12-2021

SimChain: Simulator for Supply Chain Decision Making with Blockchain

Nicola Ibba

Troy Wiipongwii

Ting-Ting (Rachel) Chung

SimChain: Simulator for Supply Chain Decision Making with Blockchain

Prototype Demonstration

Nicola Ibba
William & Mary
nibba@wm.edu

Troy Wiipongwii
William & Mary
ttwiipongwii@wm.edu

Tingting (Rachel) Chung
William & Mary
rachel.chung@mason.wm.edu

Yu Xia
William & Mary
amy.xia@mason.wm.edu

Abstract

The deployment of distributed ledger technologies (DLT) and, more specifically, blockchain, has promised to transform supply chain management. However, there has been relatively little investigation on visualization of blockchain data that may impact supply chain decision making. SimChain is a web-based simulation of a simple supply chain with or without the integration of blockchain technologies. Blockchain offers potential opportunities to improve supply chain decision making, and SimChain is designed to facilitate empirical examinations of such benefits. Specifically, SimChain is highly configurable, allowing administrators to configure nearly 50 different supply chain settings and environmental variables to simulate event-based interruptions to the supply chain. SimChain is built using the Django framework for web deployment, and Python scripts and MySQL for data simulation.

Keywords

Blockchain, Supply Chain Management, Simulation, Decision-Making, Visualization.

Introduction

With supply chain becoming increasingly integrated and complex, decision makers need access to high-quality data. Their decisions may critically impact supply chain resilience, security, integrity, and trust (Lee et al. 1997; Pastore et al. 2020). The deployment of distributed ledger technologies (DLT) and, more specifically, blockchain, has been predicted to transform supply chain management (Chang and Chen 2020; Hastig and Sodhi 2020). Specifically, blockchain can significantly reduce long-term transaction costs and substantial improve supply chain security and trust.

While the benefits of blockchain for supply chain management are widely discussed in terms of security and trust, its effectiveness in improving supply chain decision making has received little empirical examination. Blockchain produces large quantities of transactional data. Visualization of blockchain-based data can potentially improve decision-making significantly. However, a comprehensive review of visualization tools of blockchain data reveals that most visualizations lack contextual details (Tovanich et al. 2021), which can limit their value for managerial decision making. Most importantly, to our best knowledge, no study has yet to examine the impact of blockchain data on supply chain decision making.

With SimChain, we create a tool that allows researchers as well as managers to examine how visualization of blockchain-based data impacts supply chain decision making, in terms of both its process and outcomes. SimChain consists of a highly configurable Python program that simulates transactional data for a four-echelon supply chain with manufacturer, wholesaler, retailer, and consumers. It is designed to allow the adoption of one or multiple inventory ordering systems as well as various production schedules. The output of the simulator program contains a set of ledgers with detailed daily inventory information and financial records. Human decision makers interact with the program through its web-based interface The program tracks user behaviors and decisions which can then be used for empirical analyses of decision quality.

Blockchain for Supply Chain Management

Blockchain is a decentralized data-recording technology that can be used in finance, supply chain and logistics (Chang and Chen 2020). The number of protocols and value propositions of blockchain have been developing rapidly. Most of the blockchain literature focuses on Bitcoin and Ethereum protocols, which are applicable to the finance sector. Other blockchain protocols, such as IoTA, use technologies that depart from the traditional blockchain data structure. These protocols are promising for reducing transaction times, and improving decentralization, for supply chain management, especially during crises that disrupt the supply chain (Min 2019). In addition to generating protocol-level data, these tools also enable rich data marketplaces which support the sharing and selling of any type of data imaginable. Some data shared on these marketplaces include weather data, air quality data, and other types of real-time sensor data[1].

The underlying protocol and data structure of the blockchain create parameters for the data generated. For instance, bitcoin protocol-generated data include blocks, transactions, timestamps, hash rates, prices, miners, transaction volume, total addresses, and active addresses. Such data can then be converted to decision-support information such as organizational relationships, ownership roles, among others. This information can also support managerial analysis and decision making, including fraud prevention, detection, and investigation (Hastig and Sodhi 2020). In addition, these protocols also allow the creation of smart contracts, or programs with predetermined rules that apply to transaction records, including their creators, source code, and contract events (Wang et al. 2019). Therefore, in addition to the protocol-level data derived from the blockchain, smart contract event data can provide information on transactions. When incorporated into supply chains, these protocols can generate and record important information on IoT sensors, and possibly provide the digital equivalent to real-life events happening outside of the chain.

Tools for visualizing blockchain data, commonly known as blockchain explorers, can provide different levels of abstractions of transaction data depending on access rights, whether the blockchain is public or private, and whether the data is stored on-chain versus off-chain (Tovanich et al. 2021). Our SimChain prototype not only simulates blockchain data, but also supply-chain activities anchored to the blockchain.

Next we describe the technical architecture of SimChain, its key design elements, functionalities, and one use case.

Method

SimChain[2] is a web-based application based the Django framework[3], a robust web framework in Python. The application simulates a simple four-echelon supply chain with manufacturer, wholesaler, retailer, and consumers, and generates inventory and orders transactions for each echelon using a time-discrete approach. The simulation uses classes and objects to generate transaction data, which are stored in a MySQL relational database in the backend. The user interface renders data visualizations using a combination of Django method and JavaScript. The user interface allows a player to create an account and log in to perform decision making tasks. The log-in process creates a user session that records user behavior data into the database.

The current version of SimChain has a single-agent design for one human player to assume a managerial role in the retail node. The administrator can configure various production schedules for the simulation. Based on the production schedule, the simulator pre-generates simulation retailer data using algorithmic forecasting for the production period before the player begins. In other words, the simulation operates

[1] https://data.iota.org/#/

[2] Trial access may be available upon request. Please contract the authors for access.

[3] https://www.djangoproject.com/

without a human player. Before the player joins, retail order quantities for the following period are estimated using exponential smoothing algorithms and historical market demand data. When the player joins the simulation and assumes the retailer role, the supply chain pipeline is already operational. Algorithmic forecasting for the retailer role stops when the human player takes over.

SimChain is designed to configure as a classic ERP with low integration, or alternatively with a blockchain system. This design allows the administrator to conduct experiments to compare decision quality outcomes between configurations with versus without blockchain.

An important feature of SimChain is the ability for the administrator to configure event-based supply chain disruptions. This feature is important, because it allows the administrator to evaluate potential impacts of such disruptions, and the role of blockchain data visualization in mitigating such impacts. Disruptions to the supply chain are modeled in two ways: weather and infestation. The administrator can specify the distribution of temperature data. Infestation, as a proxy for quality conditions, can be specified as a function of weather conditions.

Weather can be dry or rainy, and at different temperatures. Weather is locally generated, meaning that each node will have a different weather outcome; the correlation of the weather between locations can be adjusted, making the weather more, or less, homogeneous.

Infestations happen in the warehouses and are directly correlated with the weather; wet and hot weather favor infestations, increasing the likelihood of one to happen. Similarly, infestations affect the product hosted in the warehouse, increasing the odds of it becoming unusable.

SimChain is turn-based, and data is generated in the background between turns. Time is measured in days; a turn consists of one or more days (preset to 7 days), based on the inventory review period set by the administrator. For each day of a run, SimChain records player data, including order quantity and other supply chain decisions, in the database, as well as player responses to custom questions at each turn. The administrator can add questions to have feedback about the effectiveness / utility of the data presented, trust in the source and so forth.

SimChain uses the same strategy to generate blockchain-specific data. Assuming a permission-based (i.e., private) blockchain design, SimChain generates a single set of data, completely off-chain, and then use rendering strategies to present the data in a variety of formats.

Administrator variables

SimChain allows the administrator to reconfigure the simulator based on alternative designs. The administrator can use a panel of simulation and environmental variables to configure simulation settings. Those variables are accessible from an administrator dashboard that is available with the Django framework. All settings are available to both the blockchain and conventional versions of the simulation. See Table 1 for a list of main variables that can be configured by the administrator.

Category	**Variable**
Simulation Configuration	Duration, player start point
Supply Chain Configuration	Number of factories, number of distributors, number of retailers
Factories	Daily production rate, lead time, initial inventory quantity
Assemblers	Daily production rate, lead time
Retailers	Service Level, Alpha, Initial Mean Demand, Review Period, Initial Cash, Unitary product price and cost, Unitary holding cost, order cost, backlog cost, return fee, Quality Assurance testing fee, cost of information requests.
Market Demand	Daily mean demand and standard deviation
Weather	Start of hot season, duration of hot season, mean temperature hot / cold season, standard deviation of temperature low / hot season, Markov-chain probability matrix for next day weather condition (high/low season)

Infestations	Probability of infestations with each weather combination (hot/dry, hot/wet, mild/dry, mild/wet), minimum duration of an infestation

Table 1. List of main administrator variables

We also plan to add options for the administrator to easily switch between inventory management policies, which are currently hard-coded in the simulation. We will also allow the administrator to create setting profiles, and to have the ability to easily configure a new experiment by modifying settings (e.g., weather) of an existing profile.

Use Case

Here we describe a use case that involves a supply chain of two factories, one wholesaler, one retailer, and multiple consumers, illustrated in Figure 1.

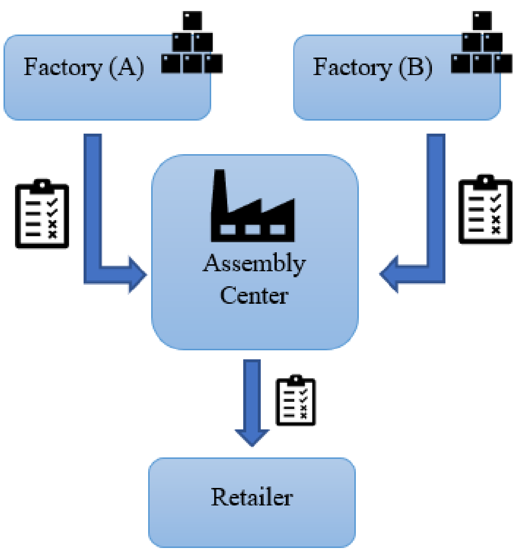


Figure 1. Supply Chain Use Case

The factories produce at a constant daily rate and have infinite storage. The assembly center buys components from the factories following a fixed-time re-ordering policy. The retailer purchases from the assembly center. The retailer's demand is driven by the market demand from the consumers, which is generated at each turn of the simulation based on the administrator's settings.

An overview of the supply chain configuration is provided in the beginning of the simulation. The player is assumed to have some knowledge of supply chain management. During the simulation, the player has access to a wide array of information, with focus on the retailer's inventory and market demand. When the simulation begins, the player is presented with a data dashboard that allows navigating multiple aspects of the inventory available, past orders, and environmental information such the weather. The overview tab, illustrated in Figure 2, shows trends in the market demand, as well as past order amount, sales, and cash flow. The dashboard also has a tab dedicated to inventory data; data available here include daily demand, filled order, backorders, incoming orders, replenish amount, on-hand inventory, inventory position. A selection of some or all data is presented to the player, depending on administrator settings. Figure 3 illustrates tabular representation of data on blockchain-like transactions.

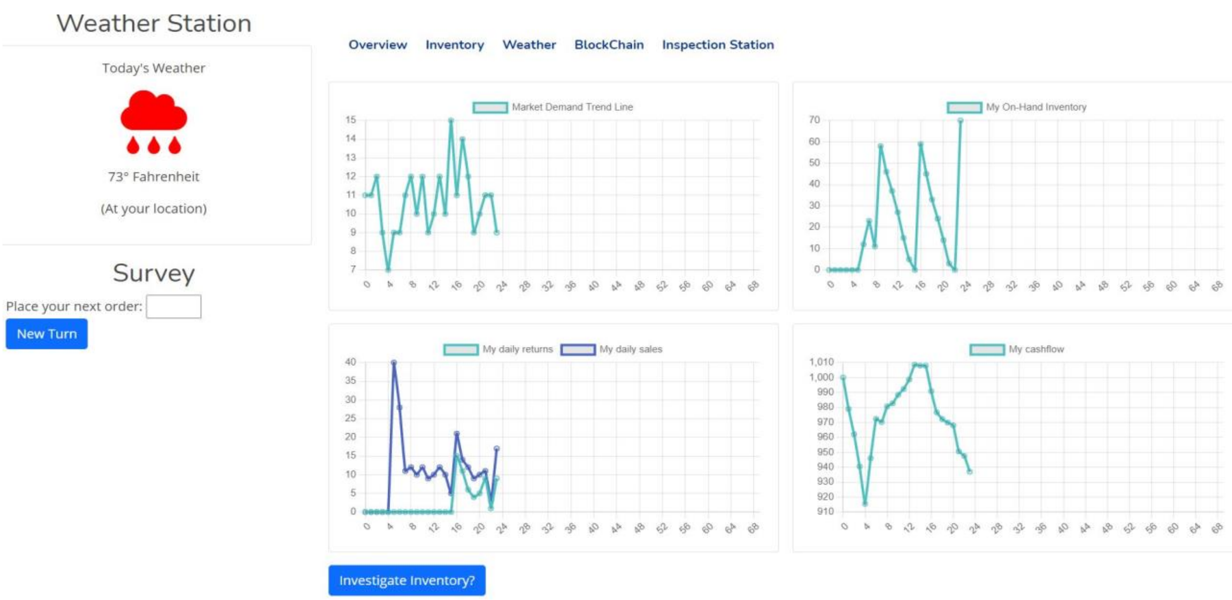
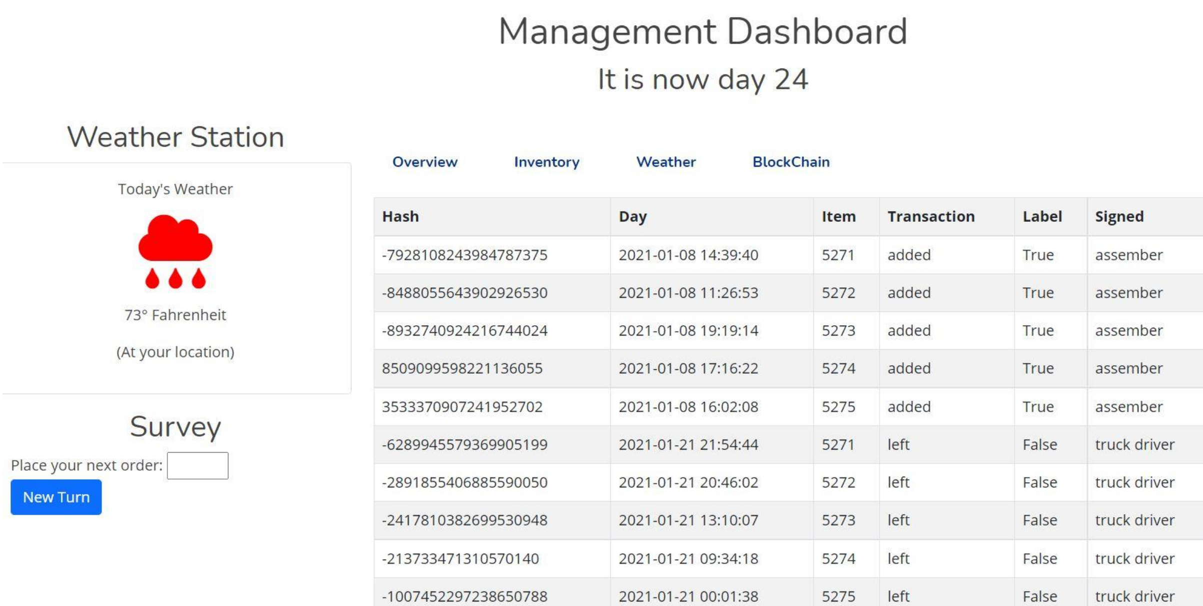


Figure 2. A snapshot of the dashboard



Hash	Day	Item	Transaction	Label	Signed
-7928108243984787375	2021-01-08 14:39:40	5271	added	True	assember
-8488055643902926530	2021-01-08 11:26:53	5272	added	True	assember
-8932740924216744024	2021-01-08 19:19:14	5273	added	True	assember
8509099598221136055	2021-01-08 17:16:22	5274	added	True	assember
3533370907241952702	2021-01-08 16:02:08	5275	added	True	assember
-6289945579369905199	2021-01-21 21:54:44	5271	left	False	truck driver
-2891855406885590050	2021-01-21 20:46:02	5272	left	False	truck driver
-2417810382699530948	2021-01-21 13:10:07	5273	left	False	truck driver
-213733471310570140	2021-01-21 09:34:18	5274	left	False	truck driver
-1007452297238650788	2021-01-21 00:01:38	5275	left	False	truck driver

Figure 3. Tabular representation of blockchain-like transactions.

The player is asked to complete two tasks: decide on the quantity to order for the next time period, and decide merchandise recalls or returns. Based on the player's decisions, SimChain keeps track of player performance in terms of cash-on-hand.

The player is given the opportunity to make informed decisions, but also to "jump to conclusions", for example by taking the chance of returning an item without prior asking (or visualizing) data and details about past recorded transactions related to that item. Indeed, players who do not (yet) have access to the blockchain-data dashboard, are invited to request more information from their suppliers. Upon request, SimChain reveals more details about a set of items; the cost of requesting more information is always counter-balanced by the risk of losing or paying a different price (e.g., an unhappy customer returns a defective item).

On the other hand, blockchain adds different types of information and increases the visibility within the supply chain. In the blockchain condition, the player is shown production schedules or inventory levels of other nodes, as well as reveals environmental variables information, which is assumed to be stored in the blockchain through IoT sensors.

Our current focus is on simulating and visualizing smart-contract information, signatures, and wallet addresses. We also leverage the use of timestamps and simulated intra-day events (like logging on an infestation journal / report). See Figure 4 for an illustration of timestamps presented in a blockchain-like visualization.

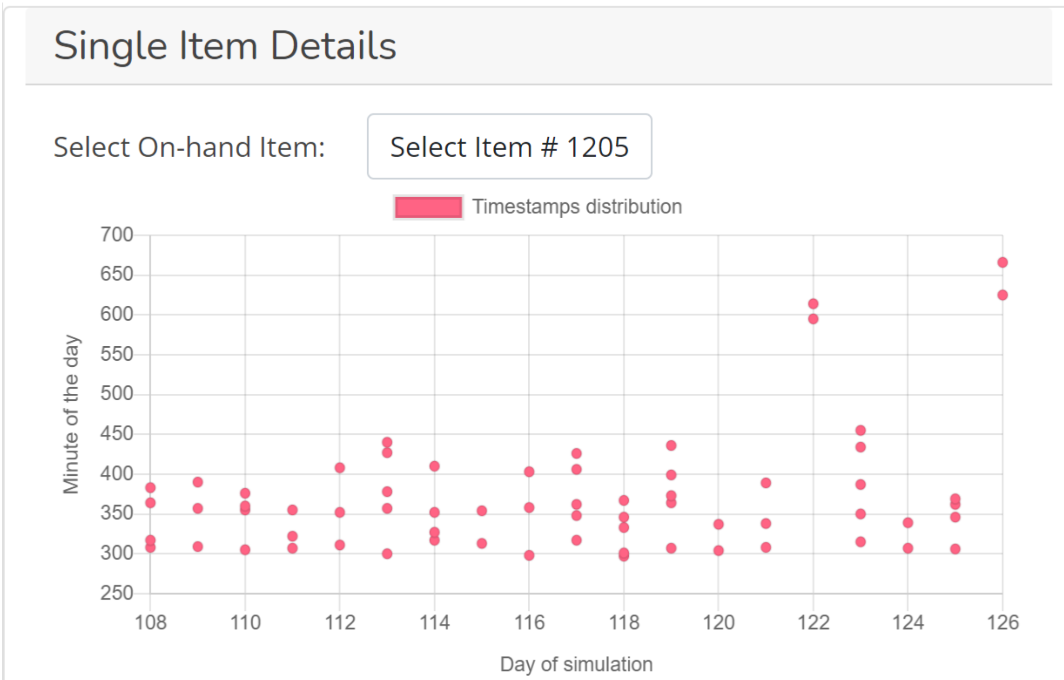


Figure 4. Distribution of timestamps in a blockchain-like visualization.

Conclusion

While value propositions of blockchain integration in supply chain management have been explored in the current literature, there has been relatively little empirical investigation on how visualization of blockchain data may improve supply chain decision making. We have developed SimChain, a flexible web application powered by a Python simulation program, which can be used for both educational and research purposes. It facilitates the study of supply chain decision making, with or without blockchain, and with or without event-based interruptions. With a wide array of configurable options, SimChain enables the administrator to have complete control over the simulated environment and to explore alternative scenarios in an efficient manner.

Acknowledgement

We would like to thank the Blockchain Lab of William & Mary's Global Research Institute for financial support, and Baxter Difabrizio, Tarek Elamine, Kristina Posner, and Pratyush Sharma for their valuable feedback and research support.

References

Chang, S. E., and Chen, Y. 2020. "When Blockchain Meets Supply Chain: A Systematic Literature Review on Current Development and Potential Applications," *IEEE Access* (8), pp. 62478–62494. (https://doi.org/10.1109/ACCESS.2020.2983601).

Hastig, G. M., and Sodhi, M. S. 2020. "Blockchain for Supply Chain Traceability: Business Requirements and Critical Success Factors," *Production and Operations Management* (29:4), pp. 935–954. (https://doi.org/10.1111/poms.13147).

Lee, H. L., Padmanabhan, V., and Whang, S. 1997. "The Bullwhip Effect in Supply Chains," *Sloan Management Review* (38:3), Cambridge, United States: Massachusetts Institute of Technology, Cambridge, MA, pp. 93–102.

Min, H. 2019. "Blockchain Technology for Enhancing Supply Chain Resilience," *Business Horizons* (62:1), pp. 35–45. (https://doi.org/10.1016/j.bushor.2018.08.012).

Pastore, E., Alfieri, A., Zotteri, G., and Boylan, J. E. 2020. "The Impact of Demand Parameter Uncertainty on the Bullwhip Effect," *European Journal of Operational Research* (283:1), pp. 94–107. (https://doi.org/10.1016/j.ejor.2019.10.031).

Tovanich, N., Heulot, N., Fekete, J.-D., and Isenberg, P. 2021. "Visualization of Blockchain Data: A Systematic Review," *IEEE Transactions on Visualization and Computer Graphics* (27:7), pp. 3135–3152. (https://doi.org/10.1109/TVCG.2019.2963018).

Wang, S., Ouyang, L., Yuan, Y., Ni, X., Han, X., and Wang, F.-Y. 2019. "Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends," *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (49:11), pp. 2266–2277. (https://doi.org/10.1109/TSMC.2019.2895123).