Is Explanation the Real Key Factor for Innovation?

Matteo Baldoni[0000-0002-9294-0408] (✉), Cristina Baroglio[0000-0002-2070-0616],
Roberto Micalizio[0000-0001-9336-0651], and Stefano Tedeschi[0000-0002-9861-390X]*

Università degli Studi di Torino - Dipartimento di Informatica, Torino, Italy
firstname.lastname@unito.it

Abstract. Explainability is becoming a key requirement of AI applications. The availability of meaningful explanations of decisions is seen as crucial to ensure a wide range of system properties such as trustability, transparency, robustness, and innovation. Our claim is that this *need for explanation* is part of a broader problem related to the fact that most of the current architectures lack properly devised channels for collecting and for propagating feedback about decisions and actions: that is, they do not envisage nor support *accountability*. The aim of this paper is to clarify the differences between the concepts of explainability and accountability, which are often (and wrongly) used interchangeably. We draw a line of thought seeing in accountability a key factor for innovation in AI applications, and we suggest a paradigm shift from a *need for explanation* to a *need for accountability*.

1 Introduction

In the last few years, explainability has become more and more a central issue in the development of Artificial Intelligence (AI) techniques and its applications. As pointed out in [22], the current generation of AI systems offer tremendous benefits, even in distributed applications and sensitive tasks, such as self-driving cars, healthcare support, industry 4.0, etc. Their effectiveness, however, will be limited by the machine inability to explain its decisions and actions to users. Explainable AI, in turn, aims at overcoming this limit by enabling users to understand, trust, and manage this incoming generation of AI systems. Indeed, AI systems, in particular in the field of machine learning, are seen as "black boxes" because, in certain circumstances, even the designers cannot determine *why* the system came up with a given decision in high-level, meaningful terms, rather than relying on mathematical/statistical considerations. The problem is felt also in the field of Multi-Agent Systems (MAS) [32], where agents are, by their own nature, *autonomous*, meaning that their decision-making process may be opaque to the user and to the other agents. MAS proved to be valuable tools for software engineering and business process modeling, and they effectively support the design and the realization of distributed software systems. We beleive the MAS paradigm to offer powerful abstractions for realizing such complex systems (composed of many independent, but interlinked, components). In this context, the need of providing agents and MAS with capabilities and infrastructures, that make decisions/behaviors explainable, emerges not only to provide the end user with human-understandable explanations, but also from a software engineering perspective because the exchange of explanations between components is functional to the objectives of the system as a whole. For instacne, in order enable its components to successfully adapt to stressful situations.

We claim that this *need for explanation* is part of a broader problem, that is related to the absence of properly devised channels for collecting and propagating feedback about decisions and actions through a network of autonomous, yet interconnected, parts. We believe that the concept of *accountability* can provide useful tools to fill this gap. We see in accountability a key factor for innovation in AI applications, where by innovation we mean the possibility to identify a need for change and act so as to improve the system in a way that meets such a need. Consequently, we suggest a paradigm shift from a *need for explanation* to a *need for accountability*.

Modeling Explanation Explanation has been extensively studied in philosophy, psychology and cognitive sciences (for an in depth review see [28]). When referring to the term explanation we can identify two different levels: (i) the process of explaining, and (ii) the product of an explanatory process. The process of explaining generally consists in giving an answer to a "why" question about a given statement of interest [23]. Within the AI field, this process has been considered since the seminal work by Reiter [27], that lays the foundations of Model-Based Diagnosis (MBD). Generally speaking, the diagnostic process provides an interpretation of the available observations about a system of interest (i.e., symptoms), against a behavior model of the same system. The nature of the model impacts on the nature of the explanation that can be actually inferred. When the model keeps only the normal behavior of a system, diagnosis just infers a set hypotheses that are consistent with the observations, but that are not necessarily a cause of the observations. Conversely, when the model includes also abnormal behaviors, abductive diagnosis [10] infers hypotheses that not only are consistent with the observations, but are also root causes of the observations. Namely, hypotheses predict the observations by means of the system model.

On the explanation side, the commonly adopted ontology divides an explanation into two major constituents, the *explanandum* (the sentence describing the phenomenon to be explained) and the *explanans* (the class of those sentences which are adduced to account for the phenomenon). Still according to [23], which focuses on *scientific* explanation, a proposed explanation is sound if its constituents satisfy certain conditions of adequacy: (i) the explanandum must be a logical consequence of the explanans, (ii) the explanans must contain general laws, (iii) the explanans must have empirical content, (iv) the sentences constituting the explanans must be true.

Following the controversial position of [28], explanation is a pragmatic concept, in the sense that the mechanism for producing it is geared to a specific audience and selects elements to be kept or omitted accordingly. For this reason, explanations might be incomplete and what is taken for granted might be omitted accordingly. Indeed, [24] suggests that there are actually two processes in explanation, together with the product: (i) a *cognitive process* in which the causes for the event are identified, perhaps in relation to a particular counterfactual cases, and a subset of these causes is selected, and (ii) a *social process* of transferring knowledge between explainer and explainee, generally an interaction between a group of people, in which the goal is that the explainee has enough information to understand the causes of the event.

Explainability needs accountability: an example In a distributed setting, where autonomous agents are interconnected and interact, action and decisions are subject to the availability of the right contextual information, especially when an agent has to deal with abnormal events. In other terms, the relevant causal dependencies between events (a *good* explanation) can only be built in the right context. The problem is that this context is rarely the one in which the perturbation occurs, especially in complex systems, where each agent has only a partial view of the overall ongoing process. This can, however, already be seen in simple systems, like the one described below.

Money withdrawal at an ATM involves two steps: (i) the user types the desired amount; (ii) the money is provided. Suppose the typed amount is fed as a string (e.g., "100"), whose characters correspond to digits, and then it is parsed. A MAS realizing the ATM could consist of a *user agent*, in charge of interacting with the user – namely gathering the input, and providing the money –, and a *parser* agent, that receives the input string from its partner agent and converts it into a number. If the string, that is inserted by the user, is not a number in digits (e.g., "one hundred") parsing fails. A desirable behavior would be that the system, instead of crashing, were able to cope with such a situation. To this end, a good explanation (i.e., an explanation containing the information needed for recovery), should be built. Intuitively, such an explanation would highlight that the user behaved erroneously, and should support to the ATM system in deciding how to solve the problem.

Apparently, this is a simple task. However, when the parser is asked about the reasons why parsing failed, it can only provide information belonging to its context of operation: for instance, that it found an alphabetical character at index 0 in the string. Let's recall that the parser agent is unaware of the source by which data is fed as well as of the aims for which the parsing is requested. This is why "its" explanation, which is correct, turns out to be unsatisfactory. In other words, *per se* it is not really helpful for recovering from the failure – at system level.

On the other hand, the availability of proper feedback, concerning the fault, in the right context, can enable the successful identification of the root cause of the problem and, thereby, the adoption of a suitable strategy for addressing it. In the case at issue, only the user agent has the necessary contextual information for correctly interpreting the account that is produced by the parser. If the parser notifies the user agent with the relevant facts concerning the fault the latter will be in position for building an explanation of what happened and, in the simplest case, decide that a new input must be requested to the user. A more sophisticate implementation may even restructure the agent-based ATM system: if it is found that users are keen on providing the amount in words, the user agent could involve a new component, a *natural language interpreter* agent, that is able to read such strings and turn them into numbers.

This example poses a fundamental question: *is really only an explanation what is missing in AI systems or do we need something else?* As we have seen, an explanation does not have an absolute value. This is especially true in case of complex and distributed systems. Its significance depends on the context in which it is produced, and not every agent will be in position to produce explanations that are 'good' or 'useful'

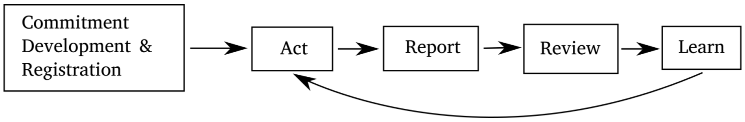

Fig. 1. A general scheme for accountability frameworks.

to certain aims, and that are functional to allow the system to adapt to a changing environment. In the following, we briefly explain how the notion of accountability can be an effective tool for this purpose, since it supports the specification of an appropriate infrastructure for building meaningful explanations.

2 Accountability in short

Accountability is extremely important in the human world, but it is a term with many meanings. The kind of accountability we refer to, which is functional to innovation, is well-described in a document by the "Executive Board of the United Nations Development Programme (UNDP) and of the United Nations Population Fund" [19]. UNDP's accountability framework describes organization-wide processes for monitoring, analysing, and improving performance in all aspects of the organization. It is an essential function that allows verifying the achievement of results, and assessing performance, based on actual data.

The framework gives managers the means to address recurring and systemic issues, and to incorporate lessons learned into future activities. Inside the framework, accountability is supported, among other things, by formally documented functions, responsibilities, authority, management expectations, policies, processes and instruments, a (complex) infrastructure for enhancing capacity-building and continuous learning. The reason is that organizations, like UNDP, are constantly evolving entities, with goals that are hard to reach and cannot be achieved at once. Failure, or partial achievement of the desired results, needs to be understood, conditions are examined, in order to either modify the organizational goals (when they turn out to be unreachable in that context with those resources) or to modify the organization itself, its practices, its structure, its competences, in order to improve performance along time. Accounts are used also by external bodies, with an oversight function, with the aim of verifying the adherence of behaviour to specific standards.

Although accountability frameworks vary considerably, depending on the kind of actors that are involved, on the kind of commitments, and on the activities that may be put under scrutiny, the same can be seen in many other (human) organizations, e.g., [31, 34, 33, 25]: the accountability framework provides the infrastructure that is necessary to a body made of many offices and individuals, geographically distributed, to collect information and provide it to those who are competent to interpret it, to take decisions and influence the future activity of the whole organization. Figure 1[1] draws a pretty

[1] The picture is inspired by the framework schemas described in [30, 34].

general schema showing the loop that goes from decisions, and actions through report to learning, a term that is used here to capture the modification of the organization itself aimed at bettering its performance, based on the gathered accounts of those who were involved. It is worth noting that this process is not an end in itself, but its ultimate goal is to exploit the information obtained through the reporting activity to learn how to the whole organizational structure can be modified and improved w.r.t. its objectives in a virtuous circle.

The kind of accountability that is put at work in organizations is well-known in sociology. Dubnick says that accountability "emerges as a primary characteristic of governance where there is a sense of agreement and certainty about the legitimacy of expectations between the community members." [18]. Even though in many contexts it is often associated to blame, this is but a partial view that disregards the potential arising from the ability and the designation to provide response about something to someone who is legitimated to ask [17]. Garfinkel, founder of ethnomethodology, considered it as a basic mechanism that allows individuals to constitute societies [21, 26].

For what concerns modeling accountability in computational terms, basically, it can be seen as a relationship between two parties: one of the parties (the "account taker" or *a-taker*) can legitimately ask, under some agreed conditions, to the other party an account about a process of interest; the other party (the "account giver" or *a-giver*) is legitimately required to provide the account to the a-taker [1, 12]. As proposed in [5], we ascribe to it two main dimensions:

1. *Normative dimension*, capturing the legitimacy of asking and the availability to provide accounts, yielding expectations on the agents' behavior;
2. *Structural dimension*, capturing that, for being accountable about a process, an agent must have control over that process and have awareness of the situation it will account for.

Intuitively, control means that a-givers are in position to produce an account, either because they were directly involved in the attempt of bringing about some event, or because they are in the position of getting the necessary information from other agents, through their accountability. An information model encompassing accountability from a computational point of view is presented in [4]. It captures what kind of data (facts) must be available to develop systems that, in any situation of interest arising in a group of interacting agents, allow the identification of account-givers.

3 Innovation through Accountability

The ability to evolve and innovate is an important property of software systems. By this we mean the ability of a system to face abnormal events (i.e., perturbations) in a constructive way, and leverage the information regarding the perturbation to restructure and possibly improve the whole system. The availability of a feedback is crucial for the realization of such a picture, yet not easy to obtain in case of distributed systems of interconnected components. Broadly speaking, the feedback can be seen as a piece of information concerning an execution of interest, that can be passed from one component to another and be exploited to face perturbations.

Multi-agent systems, and especially multi-agent organizations (MAO), are powerful abstractions that the Artificial Intelligence area proposes to build distributed systems. MAOs [6, 11, 13–16, 20] are social structures defining how multiple agents ought to interact in order to ensure a consistent global behavior oriented towards achieving one or more organizational objectives. Key features of many organizational models are a functional decomposition of the organizational goal and a normative system for coordinating the agents - norms regulate the distributed execution, targeting the organizational goals and capturing what agents have to do and which sanction is applied if they do not comply. In other words, normative organizations provide the means to realize the *correct* behavior, capturing what agents should do.

Accountability fits nicely in this picture, in the sense that it can seamlessly be integrated inside MAOs for supporting innovation, in the explained sense. Indeed, by way of accountability, a designer can specify how relevant information produced during the achievement of goals flows from an agent to another through appropriate channel; the objective is to provide an adequate context for the account-taker's decision-making, for instance, in front of abnormal situations. Accountability can, in fact, comes into play when the feedback about a perturbation is reported to the agent who is responsible for treating that perturbation. Generally speaking, treating a perturbation can mean restoring the normal execution flow disrupted by that perturbation, either by acting directly or by propagating the feedback to further agents. So, by enriching the specification of an organization with a proper set of accountability relationships among the agents, a designer can capture how the relevant information concerning abnormal events is to be propagated along the organizational structure. This is, however, not the only possible use of accountability that can support also the integration of oversight frameworks.

4 Conclusions

Many organizations, and international agencies (see e.g., [19, 31, 34, 33, 25]), recognize accountability as a key component for the proper functioning of human organizations. Accountability is, in fact, the mechanism through which important properties (such as trust, transparency, and robustness, just to mention some), can be established within a human organization. Some recent works [7, 8, 1, 5, 12] have pointed out how accountability is a useful concept also in software engineering, especially when the system at hand can be seen as a multi-agent system.

In this paper, we have put forward how accountability is essential for innovation, too. In software terms, accountability enables the collection and sharing of data, relevant for the synthesis of explanations, upon which a decisor can select new objectives, or can reconfigure the system to better meet dynamic contextual conditions. Innovation, in fact, presupposes the deviation from norms [9]: the violation of norms is not always bad, since it can sometimes lead to improving the whole system. In this process, the account provided by the norm violator may hint a lack in the system, and the decisor has, thus, the chance to solve the issue by changing the system itself.

In [1, 2, 5], we have proposed a first conceptual model of agent organization encompassing accountability as first-class element. These works outline how, by way of accountability, a multi-agent system can enjoy the property of robustness against known perturbations. To gain innovation, one has also to consider explanations (that are purpose-oriented), and the strict relation that exists between explanations and accountability. Some useful insights along this directions can be found in [12]. From an operational point of view, our proposal to support accountability in an organizational setting has found a first realization in [3], where a protocol for creating and manipulating accountability relationships has been proposed. The main intuition is that, when an agent joins to an organization, it must accept a set of accountability requirements, expressed as social commitments [29]. The protocol specifies the shapes of these commitments, and controls their creation. Accountability as a first-class modeling concept is proposed in [1] as a complement to the specification of agent organizations. We, then, presented two programming patterns for developing agents according to the accountability specifications [2]. The proposal allows one to map an accountability specification into a set of well-defined agent plans. Such plans define the behavior agents should exhibit to produce accounts for the goals they are responsible for, directed to the agents entitled for treating them.

References

1. Baldoni, M., Baroglio, C., Boissier, O., May, K.M., Micalizio, R., Tedeschi, S.: Accountability and Responsibility in Agents Organizations. In: PRIMA 2018: Principles and Practice of Multi-Agent Systems, 21st Int. Conf. No. 11224 in LNCS, Springer (2018)
2. Baldoni, M., Baroglio, C., Boissier, O., Micalizio, R., Tedeschi, S.: Accountability and responsibility in multiagent organizations for engineering business processes. In: Post-Proc. of the 7th International Workshop on Engineering Multi-Agent Systems, EMAS 2019, Revised Selected Papers. pp. 3–24. Springer (2020)
3. Baldoni, M., Baroglio, C., May, K.M., Micalizio, R., Tedeschi, S.: Computational Accountability in MAS Organizations with ADOPT. Applied Sciences **8**(4) (2018)
4. Baldoni, M., Baroglio, C., May, K.M., Micalizio, R., Tedeschi, S.: MOCA: An ORM MOdel for Computational Accountability. Journal of Intelligenza Artificiale **13**(1), 5–20 (2019)
5. Baldoni, M., Baroglio, C., Micalizio, R.: Fragility and Robustness in Multiagent Systems. In: Post-Proc. of the 8th International Workshop on Engineering Multi-Agent Systems, EMAS 2020, Revised Selected and Invited Papers. LNAI, Springer (2020), To appear
6. Boissier, O., Bordini, R.H., Hübner, J.F., Ricci, A., Santi, A.: Multi-agent oriented programming with JaCaMo. Science of Computer Programming **78**(6), 747–761 (2013)
7. Chopra, A.K., Singh, M.P.: The thing itself speaks: Accountability as a foundation for requirements in sociotechnical systems. In: IEEE 7th International Workshop on Requirements Engineering and Law (RELAW). pp. 22–22. IEEE (2014)
8. Chopra, A.K., Singh, M.P.: From social machines to social protocols: Software engineering foundations for sociotechnical systems. In: Proc. of the 25th Int. Conf. on WWW (2016)
9. Chopra, A.K., Singh, M.P.: Sociotechnical Systems and Ethics in the Large. In: AIES '18: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society. pp. 48–53. ACM (2018)
10. Console, L., Torasso, P.: A spectrum of logical definitions of model-based diagnosis. Computational Intelligence **7**(3), 133–141 (1991)
11. Corkill, D.D., Lesser, V.R.: The use of meta-level control for coordination in distributed p roblem solving network. In: Proceedings of the 8th International Joint Conference on Ar tificial Intelligence (IJCAI'83). pp. 748–756. William Kaufmann (1983)

12. Cranefield, S., Oren, N., Vasconcelos, W.: Accountability for practical reasoning agents. In: Agreement Technologies - 6th International Conference, AT 2018, Bergen, Norway, December 6-7, 2018, Revised Selected Papers. LNCS, vol. 11327, pp. 33–48. Springer (2018)
13. Dastani, M., Tinnemeier, N.A., Meyer, J.J.C.: A programming language for normative multi-agent systems. In: Handbook of Research on Multi-Agent Systems: semantics and dynamics of organizational models, pp. 397–417. IGI Global (2009)
14. Dignum, V.: A model for organizational interaction: based on agents, founded in logic. Ph.D. thesis, Utrecht University (2004), published by SIKS
15. Dignum, V.: Handbook of Research on Multi-agent Systems: Semantics and Dynamics of Organizational Models. IGI Global (2009)
16. Dignum, V., Vázquez-Salceda, J., Dignum, F.: OMNI: introducing social structure, norms and ontologies into agent organizations. In: Programming Multi-Agent Systems, 2nd Int. Workshop ProMAS, Selected Revised and Invited Papers. LNCS, vol. 3346, pp. 181–198. Springer (2004)
17. Dubnick, M.J.: Blameworthiness, trustworthiness, and the second-personal standpoint: Foundations for an ethical theory of accountability. Presented at EGPA Annual Conference, Group VII: Quality and Integrity of Governance, Edinburgh, Scotland (11-13 September 2013)
18. Dubnick, M.J., Justice, J.B.: Accounting for accountability (September 2004), https://pdfs.semanticscholar.org/b204/36ed2c186568612f99cb8383711c554e7c70.pdf, annual Meeting of the American Political Science Association
19. Executive Board of the United Nations Development Programme and of the United Nations Population Fund: The UNDP accountability system, accountability framework and oversight policy. Tech. Rep. DP/2008/16/Rev.1, United Nations (2008)
20. Fornara, N., Viganò, F., Verdicchio, M., Colombetti, M.: Artificial institutions: a model of institutional reality for open multiagent systems. Artificial Intelligence and Law **16**(1), 89–105 (2008)
21. Garfinkel, H.: Studies in ethnomethodology. Prentice-Hall Inc. (1967)
22. Gunning, D.: Explainable Artificial Intelligence (XAI). Defense Advanced Research Projects Agency (DARPA), nd Web **2**, 2 (2017)
23. Hempel, C.G., Oppenheim, P.: Studies in the logic of explanation. Philosophy of science **15**(2), 135–175 (1948)
24. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence **267**, 1–38 (2019)
25. Office of the Auditor General of Canada: 2002 December Report of the Auditor General of Canada: Chapter 9 (2002), http://www.oag-bvg.gc.ca/internet/English/parl_oag_200212_09_e_12403.html
26. Rawls, A.W.: Harold Garfinkel, Ethnomethodology and Workplace Studies. Organization Studies **29**(5), 701–732 (2008)
27. Reiter, R.: A theory of diagnosis from first principles. Artificial intelligence **32**(1), 57–95 (1987)
28. Ruben, D.H.: Explaining explanation. Routledge (2015)
29. Singh, M.P.: An ontology for commitments in multiagent systems. Artif. Intell. Law **7**(1), 97–113 (1999)
30. Sustainable Energy for All Initiative: Accountability framework, https://sustainabledevelopment.un.org/content/documents/1644se4all.pdf
31. United Nations Children's Fund: Report on the accountability system of UNICEF. https://www.unicef.org/about/execboard/files/09-15-accountability-ODS-English.pdf (2009), e/ICEF/2009/15

32. Wooldridge, M.J.: Introduction to multiagent systems. Wiley (2002)
33. World Health Organization: WHO Accountability Framework. `http://www.who.int/about/who_reform/managerial/accountability-framework.pdf` (2015)
34. Zahran, M.: Accountability Frameworks in the United Nations System (2011), `{https://www.unjiu.org/en/reports-notes/JIU\%20Products/JIU\_REP\_2011\_5\_English.pdf}`, UN Report