An Accountability-Driven Organization Programming Technique

Stefano Tedeschi

Università degli Studi di Torino
stefano.tedeschi@edu.unito.it

Advisors: Matteo Baldoni, Cristina Baroglio, Roberto Micalizio

The continual rapid evolution of computer systems raised the issue of building complex systems composed of heterogeneous actors operating in dynamic and distributed environments. Studies in the field of multi-agent systems (MAS) showed the effectiveness of an agent-oriented approach in implementing and handling this kind of systems. This context gave rise to the development of many different programming paradigms and frameworks, such as, just to mention a few of them, *JADE* [3] and *Jason* [6] for programming agents, *CArtAgO* [12] for programming environments, and *JaCaMo* [5] for programming multi-agent organizations.

At the same time, questions emerged about the adoption of AI techniques in search engines, self-driving cars, big data analysis, and so on. Wishing to voice my own contribution, the aim of my PhD will be to investigate the notion of *computational accountability* (see [1]) in software systems, especially in multi-agent organizations, in order to develop a complete conceptual framework and programming platform for it.

1 Context

Accountability offers an example of how AI and ethics may interact, which concerns the traceability, evaluation, and communication of values and good conduct, for instance in MAS.

In general, one might see accountability as the assumption of responsibility for decisions and actions that an individual or an organization has towards another party; principals must account for their (bad) behavior when put under examination. With the rising complexity and heterogeneity of software systems, we face a crucial need for tools and mechanisms capable of ascertaining who is accountable for what in an automated way.

While in human societies accountability investigation and determination is an extremely complex task, with massive philosophical and legal implications, computational accountability remains an open challenge which could be effectively faced with the support of intelligent systems.

Many different research communities have dealt with the topic of accountability in software systems. For instance, Chopra and Singh [10] see accountability as an explicitly established context-specific relationship between two parties identified as account-giver and account-taker. Burgermeestre and Hulstijin [7], on the other hand, focus on the entire process of accountability determination, from relationship establishment to investigation, discussion and evaluation of a possible relationship violation.

The term, accountability, itself presents a rather vast range of definitions. While the main features of accountability remain relatively static, definitions vary in approach, scope and understanding in different communities. The cause of such variability lies with its socio-cultural nature as well as with considerations about the application domains. That very changeability most likely gives origin to the current lack of a comprehensive system of accountability for multi-agent environments.

Nonetheless, it is widely agreed that accountability implies certain characteristics. Some of them seem particularly interesting from a computational perspective. It's important to notice that accountability does not hinder freedom or autonomy. Accountable agents have complete freedom to do as they choose, but only will later potentially be taken to account for their actions [10]. Similarly, accountability implies agency. If an agent does not possess the qualities to act "autonomously, interactively and adaptively", that is with agency, there is no reason to speak of accountability because the agent would then be but a tool, and a tool cannot be held accountable [13]. Last, but not least, a system of accountability should be sound and complete. "In plainer words, accountability allows to place blame with all faulty agents (completeness aspect), and only with those agents (soundness aspect)" [11].

2 Organizational accountability

In human societies, organizations embody a powerful way to coordinate a complex behavior of many autonomous individuals. According to [4], an organization is a formal group of people with one or more shared goals. Agent organizations [8], in particular, are social structures and patterns of agent interaction imposed to agents to ensure a coherent global behavior.

The key notions used in defining an organization are the concepts of *role* an agent plays and *norms* imposed to this role. Following [8], the constraints imposed to an agent playing a role in an organization can be classified into several categories: goals to achieve, context-dependent obligations, authority relations, permissions and prohibitions. In other words, to enable agents understand the organization they belong to, one must describe the constraints that playing a role imposes on the behavior of an agent. This includes the specification of how playing a role influences the agent's own goals, what an agents should do when playing a role in a certain situation, what are the relations with other roles, and what it may and may not do.

In this setting the development of tools and techniques to determine and discern accountability is even more important. Organizations, indeed, often have to provide evidence of performance to stakeholders and this concerns exactly the values of accountability and responsibility.

The first part of my project will be, then, focused on the development of a sound and complete methodology and framework able to simplify the design and development of accountability-supporting organizations of agents (i.e. organizations in which it is possible to reason about accountability). Actually, the construction of a comprehensive system requires many different elements: a reliable means of communication allowing traceability and provability of (lack of) fault, an automatic forum able to at-

tribute degrees of accountability to all the involved agents, and a mechanism that keeps track of who could be accountable for what.

3 Accountability-driven programming

The approach consists in further developing the programming technique that will be presented in my M. Sc. thesis – ADOPT – and integrating it into JaCaMo+ [2]. The acronym stands for Accountability-Driven Organization Programming Technique and involves the investigation of the process for the construction of an organization. Indeed, the core of the analysis is the notion of role.

The first steps in this direction concern the development of a methodology to obtain accountability *as a design property*. In this context, it is necessary to assume that all the collaboration and communications among the agents subject to considerations of accountability occur within a single scope, called organization, and that agents can enroll in it only by playing a role defined in the organization itself. This is due to the fact that accountability operates in a specific context, given by the organization. The same role in different contexts could have radically diverse impacts on accountability attribution. From the organizational side, the organization should disclose all the expected goals associated with each of its roles before agents start to adopt them. An agent cannot be held accountable for an unknown goal and, therefore, must have full prior knowledge of all its role's goals before adoption. On the agents' side, in order to be accountable for a goal, an agent must somehow explicitly accept to bring it about and must have a leeway putting before the provisions it needs for achieving the given goal, which can be then accepted or rejected by the organization. Reasonably the organization will choose the appropriate players for each role so that their provisions will not be conflicting.

An interesting point, here, concerns the implementation of an actual protocol to be followed in order to inherently design and build accountability-supporting organizations. The starting point the will be the JaCaMo+ framework [2], a commitment-based infrastructure for programming MAS, built on the top of JaCaMo [5]. Multi-agent systems offer abstractions that provide, indeed, a promising basis of development. This is due to a representation of interaction as of social relationships among the agents, and in terms of the rules that cause the interaction to evolve. These social relationships provide expectations of agent behavior; existing ones affect the decisions of the agents they involve. In particular, the foundations of my work will be the results developed in [16, 14, 9, 15] that use the notion of *social commitment* to realize social relationships. Actually, it is possible to reify the social environment in a way that supports accountability. A social commitment $C(x,\ y,\ p,\ q)$ models the directed relation between two agents, a debtor x and a creditor y [16]. The debtor commits to its creditor to bring about the consequent condition q when the antecedent p holds, thereby creating expectations and consequently a normative value. Antecedent and consequent are generally logical formulas representing conjunctions or disjunctions of events. Commitments can be then used to realize a relational representation of interaction, where agents directly create normative binds with one another and use such binds to coordinate their activities.

The individuals involved in an interaction share a *social state* which contains all the commitments created during the execution. Every commitment has a well-defined

lifecycle and the status of it depends on the truth values of its antecedent and consequent. The evolution of the commitments in the social state guides the definitions of the activities needed by the interacting parties.

The main intuition behind an accountability protocol is that, when an agent wants to play a role in an organization, it has to explicitly accept all the *accountability requirements* associated with the role itself. Accountability requirements are expressed as a set of abstract commitments, directed from organizational roles towards the organization itself. The antecedent has the shape $prov_{i,j} \wedge assign_{Org}(R_i, g_{i,j})$ and the consequent has the shape $achieve_{R_i}(g_{i,j})$. Here, R_i is an organizational role, $g_{i,j}$ is a goal associated with it, while $prov_{i,j}$ stands for a provision. Such a provision is to be instantiated with those prerequisites that an agent, playing role R_i discloses as necessary to complete its job and that the organization (Org) is expected to provide. After the instantiation of these commitments, the organization has the power to assign goals to the agents playing the various roles through $assign_{Org}$. In this case the agent, provided that the related provisions hold, becomes obliged to achieve the goal, lest the violation of the commitment and, consequently, of the accountability requirement.

When an unexpected outcome is detected, namely an organizational goal is not achieved, it is possible to examine the social state of the interaction and detect commitments which have been violated, thereby leading to the involved parties accountable for the failure. Similarly it is possible to check whether all the needed provisions had been provided before the assignment of a given goal.

4 Impact and future work

Given that agents in a social state can influence the environment and the agents around them, social reasoning permits the exposure of more convoluted causes of a certain outcome. This can be very useful in complex systems where the more significant cause of an outcome may not stem from the last agent who affected change on the result. Potential coercion amplifies the importance of such considerations. For instance, modern enterprises are complex, distributed, and aleatory systems where business ethics and compliance programs are becoming more and more central, bringing consequently to the forefront the importance of accountability. Organizations have to be held accountable for their (mis)behavior and, therefore, provide evidence of performance. An accountability platform could provide this evidence in a transparent and automated way while simultaneously simplifying the system design, maintenance and diagnosis. Potential applications range widely from, to name a few, the field of business transactions to resource management, consumer protection, and decision support.

Future research will be developed in two directions. The first includes a further refinement of the accountability protocol for agent organizations introduced in section 3. This includes the development of an actual programming platform in JaCaMo+ able to support the design and development of accountability-supporting organizations of agents. In parallel it would be interesting to investigate the notion of accountability outside the scope of agent organizations. Actually, the absence of some constraints related to the organizational structure could lead to a far more complex scenario. One field which seems particularly intriguing is the one of business processes.

References

1. Matteo Baldoni, Cristina Baroglio, Federico Capuzzimati, Katherine M. May, Roberto Micalizio, and Stefano Tedeschi. Computational accountability. In Federico Chesani, Paola Mello, and Michela Milano, editors, *Deep Understanding and Reasoning: A Challenge for Next-generation Intelligent Agents (URANIA)*, number 1802 in CEUR Workshop Proceedings, pages 56–62, Aachen, 2016.
2. Matteo Baldoni, Cristina Baroglio, Federico Capuzzimati, and Roberto Micalizio. Programming with commitments and goals in jacamo+. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1705–1706. International Foundation for Autonomous Agents and Multiagent Systems, 2015.
3. Fabio Bellifemine, Agostino Poggi, and Giovanni Rimassa. JADE - a FIPA-compliant agent framework. In *Proceedings of the Practical Applications of Intelligent Agents*, 1999.
4. Guido Boella and Leendert van der Torre. Coordination and organization. *Electronic Notes in Theoretical Computer Science*, 150(3):3 – 20, 2006. Proceedings of the First International Workshop on Coordination and Organisation (CoOrg 2005).
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, June 2013.
6. Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldridge. *Programming Multi-Agent Systems in AgentSpeak Using Jason (Wiley Series in Agent Technology)*. John Wiley & Sons, 2007.
7. 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.
8. Cosmin Carabelea and Olivier Boissier. Coordinating agents in organizations using social commitments. *Electronic Notes in Theoretical Computer Science*, 150(3):73 – 91, 2006. Proceedings of the First International Workshop on Coordination and Organisation (CoOrg 2005).
9. Cristiano Castelfranchi. Commitments: From individual intentions to groups and organizations. In *ICMAS*, volume 95, pages 41–48, 1995.
10. Amit K. Chopra and Munindar P. Singh. The thing itself speaks: Accountability as a foundation for requirements in sociotechnical systems. In *2014 IEEE 7th International Workshop on Requirements Engineering and Law (RELAW)*, pages 22–22, 8 2014.
11. Simon Kramer and Andrey Rybalchenko. A Multi-Modal Framework for Achieving Accountability in Multi-Agent Systems. In *Proc. of Workshop on Logics in Security*, pages 148–174, 2010.
12. Alessandro Ricci, Mirko Viroli, and Andrea Omicini. Cartago: A framework for prototyping artifact-based environments in mas. *E4MAS*, 6:67–86, 2006.
13. Judith Simon. *The Online Manifesto: Being human in a hyperconnected era*, chapter Distributed Epistemic Responsibility in a Hyperconnected Era. Springer Open, 2015.
14. Munindar P. Singh. Social and psychological commitments in multiagent systems. In *In AAAI Fall Symposium on Knowledge and Action at Social and Organizational Levels*, pages 104–106. AAAI, Inc, 1991.
15. Munindar P. Singh. A conceptual analysis of commitments in multiagent systems. Technical report, Raleigh, NC, USA, 1996.
16. Munindar P. Singh. An ontology for commitments in multiagent systems:. *Artificial Intelligence and Law*, 7(1):97–113, 1999.