MAS-aided Approval for Bypassing Decentralized Processes: an Architecture

Timotheus Kampik
Department of Computing Science
Umeå University
Umeå, Sweden
Signavio GmbH
Berlin, Germany
tkampik@cs.umu.se

Amro Najjar
Department of Computing Science
Umeå University
Umeå, Sweden
najjar@cs.umu.se

Davide Calvaresi
Institute of Information Systems
University of Applied Science Western Switzerland
Sierre, Switzerland
davide.calvaresi@hevs.ch

***Abstract*—Executing business processes in a decentralized manner can improve inter-organizational efficacy. For example, blockchain-based process execution allows, at least conceptually, for cross-organizational compatibility, data integration, and integrity assurance without the need for a centralized trusted operator. However, most business processes run in agile and rapidly changing business environments. Updating a decentralized process requires continuous and extensive consensus-building efforts. Reflecting all organizations' business requirements is hardly practicable. Hence, in many real-life scenarios, to support cases with initially unforeseen properties, organizations can allow to bypass the decentralized process and fall-back to local variants. Yet, the decision to bypass or update a given process can have significant social implications since it may encourage a social dynamic that encourages collective avoidance of the decentralized process. This paper proposes a multi-agent simulation system to assess the social consequences of approving a bypass under given conditions. The proposed simulation is intended to inform the decision-maker (human or machine) on whether to allow to bypass a process or not. Moreover, we present an architecture for the integration of multi-agent simulation system, local process engine, and decentralized process execution environment, and describe a possible implementation with a particular tool chain.**

***Index Terms*—multi-agent systems, business process management, socio-technical systems**

I. INTRODUCTION

With the increasing attention *blockchain*-based technologies receive in both academia and industry, the concept of decentralized executable business processes is regarded as promising by experts in the field [1]. In this paper, we argue that the possible rise of decentralized process execution across organizational boundaries can introduce a new socio-technical problem: Strategic business process alignment is a core challenge of business process management [2]. When a decentralized process is executed across organizational boundaries, (re-)deployments and updates require the consensus of an even wider range of (inter-organizational) stakeholders. Hence, decentralized processes cannot be expected to be perfectly aligned with the requirements of an individual organization. This is why under specific circumstances (*e.g.*, time pressure), organizations may tend to allow to bypass the decentralized process in favor of a locally executed process variant, since doing so may serve the economic interest of the organization.

However, a lenient policy regarding the bypass of the decentralized process will encourage employees to try to bypass the decentralized process for the sake of convenience. For example, in a purchasing scenario, this could result in the circumvention of the standard purchase process, also known as *maverick buying*, which is in fact a common challenge, even for traditional centralized/local business processes [3]. To overcome these problems, in this paper we explore the possibility of using agent-based[1] social simulation as a decision-automation aid to determine whether a request to bypass a decentralized process should be granted. Employing Multi-Agent Systems (MAS) for social simulation is a well-established approach within the scientific community [5].

The contribution of this paper is fourfold: 1) It identifies the social implications of approving the bypass of a decentralized process as a socio-technical problem that can be expected to have a profound impact on highly decentralized business processes, and in particular blockchain-based processes. 2) It proposes a multi-agent simulation-based system to address the limitation highlighted in 1). 3) It models an architecture that integrates multi-agent simulations with business process execution (BPX) environments as a first step towards a reconciling system. 4) It selects a set of technologies to implement the proposed architecture and describes the implementation.

The paper is organized as follows. Section II introduces the core concepts relevant for this paper and motivates the problem this paper works towards addressing. The problem is formulated in Section III. Section IV describes an architecture that integrates MAS-simulation, local BPX environment, and decentralized process. Then, Section V outlines the implementation of the architecture with a set of specific technologies. Finally, Section VI discusses future work and highlights potential limitations of the introduced architecture.

II. BACKGROUND

This section discusses the three domains that are of special relevance for this paper. Section II-A presents decentralized

[1] An agent is an autonomous and self-interested software entity capable of interacting with other agents as well as its own environment [4].

business process execution, as it is the application domain for this work. Since our aim is to rely on multi-agent simulation to study the social consequences of bypassing the decentralized business process, Section II-B reviews existing works using MAS for social influence. Section II-C discusses possible synergies between blockchain technology and multi-agent systems.

A. Decentralized Business Process Execution

Business Process Management Systems (BPMS) are "generic software system(s) [...] driven by explicit process designs to enact and manage operational business processes" [6]. BPMS allow for the rapid development of complex process-oriented applications. This is typically achieved by relying on a business process diagram as a starting point, and often by automatically serializing a graphical model and executing it with a *Business Process Execution* (BPX) engine [7]. The *Business Process Model and Notation* (BPMN) [8] has established itself as a widely adopted open standard that specifies a graphical notation and a data format for business process models.

Recently, blockchain technology (BCT), and in particular the concept of *smart contracts* has been identified as a possible solution to allow for decentralized process execution. *i.e.*, when serving a set of untrusting parties, blockchain-based process execution has the potential to make trusted intermediates obsolete, since involved actors can maintain a ledger of authenticated transactions without the need to rely on a central or external authority [1].

However, decentralized business processes, executed as smart contracts, are technically immutable. Consequently, once the participating actors have agreed upon and deployed a smart contract, replacing it with a new one (*e.g.* in case the business requirements of one of the parties have changed) is a socio-technical challenge that cannot be solved with BCT itself. To overcome this challenge, organizations participating in smart contract-based business processes may choose to allow the bypass of decentralized processes and opt for a local process variant if the latter better suits their business requirements in a given case.

B. Multi-agent Systems for Social Influence Simulation

Agent-Based Social Simulation (ABSS) is a domain studying the "use of agent technology for simulating social phenomena on a computer" [9]. ABSS has been applied to simulate social phenomena in different domains such as social networks (*e.g.*, assessing marketing strategies [10], the impact of rumors [11] and population migration [12], [13] and its consequences [14]). Furthermore, multi-agent simulations have been used to build real-time decision support systems such as in [15], where the authors propose to rely on multi-agent simulation to build a decision support system for traffic regulation.

Taking into account the social dimension is common in ABSS. For instance, Barnaud *et al.* [16] rely on multi-agent simulation as a decision support system helping to determine when to update existing rules of loan allocations for farmers. In order to take the decision about the credit allocation, the system takes into account the farmers' networks of acquaintances to assess access opportunities to informal credits.

Simulating business processes aiming to assess the impact of changes made on process flow, demand, and (human) resource availability and behavior is a well-established practice both in academia and industry. While multi-agent simulation approaches are, according to our knowledge, rarely applied [17], some research adopting such an approach exists. For instance, Tarumi *et al.* [18] use multi-agent simulation to assess the impact of different personal priorities and work ethics on business process performance. However, we found no other research trying to simulate the social consequences of approving process "short cuts" or bypasses.

C. Multi-Agent Systems and Blockchain

Combining MAS and BCT yields potential advantages for both technologies. Nevertheless, the implications of such an emerging trend have still to be fully understood. The advantages can be twofold:

BCT Supporting MAS: Recent studies identified in BCT a means to foster accountability, data-protections, and trusted interactions among systems actors/agents [19]. The multi-agent paradigm and technology is used in intelligent distributed systems, often involving numerous collaborating or competing actors (e.g., Cloud Computing [20], [21], [22], e-commerce [23], and robotic applications [24]), in which sensitive data management is a common practice (*e.g.*, health care [25], [26], energy trading [27], and end-user satisfaction management [28]). Although limited, a few practical implementations of MAS equipped with BCT are emerging. Still, correct adoption (*e.g.*, if the envisioned use is in accordance with norms and laws), utility, and efficiency (*e.g.*, which data should be on-/off-chain to avoid useless burdens) cannot be taken for granted, and must be in turn critically evaluated. However, even if the technological binding is justified and correctly employed, several challenges are, according to Calvaresi *et al.*, still open: "*(i)* creating/extending legal bases for BCT, *(ii)* verifying correctness and authors of the chaincode/smart contracts, *(iii)* preserving the distributed nature of BCT, (*e.g.*, by preventing the creation of mining pools, and collusion among the nodes in the framework of public BCT), *(iv)* ensuring privacy and anonymity where appropriate, *(v)* ensuring adoption of updated BCT *(vi)* managing membership service in the framework of permissioned BCT, *(vii)* addressing scalability issues of BCT, *(viii)* ensuring the reliability of the mechanisms inter-operating with the BCT" [19]. Reputation, transparency, and traceability are crucial in case of competitive behavior among agents, whereas trust and accountability are of high importance for collaborative behavior. Nonetheless, only a subset of the MAS features that require support or improvement could gain advantages from the employment of BCT (see Table 2 in [19]).

MAS Simulations Supporting BCT-enabled Systems: MAS technology providing efficient solutions for BCT is still unexplored. For instance, coordination, negotiation and conflict resolution, domains where MAS is known to offer widely accepted contributions, are areas where MAS can contribute to enhance the processes of decentralized consensus-making and smart contract updates in BCT.

Since BCT and BCT-enabled systems are in their early staged of development, studying the social consequences of moving to decentralized business processes is key to understand the social dynamics generated by this move and help the decision maker assess the potential implications of her decisions on the organization and its human resources. This article follows this approach and addresses an application where a MAS simulation solution is used to assist a BCT-enabled application. In particular, we will address a case-study where a multi-agent simulation is used in an organization to help assess the social consequences of bypassing a decentralized process and opting for an alternative *local* process. The next section provides further details about the use case.

III. Problem Formalization

The problem addressed by this paper can be defined as follows. Given an organization that employs a decentralized process running across organizational boundaries, it can be expected that for some cases the process will not match the requirements of this organization particularly well. However, the organization cannot unilaterally update the process since the decentralized process can only be updated if consensus between the corresponding stakeholders of all participating organizations is reached. For this reason, the organization will allow for a locally controlled process variant to reach the same business goal in case following the decentralized process is impractical. In this scenario, the process participants who trigger the process need to request approval for bypassing the decentralized process in favor of the local one. Yet, the impact of the decision to approve or deny this bypass goes beyond an individual case. In particular, the decision will influence the staff opinions about the organization. For instance, in case the process is approved, the decision might encourage other employees to follow the example and request bypasses more frequently. This may lead to too lenient policies within the organization. On the other hand, rejecting too many requests, given that the decentralized process is not updated, will convey a brittle image of the organization since it cannot accommodate the emerging needs of the employees; neither by updating the decentralized process, nor through a local bypass.

The goal of the research is to find a way to better assess the consequences of approving a specific bypass request. Figure 1 represents the generic process–a local and a decentralized process variant, preceded by the case initiator's decision on whether to request bypassing the decentralized variant and, if a bypass is requested, by an automated decision task to approve or reject the bypass–as a *business process model and notation* (BPMN) diagram.

IV. Architecture

To address the problem discussed in the previous section, we propose integrating a MAS simulation solution into the process architecture. The MAS simulation determines the social consequences of approving a bypass on a per-case basis and adjusts the rules of the decision task accordingly. In this section, we describe the architecture of the proposed solution.

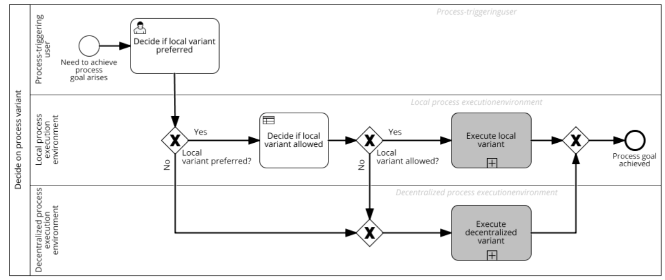

Fig. 1. Problem description: BPMN diagram

A. Process Execution Architecture

The overall process is triggered by a human user and then managed by a local BPX environment. If necessary, the MAS simulation is called from the local BPX environment as a service. If the decentralized process variant is selected, either because the user does not request a bypass or because the request is denied, the process is executed decentrally. In this case, the result of the decentralized process variant is handed back to the local BPX environment upon termination. In the BPMN diagram that describes the initial problem (Figure 1), the simulation step can be considered a BPMN *service task* that precedes the approval task (*Determine if local variant allowed*): the local BPX environment calls the MAS service to receive information about the social consequences of a potential approval.

B. Agent Architecture

Concerning the agent model, we opted for Belief Desire Intention (BDI) agents since BDI agents are widely used for social simulations [29]. BDI agents offer a reasoning formalization inspired by human mentality based on intuitive concepts that allow for a straightforward implementation in IT systems. Hence, the BDI architecture has been highlighted as a practical solution to model humans and create human-like behavior in simulated environments [30]. Furthermore, according to Adam *et al.* [29], BDI agents have been used in a variety of social simulations such as crowd simulation and escaping panic.

The following example illustrates how a BDI agent can work in our use case. The MAS simulation instantiates a set of agents representing the employees of an organization, each with an initial reputation r. The goal of each agent is to accomplish a task t, preferably before a deadline d. To meet deadline d, the agent can ask for bypass approval. Yet, the agent would only appeal for bypass in case the penalty of not making the deadline is higher than the expected combined rejection/missing deadline penalty (*e.g.* having an appeal rejected may harm the agent reputation within the community). Otherwise, the agent simply follows the standard decentralized process, even if this implies not meeting the deadline. Table I shows an example of the status of a BDI agent representing an employee at a given time instance T in the MAS simulation. Note that the beliefs, desires and intentions listed in the table are subject to change since the agent is likely to be influenced by the decisions taken by the organization, by the attitudes of its co-workers, and the evolution of the available local and decentralized process.

TABLE I
BELIEFS, DESIRES AND INTENTIONS OF A BDI AGENT REPRESENTING AN EMPLOYEE.

Beliefs	B1	The task t has a deadline d.
	B2	Following the standard decentralized process would be cumbersome for this task and will not allow me to finish a task t before the deadline d.
	B3	If I do not accomplish task t by deadline d my reputation r will be damaged and a penalty $p1$ will be withdrawn from it.
	B4	There is an alternative, *simplified* procedure that would allow me to finish before d.
	B5	I can ask for bypass to avoid the standard process variant.
	B6	Such exceptions have been granted to other employees in similar cases. I expect my chances of approval to be $x \in [0, 1]$.
	B7	Asking for a bypass that will not be not granted will hurt my reputation r and withdraw the penalty $p2$ from it.
Desires	D1	I want to finish the task t, preferably before the deadline d.
	D2	I do not want to ask for too many requests that may not be granted.
Intentions	I1	If $p1 >= (p1 + p2) * (1 - x)$: Appeal to a local approver and ask to bypass the procedure
	I2	If $p1 < (p1 + p2) * (1 - x)$: Comply with the standard decentralized process in the first place.

C. Decision Architecture

The output of the MAS simulation is provided as a process run-time variable to one or multiple rules that inform the approval decision task. For example, the MAS simulation could set the ordinarily scaled process variable *social_spread* to a value $\in \{none, low, medium, high\}$, based on the impact the simulated *approval* decision has in comparison to its *rejection* counterpart. In addition to the rules that handle these variables, the decision task may contain static rules that can be adjusted during design time, for example when the risk preferences of the process owner change. Listing 1 shows the pseudo-code for an example set of decision rules [2].

Listing 1. Approval decision pseudo-code

```
decline if:
    (social_spread = high
    AND
    business_benefit >= medium)
    OR
    (compliance_violation >= low)
else: accept
```

[2]Note that all variables are ordinarily scaled.

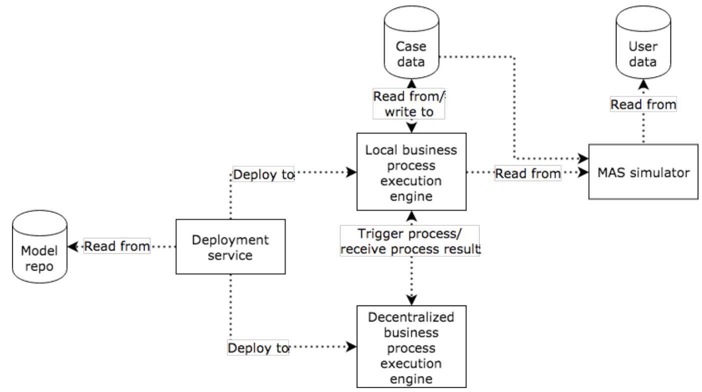

Fig. 2. Architecture proposal: graphical representation

BPX environments typically have specialized engines for dealing with decision rules [31].

D. Components

The architecture consists of the following components:

- a business process repository that contains the process definitions that are to be deployed;
- a service that deploys the process definitions to the local and decentralized process execution engines;
- a local BPX environment that logs the user request and preference (on whether to bypass or not), handles the bypass approval, and calls the decentralized process variant;
- a database that stores the case history and that can be accessed by the MAS simulator to determine the social consequences of granting a bypass approval to a specific user;
- a decentralized BPX environment that is called by the local BPX environment and then runs the decentralized process to finally return the process result back to the local environment;
- a MAS simulation environment that is called as a service from the local BPX engine to help determine whether a bypass request should be approved or not;
- a user database that contains social structures or activities within the organization.

Figure 2 depicts the proposed system architecture.

V. PROPOSED IMPLEMENTATION

In this section, we suggest a technology tool chain for implementing the proposed architecture. The purpose of the technology selection is to show how to implement simulations and simple running examples in a test environment. We acknowledge that for practical, large-scale applications, a scenario-specific software selection is necessary. We suggest implementing the architecture with the following technologies:

- **Business process modeling repository: git-based, BPMN diagrams.** The process definitions can be created and saved in the Business Process Model and Notation (BPMN) [8], which is an open standard for a graphical notation, as well as an XML-based data exchange format. BPMN diagrams can be created with a range of different modeling tools [32][3]. To model the rules of the bypass approval task, Decision Model and Notation (DMN) diagrams can be used; DMN is an open standard for decision rule modeling that seamlessly integrates with BPMN (see for example Biard *et al.* [31]). git is an open-source version control system that is commonly used to manage source code versions and hence can be regarded a good fit to manage the source of the BPMN process definition, which have source code-like properties[4].

- **Local BPX environment: jBPM.** jBPM is a Java-based business process execution engine [34] that supports the definition and execution of BPMN (and DMN) XML diagrams. As it is open-source and well documented, it is a reasonable choice for implementing the research prototype.
- **Decentralized BPX environment: Ethereum.** Ethereum is a decentralized ledger that allows for the definition and execution of decentralized business rules, so-called *smart contracts* [35]. Although Ethereum can currently be considered as not sufficiently stable for many practical use cases (in particular because of security concerns, see Atzei *et al.* [36]), its ability to execute any program (Turing-completeness) and good developer tooling support make it a reasonable choice for a research prototype. A specific advantage for the use case in focus is the existence of an Ethereum-based process execution engine (albeit a scientific prototype, see below), which does at the time of writing not exist for any other smart contract execution platform.
- **Case data base: MySQL.** Case (process instance) data is stored by jBPM, which supports a range of databases. A reasonable choice for a database is MySQL, a stable and well-established open-source relational database system.
- **Deployment service: Caterpillar/TypeScript-based.** As jBPM integrates with git by default, only the deployments to Ethereum need to be custom implemented. The key challenge is to convert the BPMN XML files into Solidity smart contracts. For this, the Caterpillar engine [37] (an academic research prototype) can be used. To integrate Caterpillar with git, it makes sense to use the JavaScript variant *TypeScript*, the language in which Caterpillar is implemented.
- **MAS simulation environment: Jason with MySQL connector.** Jason [38] is a multi-agent system development and simulation environment that is well-established in MAS research community. We suggest using Jason, as it is has proven itself as a good system for BDI simulations (*c.f.* [29] and the references therein) and–as a Java-based system–is expected to integrate well with jBPM. Also, Jason supports or goal to implement social agents particularly well, as it allows agents to easily receive communications from other agents, and to select ”sociably acceptable messages” [38].
- **User data base: MySQL.** To keep complexity low, it makes sense to use the same database system for the user data as for the case data. Hence, it is a reasonable proposal to use MySQL for *this* database as well. The user database can be filled with data that reflects the directory service structure of the organization in focus, or with aggregates of the communication behavior of the employees from corporate email and messaging applications.

[3]For an overview of BPMN XML-compliant process modeling tools see http://bpmn-miwg.github.io/bpmn-miwg-tools/.

[4]For an introduction to git for scientists, see Blischak *et al.* [33].

VI. Discussion

In this section, we highlight the limitations of the presented solution (Subsection VI-A) and discuss future research (Subsection VI-B).

A. Limitations

In this paper, we present early stage research. In particular, we want to highlight the following limitations:

- The specifics of how to run the multi-agent simulation that informs the bypass approval decision remain unspecified, primarily because many aspects of this simulation are specific to the organization that applies the solution.
- We have not yet implemented the architecture with the proposed technologies. Although the technology selection is well motivated, changes to the technology stack might be considered reasonable at the time of implementation.
- Deploying the proposed solution in practice is challenging because Ethereum and in particular Caterpillar can be considered insufficiently mature for many large-scale scenarios.

It is worth highlighting that our architecture is relevant beyond *blockchain*-based decentralized processes and can be applied for assessing the social consequences of approving the bypass of any process variant. However, applying the architecture to more generic scenarios requires re-thinking the proposed technology stack.

B. Future Research

For future research, we suggest the following: **1) Implement the proposed architecture.** In Section V, we proposed a set of technologies, with which the architecture can be implemented. We suggest implementing a scientific prototype that considers our proposal. **2) Run business process simulations that assess the benefits of the proposed approach.** Initial evaluations of the architecture can be conducted in simulation environments. Simulations can compare the effect of the MAS-aided bypass approval with the effects automated bypass approvals with a set of static decision rules would have. In an additional comparison, human approval behavior (that can be affected by social pressure) can be simulated. **3) Deploy the architecture in practice.** To have any real-world impact, our architecture needs to be deployed in a practice scenario. Ethereum-based business processes execution is to our knowledge not common practice at the time of writing. Hence, we suggest first real-world implementations of the architecture can abstract from the *smart contract* scenario, and instead use the solution to reduce the rates at which traditional cross-organizational are bypassed.

ACKNOWLEDGMENT

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

REFERENCES

[1] J. Mendling, I. Weber, W. V. D. Aalst, J. V. Brocke, C. Cabanillas, F. Daniel, S. Debois, C. D. Ciccio, M. Dumas, S. Dustdar *et al.*, “Blockchains for business process management-challenges and opportunities,” *ACM Transactions on Management Information Systems (TMIS)*, vol. 9, no. 1, p. 4, 2018.

[2] M. Rosemann and J. vom Brocke, “The six core elements of business process management,” in *Handbook on business process management 1*. Springer, 2015, pp. 105–122.

[3] K. Karjalainen, K. Kemppainen, and E. M. van Raaij, “Non-Compliant Work Behaviour in Purchasing: An Exploration of Reasons Behind Maverick Buying,” *Journal of Business Ethics*, vol. 85, no. 2, p. 245, May 2008.

[4] M. Wooldridge, *An introduction to multiagent systems*. John Wiley & Sons, 2009.

[5] R. Conte, N. Gilbert, and J. S. Sichman, “Mas and social simulation: A suitable commitment,” in *International Workshop on Multi-Agent Systems and Agent-Based Simulation*. Springer, 1998, pp. 1–9.

[6] W. M. Van Der Aalst, A. H. Ter Hofstede, and M. Weske, “Business process management: A survey,” in *International conference on business process management*. Springer, 2003, pp. 1–12.

[7] G. Li, V. Muthusamy, and H.-A. Jacobsen, “A distributed service-oriented architecture for business process execution,” *ACM Transactions on the Web (TWEB)*, vol. 4, no. 1, p. 2, 2010.

[8] OMG, *Business Process Model and Notation (BPMN), Version 2.0*, Object Management Group Std., Rev. 2.0, January 2011.

[9] P. Davidsson, “Agent based social simulation: A computer science view,” *Journal of artificial societies and social simulation*, vol. 5, no. 1, 2002.

[10] E. Serrano and C. A. Iglesias, “Validating viral marketing strategies in twitter via agent-based social simulation,” *Expert Systems with Applications*, vol. 50, pp. 140–150, 2016.

[11] E. Serrano, C. A. Iglesias, and M. Garijo, “A survey of twitter rumor spreading simulations,” in *Computational Collective Intelligence*. Springer, 2015, pp. 113–122.

[12] C. D. Smith, “Modelling migration futures: development and testing of the rainfalls agent-based migration model–tanzania,” *Climate and Development*, vol. 6, no. 1, pp. 77–91, 2014.

[13] J. A. Sokolowski, C. M. Banks, and R. L. Hayes, “Modeling population displacement in the syrian city of aleppo,” in *Proceedings of the 2014 winter simulation conference*. IEEE Press, 2014, pp. 252–263.

[14] S. Picascia, A. Termos, and N. Yorke-Smith, “Initial results from an agent-based simulation of housing in urban beirut,” in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 2045–2047.

[15] F. Balbo and S. Pinson, “Dynamic modeling of a disturbance in a multi-agent system for traffic regulation,” *Decision Support Systems*, vol. 41, no. 1, pp. 131–146, 2005.

[16] C. Barnaud, F. Bousquet, and G. Trebuil, “Multi-agent simulations to explore rules for rural credit in a highland farming community of northern thailand,” *Ecological Economics*, vol. 66, no. 4, pp. 615–627, 2008.

[17] P. Bocciarelli, A. D’Ambrogio, A. Mastromattei, E. Paglia, and A. Giglio, “Business process modeling and simulation: State of the art and msaas opportunities,” in *Proceedings of the Summer Simulation Multi-Conference*, ser. SummerSim ’17. San Diego, CA, USA: Society for Computer Simulation International, 2017, pp. 6:1–6:12.

[18] H. Tarumi, T. Matsuyama, and Y. Kamabayashi, “Evolution of business processes and a process simulation tool,” in *Proceedings Sixth Asia Pacific Software Engineering Conference (ASPEC’99) (Cat. No.PR00509)*, Dec 1999, pp. 180–187.

[19] D. Calvaresi, A. Dubovitskaya, J. P. Calbimonte, K. Taveter, and M. Schumacher, “Multi-agent systems and blockchain: Results from a systematic literature review,” in *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, 2018, pp. 110–126.

[20] D. Talia, “Clouds meet agents: Toward intelligent cloud services,” *IEEE Internet Computing*, vol. 16, no. 2, pp. 78–81, 2012.

[21] K. M. Sim *et al.*, “Agent-based cloud computing,” *IEEE transactions on services computing*, vol. 5, no. 4, pp. 564–577, 2012.

[22] A. Najjar, O. Boissier, and G. Picard, “Aquaman: an adaptive qoe-aware negotiation mechanism for saas elasticity management,” in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2017, pp. 1655–1657.

[23] M. Fasli, *Agent technology for e-commerce*. John Wiley & Sons Chichester, 2007.

[24] D. Calvaresi, P. Sernani, M. Marinoni, A. Claudi, A. Balsini, A. F. Dragoni, and G. Buttazzo, “A framework based on real-time os and multi-agents for intelligent autonomous robot competitions,” in *2016 11th IEEE Symposium on Industrial Embedded Systems (SIES)*, 2016, pp. 1–10.

[25] A. Dubovitskaya, V. Urovi, I. Barba, K. Aberer, and M. I. Schumacher, “A multiagent system for dynamic data aggregation in medical research,” *BioMed Research International*, vol. 2016, 2016.

[26] D. Calvaresi, D. Cesarini, P. Sernani, M. Marinoni, A. F. Dragoni, and A. Sturm, “Exploring the ambient assisted living domain: a systematic review,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 8, no. 2, pp. 239–257, 2017.

[27] J. Cerquides, G. Picard, and J. A. Rodríguez-Aguilar, “Designing a marketplace for the trading and distribution of energy in the smart grid,” in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 1285–1293.

[28] A. Najjar, C. Gravier, X. Serpaggi, and O. Boissier, “Modeling user expectations & satisfaction for saas applications using multi-agent negotiation,” in *Web Intelligence (WI), 2016 IEEE/WIC/ACM International Conference on*. IEEE, 2016, pp. 399–406.

[29] C. Adam and B. Gaudou, “Bdi agents in social simulations: a survey,” *The Knowledge Engineering Review*, vol. 31, no. 3, pp. 207–238, 2016.

[30] E. Norling, “Folk psychology for human modelling: Extending the bdi paradigm,” in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*. IEEE Computer Society, 2004, pp. 202–209.

[31] T. Biard, A. Le Mauff, M. Bigand, and J.-P. Bourey, “Separation of decision modeling from business process modeling using new decision model and notation(dmn) for automating operational decision-making,” in *Working Conference on Virtual Enterprises*. Springer, 2015, pp. 489–496.

[32] M. Kurz, “Bpmn model interchange: The quest for interoperability,” in *Proceedings of the 8th International Conference on Subject-oriented Business Process Management*, ser. S-BPM ’16. New York, NY, USA: ACM, 2016, pp. 6:1–6:10.

[33] J. D. Blischak, E. R. Davenport, and G. Wilson, “A quick introduction to version control with git and github,” *PLoS computational biology*, vol. 12, no. 1, p. e1004668, 2016.

[34] M. Cumberlidge, *Business process management with JBoss jBPM*. Packt Publishing Ltd, 2007.

[35] G. Wood, “Ethereum: A secure decentralised generalised transaction ledger,” *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[36] N. Atzei, M. Bartoletti, and T. Cimoli, “A survey of attacks on ethereum smart contracts (sok),” in *Principles of Security and Trust*. Springer, 2017, pp. 164–186.

[37] O. López-Pintado, L. García-Bañuelos, M. Dumas, and I. Weber, “Caterpillar: A blockchain-based business process management system,” in *Proceedings of the BPM Demo Track and BPM Dissertation Award co-located with 15th International Conference on Business Process Modeling (BPM 2017), Barcelona, Spain*, 2017.

[38] R. H. Bordini and J. F. Hübner, “BDI agent programming in AgentSpeak using Jason,” in *International Workshop on Computational Logic in Multi-Agent Systems*. Springer, 2005, pp. 143–164.