Elektrotech. Inftech.
https://doi.org/10.1007/s00502-025-01321-5

e+i elektrotechnik und informationstechnik

Supply chain optimization using model-based systems engineering and the Internet of Things

Amirali Amiri · Stefan Nastic · Wolfgang Kastner

Received: 24 February 2025 / Accepted: 24 March 2025

Abstract Supply Chain Management (SCM) is crucial for business operations, requiring careful planning and optimization to minimize product costs. New technologies like the Internet of Things (IoT) enable real-time monitoring of supply chain performance, helping detect malfunctions and adapt strategies to address issues like rising costs. However, IoT system architects face challenges in mastering diverse technologies and integrating them into SCM. Methodologies like Model-Based Systems Engineering (MBSE) simplify the design process by providing system models and views. In this paper, we propose an MBSE approach for IoT-driven supply chain optimization. Our approach automatically performs optimization based on IoT monitoring data. We illustrate how system designers can adapt our approach to their specific needs. For example, system designers can provide templates for the artifact generation, such as system code or input files for the optimization. We discuss multiple scenarios showcasing the benefits of our approach.

Keywords Supply chain optimization · Internet of Things · Model-Based Systems Engineering

Lieferkettenoptimierung durch modellbasiertes Systems Engineering und das Internet der Dinge

Zusammenfassung Das Supply Chain Management (SCM) ist entscheidend für den Geschäftsbetrieb und erfordert eine sorgfältige Planung und Optimierung, um Produktkosten zu minimieren. Neue Technologien wie das Internet der Dinge (IoT) ermöglichen eine Echtzeitüberwachung der Lieferkettenleistung, wodurch Fehlfunktionen erkannt und Strategien zur Bewältigung steigender Kosten angepasst werden können. Allerdings stehen IoT-Systemarchitekt:innen vor der Herausforderung, verschiedene Technologien zu beherrschen und sie in das SCM zu integrieren. Methoden wie Model-Based Systems Engineering (MBSE) vereinfachen den Designprozess, indem sie Systemmodelle und -ansichten bereitstellen. In dieser Arbeit schlagen wir einen MBSE-Ansatz zur IoT-gestützten Optimierung von Lieferketten vor. Unser Ansatz führt automatisch eine Optimierung basierend auf IoT-Überwachungsdaten durch. Wir zeigen, wie Systementwickler:innen unseren Ansatz an ihre spezifischen Anforderungen anpassen können, beispielsweise durch die Bereitstellung von Vorlagen für die Artefakterstellung wie Systemcode oder Eingabedateien für die Optimierung. Wir diskutieren verschiedene Szenarien, die die Vorteile unseres Ansatzes veranschaulichen.

Schlüsselwörter Lieferkettenoptimierung · Internet der Dinge · Modellbasiertes Systems Engineering

1 Introduction

Supply-Chain Management (SCM) plays a vital role in many businesses. The SCM is a complicated process that requires careful planing and optimization to reduce total costs of finished products [2, 4]. New tech-

A. Amiri (✉) · W. Kastner
Automation Systems Group, Institute of Computer Engineering, TU Wien, Treitlstraße 1–3/4, 1040 Vienna, Austria
amirali.amiri@tuwien.ac.at

S. Nastic
Distributed Systems Group, Institute of Information Systems Engineering, TU Wien, Vienna, Austria

Published online: 29 April 2025

nologies, such as the Internet of Things (IoT), help in monitoring the performance of the supply chain, e.g., by detecting and reporting a malfunction in a facility [12, 16]. Nevertheless, a barrier to entry remains, as IoT system designers must acquire knowledge of various technologies and understand how to implement and integrate them into their SCM systems.

Established methodologies like Model-Based Systems Engineering (MBSE) [8] help streamline the design process by providing abstractions through models and views. An important MBSE view is the optimization of supply chain based on the IoT data. Using these information, the decision makers can adapt and optimize the supply chain to overcome the decrease in performance, e.g., increase of the total costs. Thus, we set out to answer the following research questions:

RQ1: *How can MBSE assist the design of IoT-based supply-chain management systems?*

RQ2: *How can we optimize the supply chain using the IoT monitoring data?*

The contribution of the paper is an approach of model-based systems engineering tailored for supply-chain optimization using IoT devices and their data integration in the cloud. Specifically, we define metadata to describe these systems, which serve as tags within Systems Modeling Language (SysML) 2.0[1] model instances. Our framework performs optimization based on the IoT data. We discuss multiple scenarios and follow a detailed illustrative case, where IoT data used for adaptation yields benefit in supply-chain optimization.

The structure of the paper is as follows. Section 2 lists the related work. Section 3 presents our approach. Section 4 discusses our results. Finally, Sect. 5 concludes our study.

2 State of the art

Adama et al. [3] highlight how digital transformation reshapes supply-chain management through AI, big data, and blockchain, improving efficiency and agility. Their study emphasizes shifts to digital ecosystems, enhancing visibility and automation while addressing challenges like system integration and IoT trends—key focuses of our paper.

Abdul-Azeez et al. [2] emphasize optimizing SCM to improve efficiency, reduce costs, and enhance customer satisfaction. They integrate procurement, manufacturing, logistics, and inventory into SAP S/4HANA, using real-time data, predictive analytics, and IoT to enhance visibility, decision-making, and agility. Similarly, Ikevuje et al. [11] highlight IoT and data analytics in improving supply chain performance, addressing challenges like visibility, delays, and risk management. Their recommendations include investing in IoT infrastructure, ensuring data security, and fostering collaboration. Unlike our study, these works do not adopt a model-based systems engineering approach for system design and automated code generation.

Goli et al. [10] emphasize the importance of supply-chain network design, highlighting rising transportation costs and the need for strategic and tactical decisions. They propose a flexible, IoT-based supply chain integrating logistics across suppliers, producers, distribution centers, customers, and repair centers. Flexibility is enabled through IoT-managed direct and indirect deliveries, modeled using mixed-integer linear programming. Unlike their approach, which requires a learning curve, our study simplifies this process through MBSE.

Vogel-Heuser et al. [18] highlight how the properties and environmental dependencies of industrial systems influence automated production performance. These properties, often in supplier documents like manuals or catalogs, are represented using SysML profiles and disciplinary views, supported by a metamodel for clarity. To address the static nature of profiles, they employ business-process-modeling notation to showcase SysML's dynamic workflow benefits.

De Saqui-Sannes et al. [9] explore the evolution of Systems Engineering from document-centric to model-centric MBSE approaches. While earlier works outline MBSE's benefits and limitations, this paper focuses on providing industry professionals with criteria for selecting MBSE languages, tools, and methods, expanding beyond common techniques like SysML. Kharatyan et al. [13] discuss the impact of digitization on technical systems, such as autonomous driving, emphasizing the challenges posed by increased complexity and interconnectivity. They advocate for MBSE approaches to address these challenges, stressing the need to incorporate security considerations early in system design to ensure reliability.

Like our study, these works provide a system model and an approach to support the system design. However, we focus on the interplay of MBSE and IoT, as well as their benefits in SCM. This integrated view is lacking in the literature.

3 Approach details

The workflow of our MBSE approach is outlined in Fig. 1. Using SysML 2.0's textual representation[2], we define *metadata* specific to supply chain optimization with IoT systems, which serve as tags for creating model instances. The Optimizer leverages linear optimization [15] for supply chain management. The optimized models are passed to an *artifact generator* to

[1] https://www.omg.org/spec/SysML/2.0/Beta1.

[2] https://www.omg.org/spec/SysML/2.0/Beta1.

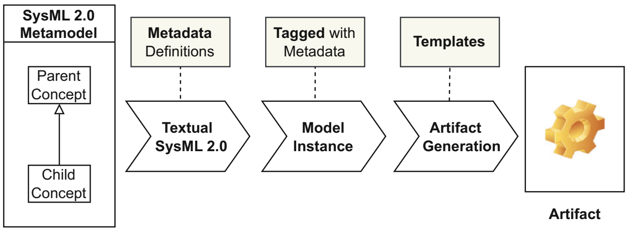

Fig. 1 The Workflow of Model-Based Systems Engineering

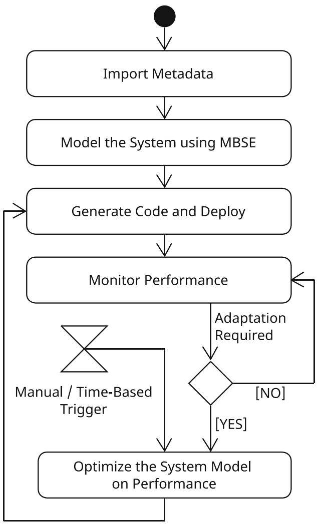

Fig. 2 High-Level Activities of Our Approach

produce outputs such as code, input files for the optimization, e.g., updated shipping costs. The *adaptation* can be triggered automatically in case the supply chain performance deteriorates. Moreover, manual or time-based triggers can start the system model optimization. Figure 2 summarizes the key activities in our approach, with further details provided in later sections. The primary task for architects or designers is tagging system components with our predefined metadata in SysML 2.0. Our system suggests optimization automatically.

Metadata Our metadata definitions for the supply chain optimization using IoT systems are presented in Fig. 3. A *System under Test* includes *Supply Chain Nodes, Infrastructure* and *Communication*. A supply chain node represents a location with supply and demand. These nodes have *Capacity* of production and *Demand* with associated *Costs*. *Infrastructure* represent the underlying IoT system. An *Execution Environment* models either virtual or physical environments, e.g., virtual machines, containers, or bare-metal servers. These environments have *Software* and *Hardware* components. The IoT system consists of *Edge, Fog, Cloud, Message Broker* [14], *Device Gateway*, and *IoT Device* nodes. These node communicate using different patterns of *Communication*. These best-practices are *Data Streaming, Synchronous*, and *Asynchronous*, which can be *Event-Based, Messaging* [7], or *Publish Subscribe* [17]. Figure 4 (Listing 1) provides a snippet of our metadata.

Supply-chain model We model supply chain nodes with specific characteristics, i.e., locations with capacity of production, demands for finished products, and associated costs. The associated costs for each location are as follows. Fixed costs are manufacturing costs, e.g., equipment, utilities, staff, or rental. Variable costs are costs for production of specific goods, e.g., production line costs and raw material. Freight costs model shipping and transport costs, e.g., costs per container of finished products. Figure 5 (Listing 2) shows a snippet of the model for the supply chain management. We have defined a composite part *location* that contains subparts.

We optimize the supply-chain performance using linear optimization [15]. The objective function is to minimize the production costs as the metric of performance. These costs are based on fixed, variable and freight costs, defined by our supply-chain model. The optimization problem can be formulated as follows. The problem states to minimize the total costs, subject to the demands of all locations being fulfilled:

$$f(p,s,d) = fixCost(p,s) + varCost(p,s,d) \qquad (1)$$

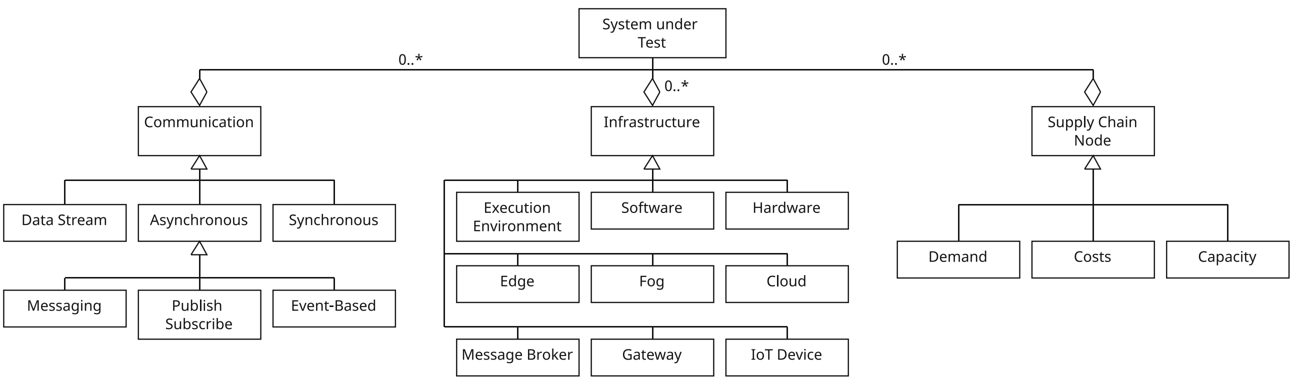

Fig. 3 Metadata of Supply Chain Management Using the Internet of Things

```
package Meta_SupplyChain {

    metadata def SupplyChainNode;
    metadata def Demand :> SupplyChainNode;
    metadata def Capacity :> SupplyChainNode;
    metadata def Costs :> SupplyChainNode;

    metadata def Infrastructure;
    metadata def Edge :> Infrastructure;
    metadata def Fog :> Infrastructure;
    metadata def Cloud :> Infrastructure;
    metadata def IoTDevice :> Infrastructure;
    metadata def Gateway :> Infrastructure;
    metadata def MessageBroker :> Infrastructure;

    metadata def Communication;
    metadata def Asynch :> Communication;
    ...
}
```

Fig. 4 Listing 1: Excerpt of Metadata Definitions in SysML 2.0

```
package SupplyChainModel {
    import Meta_SupplyChain::*;

    part demand {@Demand;}
    part supplyLow {@Capacity;}
    part supplyHigh {@Capacity;}
    part fixedCosts {@Cost;}
    part variableCosts {@Cost;}
    part freightCosts {@Cost;}

    part location {@SupplyChainNode;
      part demand;
      part supplyLow;
      part supplyHigh;
      part fixedCosts;
      part variableCosts;
      part freightCosts;
    }
}
```

Fig. 5 Listing 2: Supply-Chain Model in SysML 2.0

minimize

$$\sum_{p\epsilon products,\ s,d\epsilon locations} f(p,s,d) \tag{2}$$

subject to

$$\forall\ p \in products \tag{3}$$

$$\sum_{s\epsilon locations} capacity(p,s) = \sum_{d\epsilon locations} demand(p,d) \tag{4}$$

in which, p denotes a product, s the source of production, d the destination, *locations* the supply-chain nodes, *fixCost*(*p*,*s*) the manufacturing cost of a product p at source s, *varCost*(*p*,*s*,*d*) the cost to send a product p from source s to destination d (a function of *freight* and *variable costs*), *capacity* (*p*,*d*) the capacity of a product p at source s, and *demand* (*p*,*d*) the demand of a product p at destination d.

Artifact generation We generate artifacts, such as the input costs and the capacity of production for different supply-chain locations based on pre-defined templates. For example, when the IoT devices detect a manufacturing failure resulting in increase of the costs, the artifact generator can produce updated input costs. Our approach automatically suggests adaptation of the supply chain by running the optimization code with the newly-generated artifacts. Another example, where artifact generation can optimize the supply chain is when the shipping costs has increased, e.g., as a result of new types of transportation or routes. In these cases, the artifact generator can produce updated freight costs to optimize the supply chain. Moreover, when the infrastructure changes, the optimization code needs to be updated. An example of this case is to add new constraints to the optimization problem and generate new optimization code. Section 4 provides a sample case illustrating these scenarios. Figure 6 (Listing 3).

4 Discussion

We model an illustrative sample case and study different scenarios of the supply-chain optimization. Figure 5 (Listing 2) showed the supply-chain model. Each of these locations have two sites, i.e., low and high capacity. For the illustrative sample case, we specify this model as a system with five locations, i.e., USA, Germany, Japan, Brazil, and India. The input files specify the costs, the production capacities, and the demand for each location. For simplicity, we study only one product.

Figure 7 (Listing 4) gives a snippet of the template, where we create a linear minimization problem mod-

```
package SCMSampleCase {
    import Meta_SupplyChain::*;

    part USA :> location;
    part Germany :> location;
    part Japan :> location;
    part Brazil :> location;
    part India :> location;
    ...
}
```

Fig. 6 Listing 3: Excerpt of the Sample Case in SysML 2.0

```
from pulp import *

loc = ['USA', 'Germany', 'Japan', 'Brazil', 'India']
size = ['Low', 'High']

# minimization linear problem
model = LpProblem("Capacitated Plant", LpMinimize)

# optimization variables
production = LpVariable.dicts("production_",
      [(i,j) for i in loc for j in loc],
      lowBound=0, upBound=None, cat='continuous')

plant = LpVariable.dicts("plant_",
      [(i,s) for s in size for i in loc],
      cat='Binary')

# objective function: total costs
model += (
  lpSum([fix_cost.loc[i,s] * plant[(i,s)] * 1000
          for s in size for i in loc]) +
  lpSum([var_cost.loc[i,j] * production[(i,j)]
          for i in loc for j in loc]))
...
```

Fig. 7 Listing 4: Excerpt of the Optimization Template in Python eling 1234. The artifact generation is based on an open-source code. The code uses PuLP optimization in Python.

Baseline scenario For our baseline example, we specify variables and solve the optimization problem of costs of finished products. In Fig. 5 (Listing 2), we specified low- and high-capacity sites for each location. We specify the low as 500,000 and high as 1,500,000 capacity of production in units per month (u/m). For the demand, we have 2,800,000 u/m for the USA site, 90,000 for Germany, 1,700,000 for Japan, 145,000 for Brazil, and 160,000 for India. As it can be read, not all locations can provide the demand locally. For example, the USA can produce 2,000,000 u/m in both low- and high-capacity sites. However, the demand is 2,800,000 u/m, which means products must be shipped from overseas. We give costs, i.e., fixed, variable, and freight costs, for each location as input files in our online appendix. For example, the fixed costs, i.e., the manufacturing costs, of the USA location for the low- and high-capacity sites are 6,500,000 and 9,500,000 dollar per month. Running the optimization code with the given values, we have the total cost of 62,038,000 dollar per month.

IoT infrastructure We study multiple scenarios, where the IoT infrastructure monitors the supply chain and triggers optimization. We model multiple IoT devices, e.g., manufacturing and transportation. These devices send monitoring data to a gateway. The gateway sends the data via a message broker to the edge, fog and cloud services for analytics. These analytics can trigger a rerun of the optimization code for the supply chain management. Figure 8 (Listing 5) shows the tagged model instance.

Transportation scenario We model a second scenario. In our model of IoT infrastructure presented by Fig. 8 (Listing 5), we included a transportation device. This device can track freight costs and trigger a rerun of the optimization code if these costs increase. A simple mechanism of calculating freight costs is as follows: These devices update information regrding the routes taken and the duration of transportation to a cloud service. In case of unsual long distances, e.g., because of trafic or road constructions, the IoT infrastructure triggers an adaptation.

We study a scenario, where the IoT devices report that the transportation costs have doubled. To compensate for these high freight costs, we decide to increase the production of India as this location has the lowest fixed costs, i.e., manufacturing costs. We triple the production of the high-capacity site of India from 1,500,000 to 4,500,000 u/m. We must take into account that this increase of production also triples the fixed costs for the India high-capacity site. Running the optimization code gives us the total costs of 60,259,000 dollar per month. This scenario highlights the importance of the IoT infrastructure and the model-based systems engineering. The MBSE end-to-end approach allows the IoT system to inform of any infrastructure changes as early as possible. Having this information can help decision makers to adapt to the changes rapidly and optimize the supply chain to meet performance requirements.

```
package IoTInfrastructure {
    import Meta_SupplyChain::*;

    part MQTT {@MessageBroker;}
    part cloud {@Cloud;}
    part fog {@Fog;}
    part edge {@Edge;}
    part gateway {@Gateway;}
    part manufacturing {@IoTDevice;}
    part transportation {@IoTDevice;}

    connection {@Asynch;} connect MQTT to cloud;
    connection {@Asynch;} connect MQTT to fog;
    connection {@Asynch;} connect MQTT to edge;
    connection {@Asynch;} connect MQTT to gateway;
    connection {@Asynch;} connect transportation to
    gateway;
    ...
}
```

Fig. 8 Listing 5: Excerpt of the Sample Case in SysML 2.0: *IoT-Infrastructure Model*

Updating the optimization code Our studied scenarios so far focused on generation of the input files, e.g., increase of capacities and costs. We study a scenario where the supply chain model changes resulting in generation of new optimization code. Assume we add a third site to each location so that low capacity accounts to 500,000, medium to 1,500,000, and high to 3,000,000 u/m. Figure 9 (Listing 6) shows the updated model.

Moreover, the IoT infrastructure can add constraints to the optimization code. In the IoT-infrastructure model presented by Fig. 8 (Listing 5), we included a manufacturing device. Assume this device monitors the supply chain and informs that there is a malfunction in the low-capacity site of India. In this case, we add a logical constraint to the optimization code. Figure 10 (Listing 7).

Running the newly-generated optimization code with the updated parameters results in a total costs of 49,838,000 dollars per month. Our approach informs that to optimize the supply chain and compensate

```
package SupplyChainOptimization {
    import Meta_SupplyChain::*;

    part supplyLow    {@Capacity;}
    part supplyMedium   {@Capacity;}
    part supplyHigh    {@Capacity;}

    part location    {@SupplyChainNode;
      part supplyLow;
      part supplyMedium;
      part supplyHigh;
    }
    ...
}
```

Fig. 9 Listing 6: Excerpt of the Updated Supply-Chain Model

```python
import pandas as pd
from pulp import *

# added a medium-capacity site
size = ['Low', 'Medium', 'High']

# added a logical constraint
model += plant[('India','Low')] == 0

...
```

Fig. 10 Listing 7: Excerpt of the Updated Optimization Template

for the low-capacity site of India, we must take advantage of the low-capacity site of Brazil. Note that Brazil has the second lowest fixed costs, i.e., manufacturing costs, in our example. This scenario, again, highlights the importance of IoT infrastructure and the MBSE approach. Being informed of the manufacturing failures and automatically getting suggestions for replacement allows decision makers to optimize the supply chain in a timely manner.

5 Conclusions

Supply chain management is crucial in business [2, 4]. The new technologies, such as IoT help the efficiency of SCM. In this paper, we set out to answer the research questions how MBSE can assist the design of IoT-based supply-chain management systems (**RQ1**) and how we can optimize the supply chain using the IoT data integrated in the cloud (**RQ2**). For **RQ1**, we proposed metadata in the domain of IoT-based supply chain management. Architects can model their systems in SysML 2.0 using our metadata as tags. Our MBSE approach automatically generates artifacts, e.g., optimization code. For **RQ2**, we modeled an IoT infrastructure, where the device data is integrated in the cloud via message brokers using the event-driven communication pattern [14]. We discussed multiple scenarios, where the IoT detection data can inform the decision makers of any malfunction, and our approach automatically suggests an optimization of the supply chain.

For our future work, we plan to extend our approach to consider requirements validation. There are many standards, e.g., IEC 61131-3:2013 [1], considering system design. Our MBSE approach can automatically perform requirements validation and inform of specific system parts violating the requirements as in our previous work (see, e.g., [5, 6]). We convert the system model into a graph and perform graph-based validation of requirements.

Funding Open access funding provided by TU Wien (TUW).

References

1. Programmable controllers—part 3: Programming languages, 2013-02-20. https://webstore.iec.ch/en/publication/4552
2. Abdul-Azeez O, Ihechere AO, Idemudia C (2024) Optimizing supply chain management: strategic business models and solutions using sap s/4hana
3. Adama HE, Popoola OA, Okeke CD, Akinoso AE (2024) Economic theory and practical impacts of digital transformation in supply chain optimization. Int J Adv Econ 6(4)
4. Akinsulire AA, Idemudia C, Okwandu AC, Iwuanyanwu O (2024) Supply chain management and operational efficiency in affordable housing: An integrated review. Magna Sci Adv Res Rev 11(2):105–118
5. Amiri A, Just V, Steindl G, Nastic S, Kastner W, Gorton I (2024) Deployment architectures of mqtt brokers in event-driven industrial internet of things. In: 50th Annual Conference of the. IEEE, Industrial ElectronicsSociety (IECON)
6. Amiri A, Steindl G, Gorton I, Hollerer S, Kastner W, Sauter T (2024) Integrated safety and security by design in the IT/OT convergence of industrial systems: A graph-based approach. In: International Conference on Software ServicesEngineering. IEEE,
7. Cabane H, Farias K (2024) On the impact of event-driven architecture on performance: An exploratory study. Future Gener Comput Syst 153:52–69
8. Cederbladh J, Cicchetti A, Suryadevara J (2024) Early validation and verification of system behaviour in model-based systems engineering: A systematic literature review. Acm Trans Softw Eng Methodol
9. De Saqui-Sannes P, Vingerhoeds RA, Garion C, Thirioux X (2022) A Taxonomy of MBSE Approaches by Languages, Tools and Methods. IEEE Access 10:120936–120950
10. Goli A, Babaee Tirkolaee E, Golmohammadi A-M, Atan Z, Weber G-W, Ali SS (2023) A robust optimization model to design an iot-based sustainable supply chain network with flexibility. Cent Eur J Oper Res: 1–22
11. Ikevuje AH, Anaba DC, Iheanyichukwu UT (2024) Optimizing supply chain operations using iot devices and data analytics for improved efficiency. Magna Sci Adv Res Rev
12. Khan Y, Su'ud MBM, Alam MM, Ahmad SF, Ahmad AYB, Khan N (2022) Application of internet of things (iot) in sustainable supply chain management. Sustainability 15(1):694
13. Kharatyan A, Tekaat J, Japs S, Anacker H, Dumitrescu R (2021) Metamodel for safety and security integrated system architecture modeling. Proc Des Soc 1:2027–2036
14. Mirampalli S, Wankar R, Srirama SN (2024) Evaluating nifi and mqtt based serverless data pipelines in fog computing environments. Future Gener Comput Syst 150:341–353

15. Mohammadisiahroudi M, Fakhimi R, Terlaky T (2024) Efficient use of quantum linear system algorithms in inexact infeasible ipms for linear optimization. J Optim Theory Appl 202(1):146–183
16. Nanjundan P, James BV, George JP, Tiwari A (2024) Iot-enabled supply chain management for increased efficiency. In: International Conference on Trends in Quantum Computingand Emerging Business Technologies. In, vol 2024. IEEE, pp 1–5
17. Lohitha NS, Pounambal M (2023) Integrated publish/subscribe and push-pull method for cloud based iot framework for real time data processing. Meas Sensors 27:100699
18. Vogel-Heuser B, Zhang M, Lahrsen B, Landler S, Otto M, Stahl K, Zimmermann M (2024) Sysml'—incorporating component properties in early design phases of automated production systems. at. Automatisierungstechnik 72(1):59–72

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Amirali Amiri, Dr.; is a postdoctoral researcher at the Vienna University of Technology, Austria. Amirali defended his doctoral dissertation with distinction titled "Modeling and Multi-Faceted Reconfiguration of Cloud-Based Dynamic Routing" at the University of Vienna in 2023. He received a Certificate of Appreciation for "publications in highes ranking venues in 2023" in the first place of the group doctoral candidates. (PraeDocs) of the Faculty of Computer Science, University of Vienna. His research interest focuses on architecture of distributed systems, artificial intelligence in systems design, and model-based systems engineering. Amirali has successfully participated and concluded multiple international research projects in collaboration with the academia, e.g., University of Stuttgart, Germany, and industry, e.g., Siemens Austria.

Stefan Nastic, Prof.; is an Assistant Professor at TU Wien, working in the Distributed Systems Group. He is also a Founder and CEO of IntelliEdge GmbH—a company that provides software development and consulting services for Cloud Computing, Industry 4.0, and Artificial Intelligence worldwide. Stefan is also a member of the Technical Steering Committee in Linux Foundation's Centaurus project. Stefan got his Ph.D. in 2016 from TU Wien, with a thesis: "Programming, Provisioning, and Governing IoT Cloud Systems". His main research interests include Serverless Computing, Compound AI & Edge AI, 3D & Edge-Cloud Continuum, Reliability Engineering, and Software-defined Internet of Things. During the last decade, Stefan has been involved with many large international research and commercial projects related to Cloud Computing, Edge Computing, and Artificial intelligence, such as a multi-million Pacific Controls Cloud Computing Lab (PC3L) and Futurewei Polaris Cloud. During this time, Stefan has gained extensive experience acting as a technical coordinator, product owner, project manager, and advisor.

Wolfgang Kastner, (Senior Member, IEEE) received the Dipl.-Ing. and Dr. techn. degrees in computer science from Technische Universität Wien (TU Wien), Vienna, Austria, in 1992 and 1996, respectively. He is currently a Full Professor of the Industrial Internet of Things with the Faculty of Informatics, TU Wien. His research addresses distributed automation and (industrial) communication systems in various application domains, such as factory automation, building automation, and smart grids. His research topics tackle the safe while secure IT/OT convergence and approaches for the Industrial Internet based on information modeling and knowledge representation.