Implementing Business Processes in JaCaMo+ by Exploiting Accountability and Responsibility

Matteo Baldoni
Università di Torino, Dip. Informatica
matteo.baldoni@unito.it

Cristina Baroglio
Università di Torino, Dip. Informatica
cristina.baroglio@unito.it

Roberto Micalizio
Università di Torino, Dip. Informatica
roberto.micalizio@unito.it

Stefano Tedeschi
Università di Torino, Dip. Informatica
stefano.tedeschi@unito.it

KEYWORDS

BPMN, Accountability, Responsibility, Commitments, JaCaMo.

ACM Reference Format:
Matteo Baldoni, Cristina Baroglio, Roberto Micalizio, and Stefano Tedeschi. 2019. Implementing Business Processes in JaCaMo+ by Exploiting Accountability and Responsibility. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

A business process is "a set of activities that are performed in coordination in an organizational and technical environment. These activities jointly realize a business goal." [16] While several formalisms for specifying business processes have been proposed (see e.g., BPMN, BPEL, Petri Nets, DECLARE [12] and *ConDec* [13]), little attention has been so far devoted to the actual implementation of these processes, especially in a distributed scenario.

All these formalisms, in fact, place the concept of business activity (i.e., task) at the core of the process representation. This perspective is inadequate for capturing the relationships that exist between the actors involved into a distributed scenario. The implementation of the involved processes, thus, relies on choreographies, that specify the exchange of messages that should occur between the parties. Notably, however, such technologies do not provide the means for representing the goals processes aim at.

In this demo paper we exemplify a business process programming approach that provides proper abstractions for capturing goals and relationships between the actors. To this end, we propose a paradigm shift from a procedural (activity-oriented) to an agent-based approach, where the agent technology is the means through which processes are implemented, while relationships are based upon two fundamental concepts in human organizations: *responsibility* and *accountability* [4, 5, 8–11]. Specifically, we discuss how a business process specified via accountability and responsibility relationships can be implemented in JaCaMo+ [3], an extension to the well-known JaCaMo [7] agent platform in which social commitments [15] are made available as programming constructs.

A Motivating Example. As a motivating example let us consider the Hiring Process scenario [14]. The goal is to hire a new employee for a job. Many candidates will likely apply and be evaluated in parallel. As soon as a candidate is deemed apt for the position, the job is assigned and the position closed. Two distinct and *independent* BPMN processes (see Figure 1) are used: the *Hiring Process* manages the job position by opening and assigning it; the *Evaluate Candidate* process examines one candidate. The two processes are represented in independent pools because their respective instances do not have a 1:1 correspondence: the hiring process runs just once for a position, whereas the evaluation process runs for each candidate who shows up for the job. A coordination problem now rises because, as soon as one of the candidates fills the position, all the evaluations still in progress must be stopped. The *Evaluate Candidate* process instances, thus, although tackling different candidate applications, are all synchronized on the status of the position. Such a synchronization is guaranteed by introducing a *data storage* external to the processes and accessible to all of them.

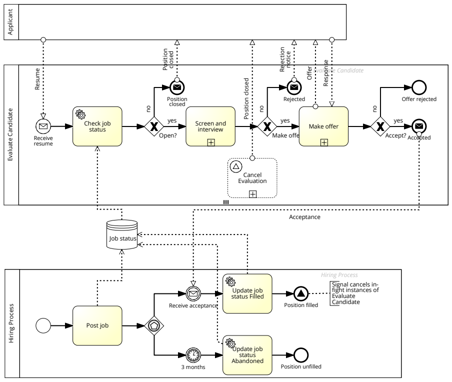

Figure 1: The Hiring Process example.

This simple yet meaningful example shows how business processes could be distributed even within a single organization. Moreover, the example shows the drawbacks of an activity-centric representation, like BPMN. In fact, the relationships between the three actors are just loosely modeled via message exchange; there is no explicit representation of the responsibilities each of them takes as a party of the interaction. For instance, the hirer is in charge for timely updating the job status in the data store when a good candidate is chosen, but the hirer does not take on such responsiblity explicitly. What is actually missing, thus, is an explicit declaration from each party of the interaction, that they are aware of their duties and of the relationships they have with others.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

2 IMPLEMENTING BUSINESS PROCESSES

Accountability and responsibility are two notions, borrowed from human organizations, that we use as a means for specifying system requirements. Intuitively, the notion of responsibility, denoted as $\mathrm{R}(x, q)$, captures the responsibility assumption by role (or agent) x about expression q. When x declares to be responsible for q, it declares to be in the position for causing q. This may imply that x has the capacity of producing q directly, or that x can exert some control on some other agent that will bring about q. Accountability, instead, is a relationship between an *account-giver* (*a-giver*) x and an *account-taker* (*a-taker*) y: x is obliged to give y an account about some condition of interest u, when a contextual condition r holds. We denote an accountability relationship by $\mathrm{A}(x, y, r, u)$. To give an account about u, x has to operate so as u becomes true [2]. In this case, the account coincides with the execution of actions that are along the path for getting u done. Now, when we think of a process being collectively executed, we can say that when the r part of the process is done, then x becomes accountable of the u part.

We now sketch the specification of the Hiring Process scenario by means of accountability and responsibility relationships. First, we identify the responsibility distribution $\mathbf{R}_{hiring}$, i.e., which responsibilities are ascribed to each agent. We have the following responsibility assignments: R(*hi*, *fill position*), the hirer is in charge of fulfilling the objective *fill position*; R(ev_i, *evaluate candidate*), the evaluator ev_i is in charge of the evaluation of a single candidate; R(i, *follow-through application*), every candidate has the objective to complete its application process. *fill position*, *evaluate candidate*, and *follow-through application* are shortcut labels standing for the processes carried out by the corresponding agents. The accountability specification $\mathbf{A}_{hiring}$ keeps all the accountability relationships between the agents, for instance: a_1 : A(ev_i, hi, post-job$_{hi}$ · apply$_i$, post-job$_{hi}$ · apply$_i$ · evaluate-candidate$_{ev_i}$) specifies that evaluator ev_i is held to give an account to hirer hi on the evaluation process (evaluate-candidate$_{ev_i}$), when a position is opened (post-job$_{hi}$), and a candidate has applied for it (apply$_i$). A complete accountability specification for the Hiring Process can be found in [6].

To be an effective instrument in the hands of the engineer, responsibility and accountability should be means for implementing process coordination at runtime in a way which is compliant with the model defined at design time. Interestingly, accountabilities (as we proposed them) can be mapped, under certain conditions, into commitment-based protocols. Consequently, any agent-based platform supporting such protocols (e.g., JaCaMo+ [3]), is a good candidate for implementing coordination among business processes. Briefly, a commitment-based protocol $\mathcal{P}$ is a set of social commitments [15] that agents can manipulate via a predefined set of operations. A commitment is denoted as $\mathrm{C}(x, y, p, q)$, meaning that agent x (debtor), is committed towards y (creditor) to bring about the consequent condition q in case the antecedent condition p is satisfied. A commitment thus formalizes a promise, or a contract, between the two agents.

In JaCaMo+, like in JaCaMo, agents are programmed as a set of *plans*, a sort of Event-Condition-Action rules, each having the following structure: ⟨*triggering event*⟩ : ⟨*context*⟩ ← ⟨*body*⟩. The triggering event is a change in the beliefs of the agent (e.g., a novel perception about the environment), the context specifies the circumstances when the plan could be used, and the *body* is the course of action that should be taken to properly handle the event. A peculiarity of JaCaMo+ is that the triggering event can also be a change of state of some commitment, of interest to the agent at hand. This allows implementing a business process, compliant with an accountability specification, as a set of plans that react to commitment state changes. Let us consider, for instance, accountability requirement a_1 reported above. Here, evaluator ev_i is held to give an account to the hirer for the evaluation process, that is, for carrying out the evaluation whenever a position is open and a candidate shows up. This is implemented in JaCaMo+ as follows: first, the accountability relationship is mapped into a similarly structured commitment, and then a set of plans is implemented to capture all the relevant state changes of that commitment. For instance, the following JaCaMo+ plan shows how the evaluator agent reacts to the detachmenent of the commitment corresponding to requirement a_1.

```
+cc(ev_i ,hi ,post-job_hi · apply_i ,
            post-job_hi · apply_i · evaluate-candidate_ev_i ,DETACHED)
:   not positionStatus(POSITION_FILLED)
<- [walk through the evaluation as specified by the process]
```

When the position is still open (context), the body (abstracted) is the sequence of actions that, according to the process in Figure 1, the evaluator performs for each candidate. A complete implementation of the Hiring Process scenario in JaCaMo+ can be downloaded from http://di.unito.it/hiringdemo.

3 DISCUSSION AND CONCLUSIONS

Goals are the final purposes that justify every activity in business processes [1]. Surprisingly enough, however, goals are not modeled explicitly in the standard modeling languages for business applications (e.g., BPMN). Especially in cross-organizational settings, the lack of an explicit representation of the business goal raises many problems, including documentation, design checking, and compliance of the implementation. We have used the notions of responsibility and accountability as explicit modeling tools, that allow distributing a business goal among different processes, yet maintaining their relations by means of accountabilities. While responsibility is concerned with the requirements agents must satisfy to play specific roles, accountability is focused on the coordination and recovery aspects. This separation of concerns fosters modularity and reuse. In fact, accountabilities and role responsibilities can be checked independently of the actual agents that will play the organizational roles [2]. The same role can be plaid by many agents, while the same agent can take part in many organizations.

Indeed, an explicit representation of accountability relationships has several advantages. First of all, it makes the assessment of the correctness of an interaction model possible. Moreover, our proposal paves the way to *compatibility* and *conformance* checks. We have also pointed out that our accountability relationships are not just an abstract modeling tool, but find a proper implementation in commitment-based protocols. The obvious advantage, thus, is to translate the interaction model into a compliant implementation.
Link to demonstration video: http://di.unito.it/hiringdemovideo
Link to source code of JaCaMo+: http://di.unito.it/2comm

REFERENCES

[1] Greta Adamo, Stefano Borgo, Chiara Di Francescomarino, Chiara Ghidini, and Nicola Guarino. 2018. On the Notion of Goal in Business Process Models. In *AI*IA 2018 - Advances in Artificial Intelligence - XVIIth International Conference of the Italian Association for Artificial Intelligence (Lecture Notes in Computer Science)*, Chiara Ghidini, Bernardo Magnini, Andrea Passerini, and Paolo Traverso (Eds.), Vol. 11298. Springer, 139–151.

[2] Matteo Baldoni, Cristina Baroglio, Olivier Boissier, Katherine M. May, Roberto Micalizio, and Stefano Tedeschi. 2018. Accountability and Responsibility in Agents Organizations. In *PRIMA 2018: Principles and Practice of Multi-Agent Systems, 21st International Conference (Lecture Notes in Computer Science)*, T. Miller, N. Oren, Y. Sakurai, I. Noda, T. Savarimuthu, and Tran Cao Son (Eds.). Springer, Tokyo, Japan, 403–419.

[3] Matteo Baldoni, Cristina Baroglio, Federico Capuzzimati, and Roberto Micalizio. 2018. Commitment-based Agent Interaction in JaCaMo+. *Fundamenta Informaticae* 159, 1-2 (2018), 1–33.

[4] Matteo Baldoni, Cristina Baroglio, Katherine M. May, Roberto Micalizio, and Stefano Tedeschi. 2016. Computational Accountability. In *Deep Understanding and Reasoning: A challenge for Next-generation Intelligent Agents, URANIA 2016*, F. Chesani, P. Mello, and M. Milano (Eds.), Vol. 1802. CEUR, Workshop Proceedings, Genoa, Italy, 56–62. http://ceur-ws.org/Vol-1802/

[5] Matteo Baldoni, Cristina Baroglio, Katherine M. May, Roberto Micalizio, and Stefano Tedeschi. 2017. Supporting Organizational Accountability Inside Multiagent Systems. In *Proceedings of AI*IA 2017 Advances in Artificial Intelligence - XVIth International Conference of the Italian Association for Artificial Intelligence (Lecture Notes in Computer Science)*, Floriana Esposito, Roberto Basili, Stefano Ferilli, and Francesca A. Lisi (Eds.), Vol. 10640. Springer, 403–417.

[6] M. Baldoni, C. Baroglio, and R. Micalizio. 2018. Goal Distribution in Business Process Models. In *AI*IA 2018: Advances in Artificial Intelligence, XVII International Conference of the Italian Association for Artificial Intelligence (Lecture Notes in Computer Science)*, C. Ghidini, B. Magnini, A. Passerini, and P. Traverso (Eds.), Vol. 11298. Springer, Trento, Italy, 252–265.

[7] Olivier Boissier, Rafael H. Bordini, Jomi F. Hübner, Alessandro Ricci, and Andrea Santi. 2013. Multi-agent Oriented Programming with JaCaMo. *Sci. Comput. Program.* 78, 6 (2013), 747–761. https://doi.org/10.1016/j.scico.2011.10.004

[8] Brigitte Burgemeestre and Joris Hulstijn. 2015. *Handbook of Ethics, Values, and Technological Design.* Springer, Chapter Designing for Accountability and Transparency.

[9] Amit K. Chopra and Munindar P. Singh. 2014. The thing itself speaks: Accountability as a foundation for requirements in sociotechnical systems. In *IEEE 7th Int. Workshop RELAW.* IEEE Computer Society. https://doi.org/10.1109/RELAW.2014.6893477

[10] Amit K. Chopra and Munindar P. Singh. 2016. From social machines to social protocols: Software engineering foundations for sociotechnical systems. In *Proc. of the 25th Int. Conf. on WWW.*

[11] Christophe Feltus. 2014. *Aligning Access Rights to Governance Needs with the Responsability MetaModel (ReMMo) in the Frame of Enterprise Architecture.* Ph.D. Dissertation. University of Namur, Belgium.

[12] Maja Pesic, Helen Schonenberg, and Wil M. P. van der Aalst. 2007. DECLARE: Full Support for Loosely-Structured Processes. In *11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), 15-19 October 2007, Annapolis, Maryland, USA.* 287–300.

[13] Maja Pesic and Wil M. P. van der Aalst. 2006. A Declarative Approach for Flexible Business Processes Management. In *Business Process Management Workshops, BPM 2006 International Workshops, BPD, BPI, ENEI, GPWW, DPM, semantics4ws, Vienna, Austria, September 4-7, 2006, Proceedings.* 169–180.

[14] Bruce Silver. 2012. *BPMN Method and Style, with BPMN Implementer's Guide* (second ed.). Cody-Cassidy Press, Aptos, CA, USA.

[15] Munindar P. Singh. 1999. An Ontology for Commitments in Multiagent Systems. *Artif. Intell. Law* 7, 1 (1999), 97–113.

[16] Mathias Weske. 2007. *Business Process Management: Concepts, Languages, Architectures.* Springer.