Computational Accountability

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. Individual and organizational actions have social consequences that call for the implementation of recommendations of good conduct at multiple levels of granularity. This work focuses on the *accountability* value, on how to express it, and on the challenges of handling accountability in a computational way, in a system of interacting agents.

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

1 Introduction

In recent years we have seen a growing attention of the scientific community towards the theme of Artificial Intelligence (AI) and Ethics. In 2016, the major conferences on AI (IJCAI, ECAI, AAAI) proposed in their programs workshops that focused on the societal implications of building AI systems. Questions were posed and issues discussed about the adoption of AI techniques and methodologies in search engines, self-driving cars, electronic markets, smart homes, military technology, big data analysis, care robots. Similar discussions reached a somewhat more mature stage of development within the scientific community that works on Robotics, where the focus mainly rests on the ethical design and application of robots and robotic systems and on the legal implications of using hardware or software devices. Significant in this respect is the recent BS 8611:2016 standard, published by BSI in April 2016.

This paper presents a work in progress that introduces and studies *Computational Accountability*. Computational Accountability is an example of a different way in which AI and Ethics may intertwine, that concerns the traceability, evaluation, and communication of values and good conduct. This is currently an open challenge, that we believe can be faced with the support of intelligent systems, and that has plenty of potential applications in such diverse fields as finance and business transactions, fair business practices, (human-resource) management, consumer protection, economic systems, corruption, taxation, sales and marketing, health care, public administration, research, smart cities, and decision support.

2 The Challenges of Accountability

What is accountability? Generally speaking, accountability is the acknowledgment and assumption of responsibility for decisions and actions that an individual, or an organization, has towards another party. The concept implicitly includes the expectation of account-giving: individuals are expected to account for their actions and decisions when put under examination. In human societies, such an examination is usually carried out by a "forum" of auditors. Accountability is an ethical value, and it is crucial in many (either institutionalized or relational) contexts in which humans interact. Human beings are naturally capable of understanding and tackling accountability. However, humans are being more and more often assisted in their work, and in their lives, by sophisticated sociotechnical systems or even by socio-cognitive technical systems (see, e.g., [11, 8]) – i.e. systems where the interacting individuals may be humans as well as artificial agents. In this setting it is extremely important to realize via software the abilities to trace, evaluate, and communicate accountability, to support the interacting parties, and to help solve disputes as a forum of auditors would do. However, tracing, communicating, and evaluating accountability is a complex task, as the next example shows.

Ted and Bill are two painters, called by Jim for estimating the cost of painting a room. Ted makes a better offer and Jim decides to assign the work to him. The walls were originally white, and Jim would like to have them painted of the same color. Bill, however, is not a good loser. Since he has a spare tin of black paint in the night he paints the walls black. When Ted finds out what happened he realizes he will not be able to satisfy the commitment he made with Jim because, in order to do a nice job, he will have to use twice as much paint as expected at a much higher cost.

This simple example shows many challenges brought about when trying to tackle accountability in a computational way. Let us suppose the agreement between Ted and Bill was formalized in some way. Ted is unable to fulfill the contract that binds him to Jim. Should the simple fact that he took a commitment, which is now impossible for him to fulfill, make a computational system conclude he is accountable for the failure? Clearly, conditions changed since when he, Bill, and Ted inspected the room. One may argue that contextual conditions that constitute the prerequisites, for Ted, for the execution of the work should have been formalized. In the real world, however, contextual conditions that hardly change over time are presumed implicitly stipulated even when they are not formalized. How could Ted foresee a possible change in the color of the room to paint? Is, then, Jim accountable? Reasoning about causes, the door may have been left unlocked. Then, it could be argued that it is Jim, and not Ted, the one to blame for not having taken care of the room. On the other hand, we know that Bill is the one who changed the colors of the room but how can Bill be considered in the process of deciding who is accountable? He is not "in the system": he was not assigned the job and has no relationship with Jim or Ted. An alternative ending of the story is that Ted, feeling responsible because of

his commitment, will paint the room at the agreed price because he values the satisfaction of the contract more than earning money.

3 A Characterization of Accountability

Definitions of accountability vary in approach and scope and different communities do not share a same understanding. This is mostly due to the fact that the notion strictly depends on socio-cultural aspects that characterize different communities and different application domains. In this section, we particularly refer to [7, 6], and sum up features of accountability that seem particularly interesting from a computational perspective.

- *Accountability is a composition of processes.* Accountability can be divided up into three processes. First, one must identify accountable parties, for what they are accountable, and to whom they are accountable. Second, if a condition verifies for which a party was previously identified as accountable, a forum of some kind convenes to gather the necessary information and passes a judgment as to the accountability of said party. Third, one must assign positive or negative sanctions to the accountable party.
- *Accountability does not hinder autonomy.* An accountable agent has complete freedom to do as they choose, but only will later potentially be taken to account for their actions [7].
- *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].
- *Accountability implies causation.* That is to say, an action or inaction on the part of the accountable party must in some way cause the previously identified situation to verify.
- *Accountability implies significance.* The forum must have the capacity to make correct judgments and identify "guilty" and "not guilty" parties. That is, the forum must discern significance. Perhaps an accountable party's action did cause the identified situation but in a manner so indirect as to render the action insignificant and the party "not guilty." Perhaps even in the case of direct causation the forum identified another accountable party who spread misinformation that influenced the actions of others. The misinformation is more significant and consequently reduces the significance of others' actions.
- *A system of accountability must be sound and complete.* Soundness means that one can prove the fault of an agent. Completeness means that one can prove that agents have acted in a correct fashion. "In plainer words, accountability allows to place blame with all faulty agents (completeness aspect), and only with those agents (soundness aspect)" [10]. A system with either of the two characteristics absent creates a dysfunctional mechanism that could either place blame with agents who acted correctly or be unable to determine fault at all.

4 Business Processes and Accountability

Modern enterprises [5] are complex, distributed, and aleatory systems: complex and distributed because they involve offices, activities, actors, resources, often heterogeneous and geographically distributed; aleatory because they are affected by unpredictable events like new laws, market trends, but also resignations, incidents, and so on. For firms, business ethics and compliance programs are becoming critical and, as the OECD reports, a growing number of them issue voluntary codes of conduct to express commitment to values like legal compliance, accountability, privacy, and trust. In this sector, the realization of accountability systems is crucial. There are attempts, especially in the literature on organizational theory, to capture systems of accountability in a rigorous way. One early example is the responsibility assignment matrix, which describes what should be done by whom with which level of responsibility so that some process happens. Four kinds of responsibility are identified: responsible (who does), accountable (who signs off), consulted (who has information), and informed (who is notified). However, methodologies and tools for designing enterprise software do not yet support the realization of accountability systems.

Two are the main approaches to the development of business models (and enterprise software): the business process approach and the artifact-centric approach. A *business process* describes how a set of interrelated activities can lead to a precise and measurable result (e.g., a service) in response to an *external event* (e.g., a new order). Business processes are used for developing software systems that concretely support the work of a firm. In this light, business processes become *workflows* that connect and coordinate different people, offices, organizations, and software in a compound flow of execution. This process-centric view enables things like the analysis of an enterprise functioning, and the identification of criticalities, like bottlenecks, but since interactions among the actors are only indirectly represented by means of input/output dependencies between activities, it hinders reasoning about accountability (because accountability mainly concerns the interactions that emerge and evolve among the parties).

The *artifact-centric approach*, e.g. [4], counterposes a data-centric vision to the activity-centric vision described above. Business artifacts are concrete, identifiable, self-describing chunks of information, the basic building blocks by which business models and operations are described. They include an *information model* of the data, and a *lifecycle model*, that contains the key states through which the data evolve, together with their transitions (triggered by the execution of corresponding tasks). The lifecycle model is not only used at runtime to track the evolution of artifacts, but also at design time to understand who is responsible of which transitions. On the negative side, business artifacts disregard the design and the modularization of those processes that operate on them and this hinders the realization of accountability systems. The reason is that accountability is affected by the execution flow, by the context, and by the role of the involved parties but artifacts only tell who is assigned this or that task.

We believe that in order to realize systems of accountability it is necessary to combine the two cited levels [1]. We are currently studying the realization of ac-

countability systems by relying on multiagent systems, whose social environment is composed of business artifacts that realize social commitments [14]. Agents embody the activity-centric view and can further be coordinated by means of protocols. Business artifacts (each with its own lifecycle) constitute the environment in which agents are situated. We quickly show how the concepts that constitute the characterization of accountability, reported in Section 3 (namely, agency, norm-autonomy, significance, causation), emerge in this perspective.

Multiagent systems offer abstractions that provide a promising basis of development. Two fundamental characteristics of agents are autonomy and situatedness. Agents are autonomous in the sense that they have a sense-plan-act deliberative cycle, which gives them control of their internal state and behavior. Agents are situated because they can sense, perceive, and manipulate the environment in which operate.

In particular, it is possible to reify the social environment of the agents in a way that supports accountability. A social commitment $C(x, y, s, u)$ models the directed relation between two agents, a debtor x and a creditor y [14]. The debtor commits to its creditor to bring about the consequent condition u when the antecedent condition s holds. Commitments evolve along a standard lifecycle as a consequence of the actions agents perform. For instance, a *conditional* commitment, whose antecedent condition results being true after some action is executed, becomes *detached*.

Commitments have a normative value because the debtor of a detached commitment is expected to bring about, sooner or later, the consequent condition of that commitment otherwise it will be liable for a violation. The fact that debtors should satisfy their commitments creates social expectations on the agents' behaviors. Nevertheless, agents remain *norm-autonomous* [9] in two ways: an agent becomes debtor of a commitment by its own decision, an agent decides whether satisfying the obligation entailed by a commitment of which it is debtor.

An agent creates commitments towards other agents while it is trying to achieve its goals (or precisely with the aim of achieving its goals) [16]. The creation of a commitment starts an interaction of the debtor with its creditor that coordinates, to some extent, the activities of the two, thus supporting the achievement of goals that an agent alone could not achieve. An agent creates a conditional social commitment towards some other agent, based on its own beliefs and goals [16]. The creditor agent will detach the conditional commitment if and when it deems it useful for its own purposes, thus activating the obligation of the debtor agent. So, conditional commitments play a fundamental role in the realization of interactivity, intended by the fact that *a message relates to previous messages and to the way previous messages related to those preceding them* [12]. In other words "there is a causal path from the establishment of a commitment to prior communications by the debtor of that commitment" [15, Sect. 4.4]. This aspect is fundamental for reasoning about accountability and responsibility. In general, the difference between responsibility and accountability can be expressed temporally: responsibility's domain lies intuitively in the future while accountability looks to the past. Responsibility requires a forward-looking

approach in the form of a capability study. An entity cannot be responsible if that entity lacks the capacity to influence results. Accountability, on the other hand, involves a posterior analysis to understand how a given result came to be. With accountability, an entity might be deemed accountable even if that entity was not preemptively identified as potentially accountable in a given context. The actual analysis of accountability can be accomplished by looking at commitment relationships, which collectively form chains of action from one context to another thanks to exchanged creditor/debitor and antecedent/consequent. On this foundation, commitments can be used to realize a relational representation of interaction, where agents, by their own action, directly create normative binds (represented by social commitments) with one another, and use such binds to coordinate their activities, e.g. through responsibility assignment, as well as to identify liabilities. Future work will focus on the temporal distinction and connections between the two concepts in order to create a comprehensive system of accountability flexible enough to move from system to system yet specific enough to understand contextual particularities and unique relationships. A starting point will be the 2COMM framework, that is described in [2, 3].

Acknowledgements. 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).

References

1. Matteo Baldoni, Cristina Baroglio, Diego Calvanese, Roberto Micalizio, and Marco Montali. Data and Norm-aware Multiagent Systems for Software Modularization. In *Proc. of EMAS @ AAMAS 2016*, pages 23–38, 2016.
2. Matteo Baldoni, Cristina Baroglio, and Federico Capuzzimati. A Commitment-Based Infrastructure for Programming Socio-Technical Systems. *ACM Transactions on Internet Technology*, 14(4):23:1–23:23, December 2014.
3. Matteo Baldoni, Cristina Baroglio, Federico Capuzzimati, and Roberto Micalizio. Commitment-based Agent Interaction in JaCaMo+. *Fundamenta Informaticae*, 2017. To appear.
4. Kamal Bhattacharya, Nathan S. Caswell, Santhosh Kumaran, Anil Nigam, and Frederick Y. Wu. Artifact-centered operational modeling: Lessons from customer engagements. *IBM Systems Journal*, 46(4):703–721, 2007.
5. David M. Bridgeland and Ron Zahavi. *Business Modeling: A Practical Guide to Realizing Business Value*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008.
6. 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.
7. 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.

8. Rob Christiaanse, Aditya Ghose, Pablo Noriega, and Munindar P. Singh. Characterizing artificial socio-cognitive technical systems. In *Proc. of ECSI-2014*, volume 1283 of *CEUR Workshop Proceedings*, pages 336–346. CEUR-WS.org, 2014.
9. Rosaria Conte, Cristiano Castelfranchi, and Frank Dignum. Autonomous Norm Acceptance. In *ATAL*, pages 99–112, 1998.
10. 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.
11. Pablo Noriega, Julian Padget, and Mark d'Inverno. The challenge of artificial socio-cognitive systems. In *Pre-proceedings of Coordination, Organizations, Institutions, and Norms in Agent Systems COIN@AAMAS*, 2014.
12. Sheizaf Rafaeli. *Sage Annual Review of Communication Research: Advancing Communication Science: Merging Mass and Interpersonal Processes*, chapter (Chapter 4) Interactivity: From new media to communication, pages 110–134. Sage, 1988.
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. An ontology for commitments in multiagent systems. *Artif. Intell. Law*, 7(1):97–113, 1999.
15. Munindar P. Singh. Commitments in multiagent systems some controversies, some prospects. In *The Goals of Cognition. Essays in Honor of Cristiano Castelfranchi*, chapter 31, pages 601–626. College Publications, London, 2011.
16. Pankaj R. Telang, Munindar P. Singh, and Neil Yorke-Smith. Relating goal and commitment semantics. In *ProMAS*, volume 7217 of *Lecture Notes in Computer Science*, pages 22–37. Springer, 2011.