



Organizational Modeling for Blockchain Oriented Software Engineering with Extended-i* and UML

Anne Sofie Vingerhouts¹, Samedi Heng² , and Yves Wautelet¹ 

¹ KU Leuven, Leuven, Belgium

yves.wautelet@kuleuven.be,

² HEC Liège, Université de Liège, Liège, Belgium

samedi.heng@uliege.be

Abstract. No standard modeling technique exists for Requirements Engineering (RE) and Organizational Modeling (OM) within Blockchain-Oriented Software Engineering. This paper aims to provide preliminary advices through the study of two modeling frameworks when representing blockchain-supported processes in Supply Chain (SC) Management (SCM), namely the i* framework and the Unified Modeling Language (UML) Use Case and Sequence Diagrams. This paper illustrates findings on a real life case study called ‘Farm-to-Fork’. The case study provides a blockchain solution for the SC of farm animals. The modeling techniques are applied to uncover their pros and cons in this context. The paper points to the use of (extended) i* representations and the aforementioned UML diagrams in a complementary way because of the various perspectives they provide to develop a blockchain: while i* fits during early RE/OM phases to understand the ‘why’ of the SC processes, UML better fits the late RE/OM and design stages by offering concrete diagrams to understand the ‘what’ and ‘how’.

Keywords: i* framework, Blockchain, Blockchain-Oriented Software Engineering, Conceptual Modeling, Supply Chain Management, Distributed Ledger.

1 Introduction and Related Work

Since Bitcoin’s boom in 2018, the use of the underlying Blockchain Technology (BT) to store transactional data has driven a lot of interest. Application of BT can notably be found in SCM. Major characteristics of BT like immutability, transparency, traceability, high transactional speed, security and cost-effectiveness are indeed well aligned with SCs key objectives [10]. There is no commonly agreed modeling standard for BT adoption [15] so that this paper studies what modeling language could be used for such a purpose with a specific focus on the SC domain. Two of them, i* [28] (which has proven its relevance for complex organizational modeling) and UML’s Use Case and Sequence Diagram [13] (largely adopted in the industry) are applied and evaluated.

The i* framework is a goal-oriented graphical requirement modeling notation [28]. It allows an early requirement engineering analysis in environments where social actors

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

depend on each other for goals to be achieved, tasks to be performed, and resources to be furnished [28]. Previous researches proved the relevance and utility of i^* to model organizational requirements of a “multi agent system” [24] facilitating stakeholder’s interactions by depicting their dependencies and hence providing a mean for coordination. i^* was previously used to model several organizational settings such as online stores [8], hospital beds management [22,25], health care [28], SCs and more specifically outbound logistics [21], production support in the steel industry [26] and also for the development of higher education platforms like collaborative learning software [9] and MOOCs [23]. The i^* framework is divided in two parts providing each a different level of abstraction: The Strategic Dependency (SD) and the Strategic Rationale (SR) model [28]. Figure 1 provides the core concepts and icons of the i^* framework. The SD model shows dependencies and the SR model depicts internal intents.

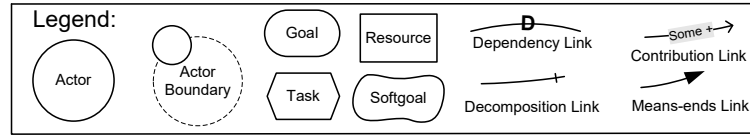


Fig. 1. Relevant i^* Concepts and their Icons.

UML Use Case Diagrams are well known to describe the cases in which a system can be used. UML Sequence Diagrams are a more widely known model that allows to describe the interactions between the different actors and a central system. This is useful for blockchain in SC Requirements Engineering (RE) because the Sequence Diagram can depict a specific order of system operations, which corresponds very well to the nature of the SC flow. This similarity makes Sequence Diagrams a well-established candidate to model blockchain initiatives in the SC domain.

While previous literature has touched upon the adoption of BT for SCM, it has failed to conceptually model these processes for software engineering. For example, Niranjanamurthy et al. [12] discuss how blockchain can meet SC objectives and present a few small case studies to demonstrate how this technology is already used in businesses. The paper nevertheless includes only superficial process descriptions. Other research articles, like Saberi et al. [18] and Apte & Petrovsky [1] discuss the use of blockchain in SC and its benefits and challenges but without providing a case study or conceptual model. Bettín-Díaz et al. [3], Roa [14] and Casado-Vara et al. [4] provide exemplary flowcharts, but these only describe a generic implementation of blockchain in the SC of virtually any company in any industry. Furthermore, Rocha et al. [15] and Marchesi et al. [11] have tried to model blockchain implementations for a fidelity point program and for the workings of a university group, by using different UML techniques. However, both these cases were mostly fictional and limited.

This paper extends on previous research by Ben Hamadi et al. [5]. The latter paper studied the use of the i^* modeling language for BT in SC for Blockchain-Oriented Software Engineering (BOSE), based on a case study of a Belgian retail giant. The study in this paper further investigates and elaborates on this notably by implementing extensions to i^* as proposed by Ben Hamadi et al. [5] but not developed in there; this

has been done on a genuine case study. Moreover, the present research additionally applies UML as a modeling technique. The latter is widely adopted in businesses for specification stages in software engineering.

2 Research Paradigm, Question and Methodology

Research Paradigm. The research presented here takes roots in the Design Science paradigm [6]; the latter aims to deliver generic solutions for known (or not yet considered) problems. The result of a design science research problem can be a solution in the form an artifact, terminology, methodology, engineering tool, and so forth. In the present research, we have enriched the *i** framework to better match with the problematic of blockchain as well as applied *i** and UML models for BOSE. Strengths and weaknesses of the models are explored, and a comparison between the frameworks is presented, based on a set of criteria.

Research Question. *Are extended *i** models and UML Use Case/Sequence Diagrams appropriate modeling techniques to visualize the organizational structure of blockchain ecosystems and how do these two frameworks compare?*

Research Methodology. To answer the research question, a case study is required [17]. The chosen case study is a ‘Farm-to-Fork’ project from a Belgian consultancy company. ‘Farm-to-Fork’ is a SC tracking prototype that uses blockchain to digitize the food SC and make it more transparent. The Farm-to-Fork project does not have any technical documentation available, so all information was gathered through interviews. Interviews have been conducted with two experts of blockchain working at the consultancy company to gather the domain knowledge.

A first interview was conducted in February 2020 with Interviewee 1 (I1) a blockchain consultant who has worked on, among others, blockchain projects for the Belgian government. A second interview was conducted in April 2020 with Interviewee 2 (I2) another blockchain expert working at the same company. He provided some additional insights. Out of the information gathered from the interviews, we have elaborated several conceptual models using both *i** and UML techniques. A third, final validation session was organized in May 2020 with I2; the latter then validated and confirmed the case study description as well as the associated representations. After applying both modeling techniques to the case study, a summary of their pros and cons is provided. Comparing both techniques is useful to determine their shortcomings.

3 Case Study

The case study, called Farm-to-Fork concerns a blockchain solution made to track farm animals throughout the SC process, from “their birth to your plate”. The solution also includes an easy to use app that gives an overview of the stages of the SC process, including QR-codes to track animals. Every participant in the network, and therefore every node in the SC, can quickly check the origin of the animal, the quality and the different previous steps that the animal has gone through.

3.1 Farm-to-Fork

The Farm-to-Fork prototype was created to meet the increasing expectation levels for improved transparency in the food industry. This solution provides an answer to many

of those struggles. I2 suggests that the most important benefits of this implementation are the traceability and the liability aspects. Traceability ensures the ability for the SC participants to closely monitor the animals and allows them to know the exact state and quality of the animal (product). Therefore, it becomes much easier to detect contaminated batches, and to identify any such batches before they can reach the final SC node (such as supermarkets) where they may create a health hazard to unknowing consumers. This also helps to reduce waste. Additionally, I2 remarks that, even if a contaminated product manages to get to consumers, it is much easier to trace down the specific faulty batches, since all product information is meticulously and individually stored in the blockchain. Therefore, in case a contaminated batch would still reach the end-consumers, the health associated consequences will be much less severe.

The liability aspect that I2 mentions refers to knowing all the actions of the SC nodes, including their consequences. For example, fragile chicken eggs that are transported from node to node throughout the SC can break at any stage. However, disputes can arise between the participants of the SC network about who is responsible for this. With blockchain, these disputes can be settled very quickly as the database can tell when and where every individual egg broke. Furthermore, the advantages do not only apply to the producers, but also to the consumers, since the idea is also to expose a part of the blockchain to them. Consumers can view information about a specific animal product in the supermarket by scanning a QR-code. This enables consumers to verify the origin and all the process steps that the animal has gone through.

However, it is important to mention that consumers should only have access to restricted, but relevant information. If a consumer would also be able to see exactly how many chicken eggs they've bought from a specific farmer, they might want to skip some parts of the SC cycle and go straight to that farmer for eggs, leaving the rest of the SC nodes redundant and unprofitable. I2 underlines the importance to carefully assign specific access rights to each participant.

More information about the case study – the type of blockchain and the used raft consensus – can be found in Appendix I³.

3.2 Overview of the Farm-to-Fork Blockchain participants

The Farm-to-Fork solution is used in a context of farm animals that go through the SC. For this paper, the case study takes an in depth look at the logistic flow of chicken meat from farmer to consumer.

The possible participants for the blockchain project, their respective roles within the SC and their minimum required input into the blockchain are listed in Table 1 (the detail is available in Appendix II). This represents a generic model of how the solution works in this context. More (or less) participants could be involved, depending on the needs and context of each specific SC's structure.

4 Using i* to model the Farm-to-Fork Blockchain

4.1 Proposed Extensions for i* for Modeling Blockchains

Two types of extensions of i* are proposed in this paper: *privacy* and *laws and norms*.

³ All appendices are available at: <http://dx.doi.org/10.17632/zkygycmz2t.1>

Table 1. Farm-to-Fork Blockchain network participants.

Blockchain participant	Task	Input
Farmer	Raising chickens.	Characteristics of chickens, the kind of poultry feed used, the sicknesses of chickens and their antibiotics, confirmation of number of caught chickens.
Catcher	Catching the chickens.	Which chickens were caught and in which order.
Transporter to butcher	Transports the chickens from the farm to the butcher.	The conditions of the transportation such as the humidity, the temperature, the shipment status.
Butcher	Prepares the chicken meat.	Number of chickens (slaughtered), treatments, treatment conditions, storage, storage conditions, sizes and volume of the pieces, quality control.
Packager	Packs the meat according to needs of the supermarket.	The amount of meat, the size and volume.
Transporter to supermarket	Transports the chicken meat from the packager to the supermarket.	The conditions of the transportation such as the humidity, the temperature, the shipment status.
Supermarket	Sells the chicken meat to consumers.	Amount of chicken meat and volume sold, stock data, waste data, quality control.

Bashir [2] and Bettin-Diaz et al. [3] note that, from the various blockchain hurdles, the privacy issue might be one of the most challenging. The privacy of all nodes in the network must be respected by restricting access to certain data. The nodes themselves should be able to determine which information can be accessed by who and what information should be anonymized. The importance of privacy should not be overlooked. Therefore, it is recommended that these privacy requirements are explicitly modeled when visualizing blockchain in SCM. Ben Hamadi et al. [5] proposed to extend the i* framework by adding the following concepts: *access control*, *privacy accountability*, *confidentiality* and *anonymity* but did not implement it, this is done in this paper.

Next to the privacy issues, I1 also stresses the importance of regulations. As blockchain is still a relatively young technology, new regulations that limit the working of the blockchain and/or smart contracts might become applicable. The legal binding of smart contracts in a court of law is often a subject for debate [2]. I1 specifically refers to the repercussions of the GDPR regulations on blockchain. Under GDPR, personal data should remain within the EU. This imposes restrictions on public blockchains because there is no control on where the nodes are located. I1 mentions that this is less of a problem with private blockchains. Additionally, the ‘right to be forgotten’ conflicts with blockchain as an immutable chain of historical transactions, although this rule lacks a real, strict definition. Currently, a workaround exists whereby personal data is stored ‘off-chain’, outside the blockchain database. A reference and a hash of this data is then registered in the blockchain. However, this destroys the purpose of the blockchain, since transparency is diminished, data-ownership becomes vague, one need to find a new way to integrate data from other participants and the data is more vulnerable to cyber-attacks. Siena et al. [19] and Siena, Perini, Susi, & Mylopoulos [20] have introduced an i* extension to model laws and norms. [19] revolves around the application of such extensions specifically for European food traceability systems. This is particularly interesting for blockchain in SCM, as it is important for system developers to understand how the

blockchain should be compliant with which regulations. Because blockchain is a technology which steadily becomes more widespread in the IT-landscape, new regulations will emerge to control it legally.

Figure 2 provides an overview of all suggested extensions, including their proposed graphical notations. The i* extensions for privacy concepts and for laws and norms are taken over from the literature.





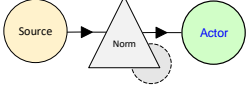
Concept	Graphical notation	Description	References
Privacy			
Access control		Access to data in the chain is restricted to certain nodes. This notation can be used on data elements. The annotation allows to specify who has access or who doesn't.	Ben Hamadi (2020)
Privacy accountability		The notation is used on a data element and allows to make third parties accountable for data manipulation under privacy requirements.	Ben Hamadi (2020)
Confidentiality		The data-owner can hide certain data from the other nodes. This notation can be used on data elements.	Ben Hamadi (2020)
Anonymity		An actor wants to anonymize his data partially or completely. The notation is used on an actor element and allows to specify what data should be anonymized.	Ben Hamadi (2020)
Laws and norms			
Norms		An actor should be compliant with a certain norm. This norm also has a source (e.g. EU).	Siena et al. (2008, 2009)

Fig. 2. Proposed Extensions of i* for BOSE.

4.2 Strategic Dependency Diagram – Farm-to-Fork with Blockchain

It should be apparent that, in case of a blockchain adoption for the Farm-to-Fork process, all actors will become connected to each other through the blockchain system. To visualize such a process, the blockchain system itself should also be represented as an actor, alongside the other participants in the network. The relationship between the nodes in the SC and the blockchain is indeed a dependent one. The blockchain network depends on the farmer, the catcher, the transporters, the butcher, the packager and the supermarket for data. The data is validated and saved into the blockchain. Certain nodes, like the transporter, can depend on the use of IoT devices to automatically capture and send data to the blockchain. For the transporter, this data can include the transportation conditions such as the humidity and the temperature. Based on this data, the blockchain system can also verify whether the contractual terms are fulfilled. The (execution of the) smart contracts therefore depend(s) on the data in the blockchain database. The system can automatically execute the contract through these smart contracts. Because these smart contracts depend on the input data of the SC nodes in the blockchain database, they are also represented as an actor.

Additionally, the blockchain depends on the supermarket to specify the attributes that must be collected by the various stakeholders as input for the blockchain database.

On the other hand, the supermarket node also depends on the blockchain data itself, to permit an analysis of the optimal quality requirements (via business intelligence techniques on this data). After the optimal conditions are estimated by the supermarket, the smart contracts need this list of quality requirements to adjust the contract specifications. Moreover, consumers can check the product's history and origin by (partially) viewing the blockchain's data. The SD model of such a SC process is shown in Fig. 3.

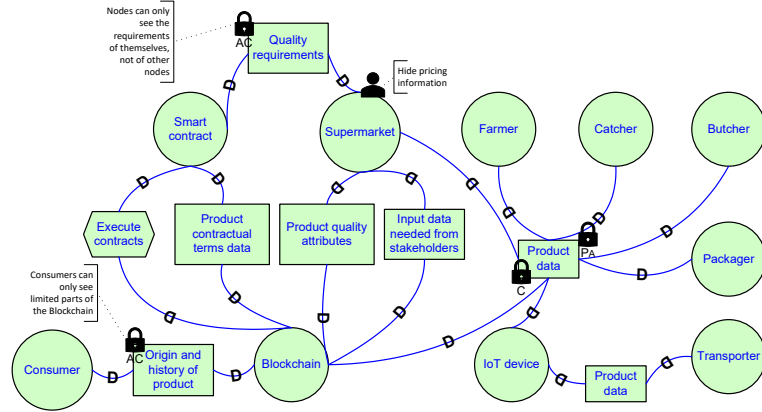


Fig. 3. Farm-to-Fork Blockchain SD Diagram using Blockchain and Smart Contracts.

Figure 3 also contains the extensions for privacy concepts. Consumers are only allowed to see a limited part of the blockchain data and process. Next, the quality requirements imposed by the supermarket are only distributed to the relevant nodes, depending on their respective responsibilities within the overall process. The supermarket can also hide its own price data, since this is classified as sensitive information. All nodes can specify what data they want to hide from other nodes, and third parties should be held accountable when given access to manipulate this data.

4.3 Strategic Rationale Diagram – Farm-to-Fork with Blockchain

The SR model focuses more on the internal rationale or reasoning of all the nodes, related to the dependencies between actors.

In addition to the interaction between the different SC participants, the supermarket's ability to specify the quality requirements for each stakeholder is also important and is therefore depicted with the SR model in Figure 4. This model focuses on the interdependencies between the supermarket, the blockchain, the smart contracts and the consumer. The supermarket is an especially important node as the final product arrives here and is sold to consumers. Hence, the chicken meat must be of the best quality in order to sell it to consumers. It is likely that most benefits of the blockchain adoption are experienced in this stage of the SC: no more wastage because of higher quality and avoidance of contaminated products, contaminated products can no longer get into the hands of consumers which limits health risks, and consumer awareness is higher because they can scan the QR-code on the packaging of the chicken meat to check the history of the product. Given these four important actors (the supermarket,

to show the possible actions of the smart contracts (i.e. checking whether contractual terms are fulfilled or not and automatically executing the smart contracts). Moreover, consumers can check the history and origin of products by scanning a QR-code.

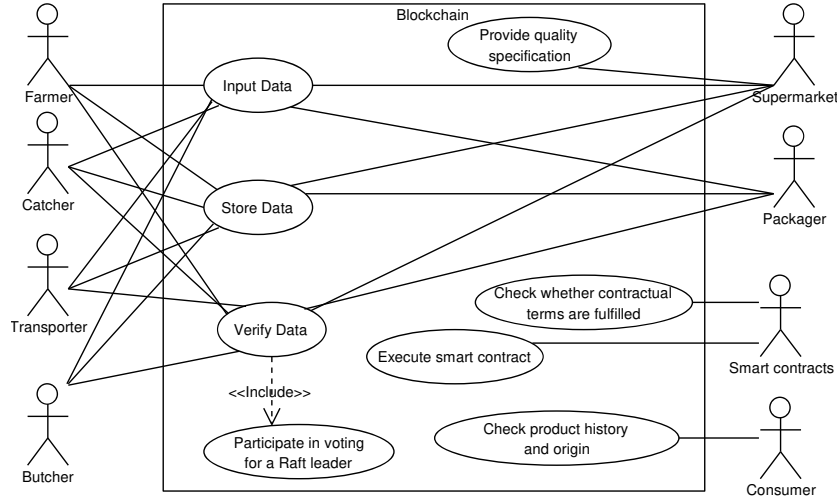


Fig. 5. Farm-to-Fork Blockchain Use Case Diagram.

5.2 UML Sequence Diagram – Farm-to-Fork

The Sequence Diagram is modeled because it shows the order in which the activities occur. As already mentioned previously, this is especially useful for SC processes. The Farm-to-Fork Sequence Diagram is depicted in Figure 6.

With every blockchain return message ‘Verification of data’, an alternative fragment should take place, which defines what happens if the verification is successful and what happens if it isn’t. However, for simplicity reasons in Figure 6, this alternative (or alt-) fragment is only explicitly modeled for the first occurrence where the blockchain wants to verify data (i.e. at the farmer’s data entry). Thus, although not explicitly modeled, this alt-fragment takes place every time the blockchain wants to verify data.

As mentioned before, the transporter can use an IoT device to automatically save and send transportation data to the blockchain. This proposed IoT device is also included in the Sequence Diagram to show the effects of the implementation. Finally, at the bottom, a loop is included. This represents the quality requirement updates that the supermarket can repeatedly implement whenever a new (local or global) optimum is found for the process conditions (e.g. by using business intelligence tools). More illustrations of sequence diagrams in the context of blockchain use cases (on the raft consensus and customer access) can be found in Appendix III.

6 Evaluation of i* and UML for modeling BOSE in SCM

This section compares the pros and cons of the SD and SR models (i* framework) versus the Use Cases and Sequence Diagrams (UML) as modeling techniques for BOSE.

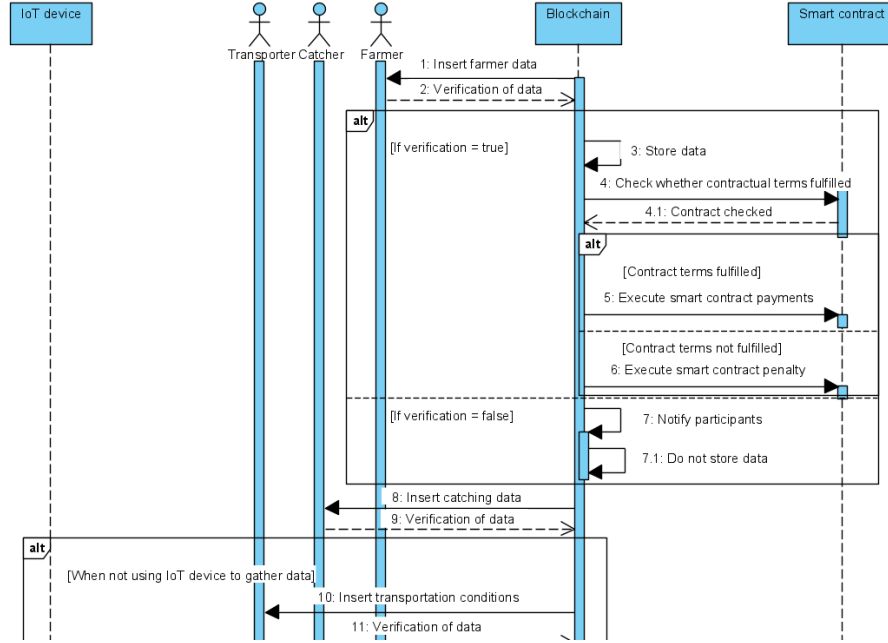


Fig. 6. Farm-to-Fork Blockchain Sequence Diagram (see the Appendix for complete version).

First, a set of the criteria to compare both models is defined. The criteria are based on three intakes: *generic modeling criteria based on existing literature* [7,16]; *blockchain-specific criteria defined in consultation with blockchain expert* (I2), and *other criteria based on findings from applying both modeling techniques*. Next, both frameworks are evaluated using these criteria. Detailed discussions on each criterion of both languages can be found in the Appendix IV. Table 2 provides a final evaluation.

As can be seen in Table 2, the two frameworks distinguish themselves by their different purpose: while i* is social-focused, the UML Use Case and Sequence Diagrams are system-oriented. Both techniques are complementary. Therefore, we recommend to use the i* framework during the early phases of RE. This enables system developers to understand the ‘why’ of the SC process, giving a clear overview of the interdependencies and the goals of all nodes in the blockchain network, as this is the core of the blockchain’s decentralized system. During later phases, UML diagrams can be applied to design the system’s interactions with the different actors of the SC in more detail.

7 Conclusion

After applying both modeling languages to the case, a comparative evaluation between both approaches was performed. A set of assessment criteria was established and we conclude on the complementarity of the approaches. The in-depth comparison between both has revealed that they also lack some elements to model BOSE in SCM to its full extent. Hence, new graphical concepts are proposed to enhance the models. First, in line

Table 2. Comparison of i* and UML for modeling BOSE in SCM.

Criteria	Description	i* UML	
Generic			
Coverage of elements	Whether certain things are difficult or impossible to express. This is about the completeness of the available elements.	-	+
Reusability	Whether models can be reused in a different context.	+	+
Guidelines and tool-support	Whether clear guidelines and tools for the model are available.	-	+
Widespread in different areas	Whether the modeling technique is standardized and generally adopted.	-	+
Expert opinions			
Restricting access and privacy concepts	Whether the model can include privacy concerns.	-	-
Scalability	Whether the model is scalable.	-	+
Ability to express workflow patterns	Whether a flow or structure can be defined in the model.	-	+
Norms	Whether the model can include the compliance of norms and regulations.	-	-
Other Criteria			
Social focus	Whether the model can represent the actor's intentions and internal reasoning.	+	-
Dual granularity	Whether the model allows for both a high-level and a more detailed view of the system.	+	+
Flexibility in modeling	Whether the model is not 'deterministic' but allows to model different scenarios.	+	-
Technical concepts	Whether the model has notations to introduce technical concepts.	-	+

with Ben Hamadi et al. [5], this paper recommends the inclusion of privacy concepts. Next, because of the importance of laws and upcoming regulations that will determine the future direction of BT, the enhancement of the i* framework for laws and norms is also recommended.

The present study combined with Ben Hamadi et al. lead us conclude that i* and its refinements are relevant for each BOSE development in SCM. Future work includes (i) the application of the enhances i* framework in other domains, (ii) the application of other frameworks line notable the business use case model together with BPMN (see [27]) and (iii) the development of transformation (forward engineering) rules to support the blockchain implementation with object and agent technology.

References

1. Apte, S., Petrovsky, N.: Will blockchain technology revolutionize excipient supply chain mgmt.? *Journal of Excipients and Food Chemicals* **7**(3), 910 (2016)
2. Bashir, I.: *Mastering blockchain*. Packt Publishing Ltd (2017)
3. Bettín-Díaz, R., Rojas, A.E., Mejía-Moncayo, C.: Methodological approach to the definition of a blockchain system for the food industry supply chain traceability. In: *Intl. Conf. on Computational Science and Its Applications*. pp. 19–33. Springer (2018)
4. Casado-Vara, R., Prieto, J., De la Prieta, F., Corchado, J.M.: How blockchain improves the supply chain: Case study alimentary supply chain. vol. 134, pp. 393–398. Elsevier (2018)

5. Hamadi, Y.B., Heng, S., Wautelet, Y.: Using i*-based organizational modeling to support blockchain-oriented software engineering: Case study in supply chain mgmt. In: The Intl. Research & Innovation Forum. Springer (2020)
6. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. *MIS Q.* **28**(1), 75–105 (2004)
7. Kelemen, Z.D., Kusters, R., Trienekens, J., Balla, K.: Selecting a process modeling language for process based unification of multiple standards and models. Tech. rep. (2013)
8. Kolp, M., Wautelet, Y., Faulkner, S.: Sociocentric design of multi-agent architectures. In: Social Modeling for Requirements Engineering. MIT Press (2011)
9. Kolp, M., Wautelet, Y.: Human organizational patterns applied to collaborative learning software systems. *Computers in Human Behavior* **51**, 742–751 (2015)
10. Kshetri, N.: 1 blockchain’s roles in meeting key supply chain mgmt. objectives. *Intl. Journal of Information Mgmt.* **39**, 80–89 (2018)
11. Marchesi, M., Marchesi, L., Tonelli, R.: An agile software engineering method to design blockchain applications pp. 1–8 (2018)
12. Niranjnamurthy, M., Nithya, B., Jagannatha, S.: Analysis of blockchain technology: pros, cons and swot. *Cluster Computing* **22**(6), 14743–14757 (2019)
13. OMG: Omg unified modeling language (omg uml). version 2.5.1. Tech. rep. (2017)
14. Rao, N.: The time is now. *Quality Progress* **51**(10), 18–23 (2018)
15. Rocha, H., Ducasse, S.: Preliminary steps towards modeling blockchain oriented software. In: WETSEB2018. pp. 52–57. IEEE (2018)
16. Ruiz, F., van Harmelen, F., Aben, M., van de Plassche, J.: Evaluating a formal modelling language. In: EKAW1994. pp. 26–45. Springer (1994)
17. Runeson, P., Host, M., Rainer, A., Regnell, B.: Case study research in software engineering: Guidelines and examples. John Wiley & Sons (2012)
18. Saberi, S., Kouhizadeh, M., Sarkis, J., Shen, L.: Blockchain technology and its relationships to sustainable supply chain mgmt. *Intl. J. of Production Research* **57**(7), 2117–2135 (2019)
19. Siena, A., Maiden, N., Lockerbie, J., Karlsen, K., Perini, A., Susi, A.: Exploring the effectiveness of normative i* modelling: Results from a case study on food chain traceability. In: CAiSE2018. pp. 182–196. Springer (2008)
20. Siena, A., Mylopoulos, J., Perini, A., Susi, A.: Designing law-compliant software requirements. In: International Conference on Conceptual Modeling. pp. 472–486. Springer (2009)
21. Wautelet, Y.: Representing, modeling and engineering a collaborative supply chain management platform. *Intl. J. of Info. Systems and Supply Chain Mgmt.* **5**(3), 1–23 (2012)
22. Wautelet, Y.: A model-driven IT governance process based on the strategic impact evaluation of services. *J. Syst. Softw.* **149**, 462–475 (2019)
23. Wautelet, Y., Heng, S., Kolp, M., Penserini, L., Poelmans, S.: Designing an mooc as an agent-platform aggregating heterogeneous virtual learning environments. *Behaviour & Information Technology* **35**(11), 980–997 (2016)
24. Wautelet, Y., Kolp, M.: Business and model-driven development of BDI multi-agent systems. *Neurocomputing* **182**, 304–321 (2016)
25. Wautelet, Y., Kolp, M., Heng, S., Poelmans, S.: Developing a multi-agent platform supporting patient hospital stays following a socio-technical approach: Mgmt. and governance benefits. *Telematics and Informatics* **35**(4), 854–882 (2018)
26. Wautelet, Y., Kolp, M., Penserini, L.: Service-driven iterative software project mgmt. with i-tropos. *J. UCS* **24**(7), 975–1011 (2018)
27. Wautelet, Y., Poelmans, S.: An integrated enterprise modeling framework using the RUP/UML business use-case model and BPMN. In: The Practice of Enterprise Modeling PoEM 2017, Leuven, Belgium, Proceedings. LNBIP, vol. 305, pp. 299–315. Springer (2017)
28. Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J.: Social Modeling for Requirements Engineering. MIT Press (2011)