with some other agents, has some duties towards the social institution which provides that power, too. Indeed, when an employee is empowered by a manager to perform a given task on behalf of the company, the result is not only a commitment of the employee with the manager, but also a commitment of the employee with the company. An agent willing to play a role is expected to create a commitment that takes the form:

$$cpwr_{R_i} :: \mathsf{C}(Ag_i, Org, \mathsf{accept\_player}_{Org}(Ag_i, R_i), cpwr_{i,1} \wedge \cdots \wedge cpwr_{i,m})$$

where $\mathsf{accept\_player}_{Org}(Ag_i, R_i)$ is a power of the organization to accept agent, Ag_i, as a player of role R_i. *Org*, then, has the power to assign goals to the agents playing the various roles through $assign_{Org}$. This is done through the creation of commitments by which the organization promises to assign some goal to some agent should the agent accept to commit to pursue the goal:

$$\begin{aligned} cass_{i,1} &:: \ \mathsf{C}(Org, Ag_i, cg_{i,1}, \mathsf{prov}_{i,1} \wedge \mathsf{assign}_{Org}(Ag_i, goal_{i,1})) \\ &\quad \ldots \\ cass_{i,n} &:: \ \mathsf{C}(Org, Ag_i, cg_{i,n}, \mathsf{prov}_{i,n} \wedge \mathsf{assign}_{Org}(Ag_i, goal_{i,n})) \end{aligned}$$

Above, $cg_{i,k=1,\ldots,n}$ denote the commitments by whose creation the agent explicitly accepts the goals and possibly asks for provisions $\mathsf{prov}_{i,k=1,\ldots,n}$. Here, $goal_{i,k}$ is a goal the organization would like to assign to the agent Ag_i. The antecedent condition of $cg_{i,k}$ has the shape $prov_{i,k} \wedge assign_{Org}(Ag_i, goal_{i,k})$, where $prov_{i,k}$ stands, as said, for a provision the agent requires for accomplishing the task, and the consequent condition has the shape $achieve_{Ag_i}(goal_{i,k})$:

$$cg_{i,1} :: \mathsf{C}(Ag_i, Org, \mathsf{prov}_{i,1} \wedge \mathsf{assign}_{Org}(Ag_i, goal_{i,1}), \mathsf{achieve}_{Ag_i}(goal_{i,1}))$$
$$\ldots$$
$$cg_{i,n} :: \mathsf{C}(Ag_i, Org, \mathsf{prov}_{i,n} \wedge \mathsf{assign}_{Org}(Ag_i, goal_{i,n}), \mathsf{achieve}_{Ag_i}(goal_{i,n}))$$

Provisions are to be instantiated with those prerequisites that Ag_i discloses as necessary for it to complete its job and that Org is expected to provide. On the agent side, these commitments are the means through which the agent arranges the boundaries of its accountability within the organization. For instance, *painter*, in our example above, is an agent hired in a painting organization including also *wall-preparer*. A provision for *painter* to paint a wall could be *wall-prepared*, a condition that is to be achieved by another agent from the same organization, and that appears in the accountability requirements of its role. Should *wall-preparer* behave maliciously (as in our example), *painter* would not be accountable for not painting the wall as provision *wall-prepared* would be missing. On the organization side, provisions are part of the information used to decide whether to assign the goal to the agent (the internal decision processes of an organization are outside the scope of the paper). An agent becomes obliged to achieve a goal only after this assignment so as to not violate the accountability requirement. Finally, $achieve_{Ag_i}(g_{i,j})$ denotes that goal $goal_{i,j}$ is achieved.

We can now introduce the protocol that regulates the enrollment of an agent, Ag_i, in an organization, Org, as a player of role R_i, and the subsequent assignment of goals to Ag_i carried out by Org.

(1) $\mathsf{create}(cpwr_{R_i})$
(2) $\mathsf{accept\_player}_{Org}(Ag_i, R_i)$
(3) $\mathsf{create}(cpwr_{i,1}), \ldots, \mathsf{create}(cpwr_{i,m})$
(4) $\mathsf{create}(cass_{i,k}), k = 1, \ldots, n$
(5) $\mathsf{create}(cg_{i,k}), k = 1, \ldots, n$
(6) $\mathsf{assign}_{Org}(Ag_i, goal_{i,k}), k = 1, \ldots, n$
(7) $\mathsf{prov}_{i,k}, k = 1, \ldots, n$
(8) $\mathsf{achieve}_{Ag_i}(goal_{i,k}), k = 1, \ldots, n$

An agent Ag_i, willing to play role R_i, makes the first step by creating the commitment, $cpwr_{R_i}$ (1). By doing so it proposes itself as role player. It is worth noting that the creation of $cpwr_{R_i}$ is possible only as a consequence of Principle 3, by which an organization must disclose the powers associated with its roles. The organization is free to decide whether to accept an agent as role player (2). In case of acceptance the agent creates the commitments by which it becomes accountable with the organization of the use of its powers (3). Step (4) allows the organization to communicate the goals it wishes to assign to the agents. The agents are expected to accept them by creating the corresponding commitments of Step (5), thereby knowing which goals it may be asked to achieve at Step (6). Steps (7) and (8) respectively allow the organization to satisfy the provisions, and the agent to communicate goal achievement.

Principle 1 finds an actualization in the fact that all the mentioned commitments are created within a precise organization instance. When Org accepts Ag_i as a player for role R_i, the enrollment of the agent is successfully completed. After this step, the agent operates in the organization as one of its members. This satisfies Principle 2, for which an agent is a member of an organization only when it plays an organizational role. Principles 4 and 5 find their actualization in terms of the commitments $cg_{i,k}$'s. Principle 4 demands that an agent is accountable only for those goals it has explicitly accepted to bring about. The creation of one of the commitments $cg_{i,k}$ represents the acceptance of being responsible, and hence accountable, for the goal occurring in the commitment consequent condition. Principle 5 states that an agent must have the leeway to negotiate its own duties, which we obtain in two ways. First, the agent creates its own commitments, which means that the mission commitments might cover just a subset of the goals. Second, the agent can make explicit provisions for each role goal.

5 Case Study: Accountability in JaCaMo

JaCaMo [5] is a conceptual model and programming platform that integrates agents, environments, and organizations. It is built on the top of three platforms, namely Jason [6] for programming agents, CArtAgO [21] for programming environments, and Moise+ [20] for programming organizations. The aim of the framework is both to integrate the cited platforms, and to integrate the related programming meta-models to simplify the development of complex MASs. The presence of an actual programming platform fills the gap between the modeling level and the implementation level. According to [19], the Moise+ organizational model, adopted in JaCaMo, explicitly decomposes the specification of an organization into three different dimensions. The *structural* dimension specifies roles, groups and links between roles in the organization. The *functional* dimension is composed of one (or more) scheme(s) that elicits how the global organizational goal(s) is (are) decomposed into sub-goals and how these sub-goals are grouped in coherent sets, called missions, to be distributed to the agents. Finally, the *normative* dimension binds the two previous dimensions by specifying the roles' permissions and obligations for missions. One important feature of Moise+ [20] is to avoid a direct link between roles and goals. Roles, indeed, are linked to missions by means of permissions and obligations. In this way, the functional and the structural specifications are kept somehow independent. This independence, however, is the source of some problems when reasoning about accountability. The reason is that schemes can be dynamically created during the execution, and assigned to groups within an organization, when agents are already playing the associated roles. This means that agents, *when entering into an organization* by adopting an organizational role, *have no information about what they could be obliged to do in the future* because this information, related to a specific scheme, could be not available or even not present at all at that time. *This contradicts principle 4.*

Let's now consider an excerpt of the *building-a-house* example presented in [5]. An agent, called Giacomo, wants to build a house on a plot. In order to achieve this goal he will have to hire some specialized companies, and then ensure that the contractors coordinate and execute in the right order the various tasks and subgoals. We will focus on the second part of the work, namely the *building phase*. A company that is to be hired needs to adopt a role in the organization. Roles are gathered in a group that is responsible for the house construction. After goal adoption, a company agent could be asked (through an obligation issued by the organization) to commit to some "missions." Now, let's suppose that Giacomo is a dishonest agent and wants to exploit the contracted companies in order to achieve some purposes that are unrelated to house construction. In particular, let's suppose he wants to delegate a `do_a_very_strange_thing` goal to the agent who is playing the `plumber` role. This would be possible because an agent, when adopting a role, has no information about the kind of tasks that it could be assigned. These are, indeed, created in an independent way w.r.t. roles, and are associated with them only later. In the example, the `plumber` agent reasonably will not have a plan to achieve the `do_a_very_strange_thing` goal. Consequently, when the corresponding obligation is created, it will not be fulfilled.

Given the above scenario, who could we consider accountable for the inevitable goal failure of `do_a_very_strange_thing`? The agent playing the `plumber` role? Indeed, the agent is violating an obligation. Giacomo, because it introduced the new goal? Perhaps the system itself, since it permits such unfair behavior? The system, however, doesn't know the agents' capabilities, and cannot consequently make a fair/unfair judgment call. Our inability to attribute accountability stems from the lack of adherence to the principles 4 and 5. Goal assignment is, in fact, without replication and is performed thorugh schemes, which can even be dynamically created and associated with an existing group. Moreover, the very independence between roles and goals violates principle 4: when enacting a role in JaCaMo, agents make no claim about what kind of goals they are willing to have assigned (indeed, they could even not have the possibility to do so). For this reason they cannot be held accountable later for some organizational goal they haven't achieved. Finally, since agents do not explicitly commit to any goal while adopting a role, they cannot specify any provision needed in order to achieve a particular state of affairs. This contradicts principle 5.

Our work with JaCaMo highlights a conceptual challenge in the concept of role and role's central place in responsibility and accountability (in the form of "role-following responsibility") as illustrated by [12]. To a certain degree, decoupling a role from an organizational execution essentially negates the role's function to limit its operational domain. As illustrated in our tinkering with building-a-house, without prior agreement of what exactly a role means in a particular organizational context, we can force a role to mean whatever we want so long as the language matches. This is in contrast with Principle 4. The consequent dynamism of roles makes automatic considerations of accountability impossible to conclude. In our construction of computational accountability, roles represent a division of responsibility and pattern of interaction that serve the investigative forum to assign accountability. The accountability protocol allows achieving accountability by design by excluding that the organization assigns goals beyond the powers the agents acquire by enacting a role. Moreover, the protocol allows agents to make their provisions explicit. As one way to enforce a behavior that respects the protocol, one could modify the conceptual model of JaCaMo (and its implementation) so that it follows the five principles. Another way is to introduce proper monitors that, if needed, can check that the protocol is respected. This calls for the realization of a kind of artifact that can monitor the interaction, represent the social state (made of the existing commitments), and track its evolution. A proposal based on [1] can be found in [2].

6 Conclusions

If we adapt the approach to roles developed in [4], in which roles essentially define an organization, accountability takes on functional implications for the very definitional existence of the organization. Should some roles remain unfulfilled, an organization would correspondingly find itself in definitional crisis. As illustrated in [17], role fulfillment means continual realization of role relationships, that is, a role's duties and obligations. Accountability allows an organization some recourse in crisis and a method of expressing the relative importance its roles play. Armed with the knowledge of relative responsibility and therefore importance in the collective, an organization enables role-playing agents to make informed decisions should conflicts arise and to make their own cost/benefit analysis should one wish to not perform its function.

A mechanism based on commitments presents numerous conceptual advantages for accountability. An agent is able to specify the exact social context in which it can fulfill the specified goal. It effectively announces to the organization that should its requirements become true, it will be accountable for fulfilling the goal. Essentially the commitments require pre-execution knowledge of expectations and requirements both on the part of the organization and of the agent, which satisfies accountability's foreknowledge requirement. Commitments can therefore provide indications of responsibility, as a pre-execution assignment, which will then, thanks to the exhaustive definitions of pre and post conditions, provide a direct mapping to accountability post execution. Since the agent by design creates the commitment to the organization, the agent, not the organization, specifies its requirements to satisfy the goal. Casual determinism and impossibilities are consequently absent at an organizational level because each agent stipulates the exact social circumstances in which it can operate and realize the goal. Moreover, role relationships become explicit through the provision stipulation, which will later provide basis for role-adherence determination. The commitment structure therefore provides the necessary characteristics for beginning to speak of accountability.

Based from our beginning discussion of accountability, responsibility plays a key role in building out our accountability system. Based on work like [13], we will need to associate levels of responsibility with roles in an organization, which will serve to later implicate accountability. In order to justify a direct mapping between responsibility and accountability, in future studies we will also work to fulfill the negative approach to accountability.

Acknowledgments. This work was partially supported by the *Accountable Trustworthy Organizations and Systems (AThOS)* project, funded by Università degli Studi di Torino and Compagnia di San Paolo (CSP 2014). The authors warmly thank the reviewers for their constructive and helpful comments which helped revising the paper.

References

1. Matteo Baldoni, Cristina Baroglio, Federico Capuzzimati, and Roberto Micalizio. Commitment-based Agent Interaction in JaCaMo+. *Fundamenta Informaticae*, 2017. To appear. Available at `http://www.di.unito.it/~argo/papers/2017_FundamentaInformaticae.pdf`.

2. Matteo Baldoni, Cristina Baroglio, Katherine M. May, Roberto Micalizio, and Stefano Tedeschi. ADOPT JaCaMo: Accountability-Driven Organization Programming Technique for JaCaMo. In *PRIMA 2017: Principles and Practice of Multi-Agent Systems, 20th Int. Conf.*, Lecture Notes in Computer Science. Springer, 2017.
3. Matteo Baldoni, Cristina Baroglio, Katherine M. May, Roberto Micalizio, and Stefano Tedeschi. Computational accountability. In *Proc. of the AI*IA WS on Deep Understanding and Reasoning: A Challenge for Next-generation Intelligent Agents 2016*, volume 1802 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017.
4. Guido Boella and Leendert van der Torre. The ontological properties of social roles in multi-agent systems: definitional dependence, powers and roles playing roles. *Artificial Intelligence and Law*, 15, 2007.
5. Olivier Boissier, Rafael H. Bordini, Jomi F. Hübner, Alessandro Ricci, and Andrea Santi. Multi-agent oriented programming with JaCaMo. *Sci. Comput. Program.*, 78(6):747–761, 2013.
6. Rafael H. Bordini, Jomi F. Hübner, and Michael Wooldridge. *Programming multi-agent systems in AgentSpeak using Jason*, volume 8. John Wiley & Sons, 2007.
7. Mark Bovens, Robert E. Goodin, and Thomas Schillemans, editors. *The Oxford Handbook of Public Accountability*. Oxford University Press, 2014.
8. Matthew Braham and Martin van Hees. An anatomy of moral responsibility. *Mind*, 121(483), 2012.
9. Brigitte Burgemeestre and Joris Hulstijn. *Handbook of Ethics, Values, and Technological Design: Sources, theory, values and application domains*, chapter Designing for Accountability and Transparency: A value-based argumentation approach. Springer, 2015.
10. Cristiano Castelfranchi. Commitments: From individual intentions to groups and organizations. In *ICMAS*, pages 41–48. The MIT Press, 1995.
11. Amit K. Chopra and Munindar P. Singh. The thing itself speaks: Accountability as a foundation for requirements in sociotechnical systems. In *IEEE 7th Int. Workshop RELAW*, page 22. IEEE Computer Society, 2014.
12. Rosaria Conte and Mario Paolucci. Responsibility for societies of agents. *Journal of Artificial Societies and Social Simulation*, 7(4), 2004.
13. Mehdi Dastani and Vahid Yazdanpanah. Distant group responsibility in multi-agent systems. In *PRIMA 2016: Principles and Practice of Multi-Agent Systems*, 2016.
14. Dave Elder-Vass. *The causal power of social structures: emergence, structure and agency*. Cambridge Univ Press, 2010.
15. Andrew Eshleman. Moral responsibility. *The Stanford Encyclopedia of Philosophy*, 2014.
16. Harry G. Frankfurt. Alternate possibilities and moral responsibility. *The Jounral of Philosophy*, 66(23), 1969.
17. Nicola Guarino and Christopher Welty. Evaluating ontological decisions with OntoClean. *Communications of the ACM*, 45(2):61–65, 2002.
18. Wesley Newcomb Hohfeld. Some fundamental legal conceptions as applied in judicial reasoning. *The Yale Law Journal*, 23(1):16–59, 1913.
19. Jomi F. Hübner, Olivier Boissier, Rosine Kitio, and Alessandro Ricci. Instrumenting multiagent organisations with organisational artifacts and agents. *Autonomous Agents and Multi-Agent Systems*, 20(3):369–400, 2010.
20. Jomi F. Hubner, Jaime S. Sichman, and Olivier Boissier. Developing organised multiagent systems using the MOISE+ model: Programming issues at the system and agent levels. *Int. J. Agent-Oriented Softw. Eng.*, 1(3/4):370–395, 2007.
21. Alessandro Ricci, Michele Piunti, Mirko Viroli, and Andrea Omicini. *Environment Programming in CArtAgO*, pages 259–288. Springer US, Boston, MA, 2009.
22. Munindar P. Singh. An ontology for commitments in multiagent systems:. *Artificial Intelligence and Law*, 7(1):97–113, 1999.