Budapest University of Technology and Economics
Department of Manufacturing Science and Control

Institute for Computer Science and Control
Research Laboratory on
Engineering and Management Intelligence

Resource sharing in distributed production systems

PhD thesis

Ádám Szaller

Supervisor:

Dr. Botond Kádár

Budapest, 2023

“If you are still looking for that
one person who will change your
life, take a look in the mirror.”

Roman Price

Declaration

Herewith I confirm that all of the research described in this dissertation is my own original work and expressed in my own words. Any use made within it of works of other authors in any form, e.g., ideas, figures, text, tables, are properly indicated through the application of citations and references.

Budapest, 01.06.2023

Ádám Szaller

Nyilatkozat

Alulírott Szaller Ádám kijelentem, hogy ezt a doktori értekezést magam készítettem és abban csak a megadott forrásokat használtam fel. Minden olyan részt, amelyet szó szerint, vagy azonos tartalomban, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Budapest, 2023.06.01.

Szaller Ádám

Abstract

Today's unpredictable variability in customer demand is a major challenge for manufacturing companies, which often have to keep extra resources to meet deadlines during overloaded periods. However, these resources remain unused during periods with lower loads. The aim of the research is to develop resource sharing methods that are applicable in distributed manufacturing structures and allow the parallel requesting and offering of manufacturing resources. One way to achieve this type of collaboration is so-called crowdsourcing, whereby an organization outsources jobs traditionally done internally to a group of external, independent organizations, typically online.

First, the related literature is reviewed, starting with the distributed production structures, then going into more details on the topic of collaborative sharing of manufacturing resources. The literature on agents, multi-agent systems and agent-based simulation – as a tool for testing and validating resource sharing mechanisms – is also briefly reviewed.

Next, the developed direct exchange-based resource sharing mechanism is presented, highlighting the communication mechanism, the computation of the available resources, and the decision logic of the participants. Then, a platform-based resource sharing mechanism is detailed, focusing on the central role and the request-offer matching logic of the platform. The presented results are tested and validated using an agent-based simulation model and a detailed overview of the architecture of this model is also provided.

Next, the topic of trust models is introduced: after a detailed literature review, an evaluation system is suggested, which is based on delivery time accuracy helping participants' decision-making. The benefits of considering the trust factor and the effect of manufacturing lead time prediction accuracy are also investigated using the simulation model. In the thesis, direct communication and platform-based resource sharing models are also compared based on different aspects, as well, an economic model for platform-based resource sharing is described. In the financial model, the different cost and revenue types of the manufacturing companies and for the platform are formalized one by one. The financial advantages and disadvantages of joining the platform are also discussed. Finally the financial model is tested with agent-based simulation, and then the contents of the thesis, new research results, areas of application and possible future research directions are summarized.

Kivonat

A vevői igények napjainkban tapasztalható, előre nem jelezhető változékonysága komoly kihívást jelent a gyártó vállalatok számára, amelyek gyakran kénytelenek többleterőforrásokat üzemeltetni annak érdekében, hogy terheltebb időszakokban is határidőre teljesíteni tudják a megrendeléseket. Ezen erőforrások azonban a kevésbé terhelt időszakokban kihasználatlanok maradnak.

A kutatás célja olyan, elosztott gyártási struktúrákban működő erőforrásmegosztási módszerek kidolgozása, amely lehetővé teszi a gyártási kapacitások párhuzamos kiajánlását és igénylését. Az erőforrásmegosztás egyik módja az ú.n. *crowdsourcing*, amelynek lényege, hogy egy szervezet a hagyományos esetben belsőleg, saját erőforrások igénybevételével elvégzett munkákat külső, független szervezetek csoportjának szervezi ki, jellemzően online formában.

Az értekezésben elsőként a kapcsolódó szakirodalmat tekintem át: az elosztott termelési struktúrákkal kezdve, majd a kollaboratív módon történő gyártási erőforrásmegosztás témakörét részletezve. Az ágensekkel, multiágens rendszerekkel és az ágensalapú szimulációval, mint az erőforrásmegosztási mechanizmusok tesztelésére és validálására alkalmas eszközzel kapcsolatos irodalmat is ismertetem. A bizalmi modellek témakörét részletes elemzése után a kapcsolódó termelési modellek pénzügyi megközelítéseit is áttekintem.

Ezután bemutatom az általam kidolgozott direkt kommunikáción alapuló erőforrásmegosztási mechanizmust, külön kitérve a használt kommunikációs struktúrára, a rendelkezésre álló erőforrások számítására, illetve a résztvevők döntéshozási logikájára. Ezt követően a platform-alapú erőforrásmegosztási mechanizmust részletezem, a platform irányító szerepére és az igény-ajánlat párosító logikára hangsúlyt fektetve. A bemutatott eredményeket ágensalapú szimulációs modell segítségével tesztelem és validálom, ezen modellnek a felépítését is részletesen áttekintem. Egy, a szavahihetőségen (főként a szállítási határidők betartásán) alapuló értékelési rendszerre teszek javaslatot, amely a résztvevők döntéshozását segíti. A szavahihetőség figyelembevételével járó előnyöket és a gyártási átfutási idő becslési pontosságának hatását szintén a szimulációval vizsgálom. Az értekezésben a direkt kommunikáció- és a platform alapú modelleket is összehasonlítom különböző szempontok alapján, majd pedig egy gazdasági modellt írok le a platform alapú erőforrásmegosztás esetére. Ez a modell formalizálva tartalmazza a gyártó vállalatok, illetve a platform egyes költségeit és bevételeit. Megvizsgálom a platformhoz való csatlakozás pénzügyi előnyeit és hátrányait, majd a gazdasági modellt szimulációval tesztelem. Ezt követően az értekezés tartalmát, az új kutatási eredményeket, az alkalmazási területeket és a lehetséges jövőbeli kutatási irányokat összegzem.

Acknowledgements

I am immensely grateful to all those who have supported me throughout my PhD journey. Without their encouragement and help, completing this dissertation would not have been possible.

First and foremost, I would like to express my gratitude to my supervisor, Dr. Botond Kádár, and to Dr. József Váncza for their constant support and guidance. Their expertise, patience, and unwavering commitment to my research have been invaluable. I am fortunate to have had them as my mentors and grateful for the insights they shared throughout my research. I would also like to highlight the advice I received from Dr. Péter Egri, who always selflessly helped me by sharing his ideas and giving feedback on my work.

I am thankful to several colleagues of the EMI Laboratory for their insightful comments and feedback. Their guidance has been invaluable in shaping my research and improving the quality of this dissertation.

I am also indebted to my wife and family for their unwavering support, encouragement, and understanding throughout my PhD journey. Their belief in me has been a constant source of motivation and inspiration.

This research has been supported by the TKP2021-NKTA-01 NRDIO grant on "Research on cooperative production and logistics systems to support a competitive and sustainable economy".

Abbreviations

Abbreviation	Description
ABM	Agent-Based Modelling
AI	Artificial Intelligence
BDI	Belief-Desire-Intention (model)
BTO	Build-To-Order
CAD	Computer Aided Design
CDD	Confirmed Delivery Date
CNC	Computer Numerical Control
CPS	Cyber-Physical System
DES	Discrete-Event Simulation
FC	Federation Center
FCFS	First Come First Served
GIS	Geographic Information System
KPI	Key Performance Indicator
MaaS	Manufacturing as a Service
MAS	Multi-Agent System
ML	Machine Learning
MRP	Material Requirements Planning
OEM	Original Equipment Manufacturer
TRS	Trust and Reputation System
SME	Small and Medium Enterprise

Table of Contents

1. Introduction

1.1. Overview and motivation

As a result of globalization, first, large manufacturing companies and then small and medium-sized enterprises (SMEs) are increasingly moving from rigid, centralized organizational structures toward distributed production networks [86]. The fluctuation of customer demands today also challenges Build-To-Order (BTO) companies, which are forced to operate with extra resources to meet the deadlines; however, these resources remain underutilized during less constrained periods. Problems like chip shortages in the semiconductor industry due to natural disasters, falling supply chains due to a war or geopolitical sanctions, demand shifts caused by COVID-19 also require production structures which are flexible enough to quickly react.

Crowdsourced manufacturing [66] can provide a solution to these challenges. Its essence is the following: an organization outsources jobs (that are traditionally done internally using its own resources) to a group of external, independent partners, typically through an online platform. This way temporarily extends its resources in a flexible way through some partners, making it easier to adapt to changes. The primary objective of the research is to develop a crowdsourced manufacturing-based resource sharing method that allows participating companies to offer unused resources, request them in case of shortages, and provides decision options for requesters by matching requests to offers. In order to justify the necessity of platform-based resource sharing, it is important that the introduced mechanism has to be more efficient and beneficial to the participants than a direct communication-based mechanism.

In a production system where participants share resources while cooperating (and, at the same time, competing) with each other, it is essential that they have an incentive to keep their promises, to be able to plan on the basis of the commitments made by their partners [132]. One aim of this thesis is to investigate how the effectiveness of resource sharing and the service level of the participants can be improved by taking trustfulness into account using a new rating system. The basis of the rating is one of the key factors in supply chains: meeting delivery deadlines, which is largely influenced by the manufacturing lead time prediction accuracy of the company that completes the outsourced job.

An additional aim is to develop a financial model for resource sharing companies and for the platform, to enable the financial justification of the effectiveness of the introduced method, furthermore, to investigate the circumstances in which it is profitable for a company to join the platform. The cost model should take the main cost elements and revenues of the manufacturing companies (production, administration, warehousing, distribution costs and sales revenues) into account. In addition, penalties for inaccurate deliveries and the platform's revenues and expenses also must be considered.

1.2. Outline of the dissertation

In Chapter 2, a literature review is presented, and state of the art is discussed in connection with the challenges in the Industry 4.0 era and manufacturing concepts trying to cope with these challenges. It is also discussed why collaborative resource sharing is needed and why it is still a very promising solution for manufacturing companies, particularly in a volatile environment. As the tool or technology for testing and validating the models is agent-based simulation, its basic concepts, multi-agent systems and simulation modelling are also discussed here. The role of trustfulness in the resource sharing mechanisms is introduced: a detailed literature review is conducted on trust and reputation systems: how they can be classified, attacking types and defending mechanisms, security issues and possible solutions, and case studies presented in the literature. Cost models used in similar manufacturing concepts are investigated, too.

In Chapter 3, a direct exchange-based resource sharing mechanism is introduced. First, the definitions are clarified, then the model itself, including the communication mechanism, calculation of available resources (to be able to consider resource constraints) and decision-making logic of the participants (choosing between offers) are described.

In Chapter 4, a new platform-based resource sharing mechanism is discussed. The basic concepts of the model are similar to the direct exchange-based mechanism. The main differences and the communication structure are also described in this part of the thesis, in parallel with the details about the functionality and role of the platform.

In Chapter 6, the agent-based simulation model is introduced including the different agent types (platform, company) with their parameters and functions. As the simulation software AnyLogic (the simulation technology applied for testing and validation) is Java-based, Java classes are discussed, which are necessary to realize the communication mechanism described in the previous chapters. The difference between reliable and non-reliable companies and the way they are modeled are also presented here.

In Chapter 5, the trust and reputation model applied in the resource sharing mechanism is introduced: how the delivery accuracy-based ratings are calculated and aggregated and how these are influencing the decision making of the companies. Experiments performed with the simulation model are also presented here: the effect of considering trustfulness in resource sharing and the effect of lead time prediction accuracy of the companies are also presented. At the end of the chapter, conclusions are drawn about taking trustfulness into account.

In Chapter 8, the comparison of the two resource sharing mechanisms (direct exchange-based vs platform-based) is presented based on several aspects, such as participant anonymity or trustfulness. Differences between them are investigated with the simulation model: average resource utilization, service level of the companies and communication loads are tested.

In Chapter 9, a new financial model is introduced for platform-based resource sharing to investigate incomes and costs for both the companies and the platform. Manufacturing, inventory, penalty, distribution, and administration costs are introduced, followed by the sales and penalty incomes of the participants, as well as the incomes and the costs of the platform. For each above-mentioned element, some examples are mentioned from the literature and a calculation method is formulated. Direct exchange-based and platform-based resource sharing are compared here from the financial perspective, too: additional costs and incomes of joining the platform are discussed. Simulation experiments are also performed to test the financial model and to investigate the effect of order interarrival time and the price of outsourced jobs.

In Chapter 10, new scientific results are summarized, and the application of the results is discussed. The presented work is summarized, and some interesting future research directions are mentioned.

2. Literature review

Nowadays, the trends of decentralization and interoperability are resulting in advanced manufacturing production models that pursue greater flexibility and functionality, such as Cyber-Physical Systems (CPS) and Cloud Manufacturing (CM). Recent manufacturing system paradigms have shifted their focus from production maximization to cost reduction, from process standardization to mass customization, and from production-centric to service-oriented [91].

Decentralization means moving from centralized control and processing to local and distributed. It enables autonomous control by distributing decision-making to the system elements [136], and has the potential to create benefits from it, e.g., self-organization, self-regulation, and flexibility [58]. Distributed control is an area with increasing interest for I4.0 and smart manufacturing applications, also [10].

According to ISO16300, interoperability is *"the ability for two or more entities that can exchange or share certain items in order to perform their respective tasks"* [67]. Interoperability is a key enabler for manufacturing to realize operations across heterogenous digital systems [90] and is still a problem for industrial implementation of new information and communication technologies [105]. Figure 1 shows the importance of interoperability for global manufacturers due to the number of distributed processes required to create a product. Authors of [89] state that interoperability in vertical and horizontal integrations, along with low-level control from Multi-Agent Systems (MAS), is required to realize CPS. Standards addressing information exchange and interfaces with legacy systems must be considered for heterogenous interoperability.

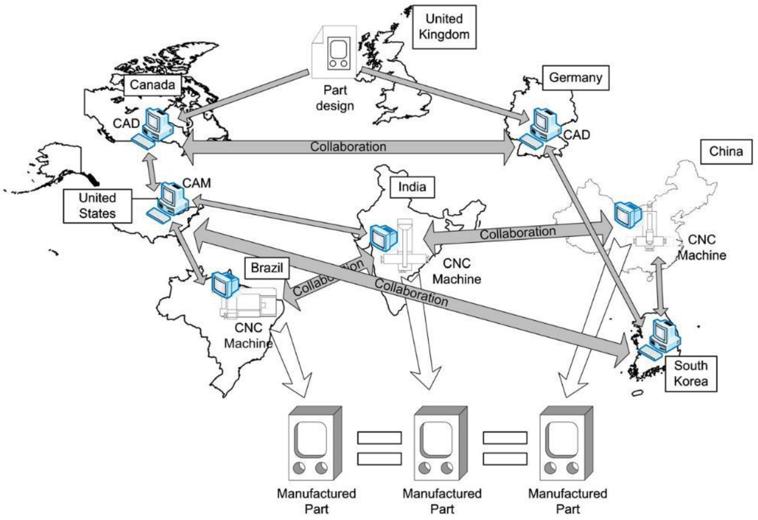

Figure 1. Interoperability for global CNC machining [99]

2.1. Manufacturing concepts

The traditional model of the manufacturing industry is getting transformed. Driven by an on-demand, fit-for-purpose service philosophy, the manufacturing industry has to deal with increasingly complex supply networks [12][16][38]. Additionally, megatrends such as mobility, urbanization, ecology and digitization cause increased environmental (external) complexity for manufacturing companies [61][87]. As part of these megatrends, logistics and supply chain management are affected by major changes [11][38]. Managing the resulting complexity is, therefore, one of the biggest challenges of supply chain management [12][29][31][129]. Only those companies that will accept and use these challenges to their advantage will remain competitive.

Ever-new crises accompany these continuous developments. The probabilities of crises and crisis-like phenomena are increasing [6]. A crisis is a difficult situation or situation with destructive development trajectories [84]. Crisis-like phenomena are conflicts, risks, and disasters. The nature and characteristics of these crises, and thus their impact on the supply chain, vary widely. The most recent crises are the semiconductor and the corona crisis, and unfortunately the war in Ukraine. The latter does not affect any specific sector, industry or geographic area. It affects global supply networks in different areas at different times and intensities. The German economy was hit with -4.9% GDP drop of economic output in 2020, 74.5% fewer air passengers travelled, which caused major airlines to struggle. Consumers' spending dropped 4.6%, while online retail experienced a rise of 27.8% [120]. Unique to the corona crisis is especially the influence of it from both sides (demand and supply) on the supply networks. This creates lasting effects and bi-directional (forward and backwards) disruption propagation [68]. Often the problem was not the change in the total demand but the demand shift to another type of similar product (e.g., flour, yeast, toilet paper) [81]. Since many international suppliers could no longer deliver, it led to global material and supply shortages [79]. Major suppliers were unable to produce in part due to regulatory requirements, which led to a production shutdown [15]. Capacity utilization on the one hand and capacity overload on the other, led in part to a bullwhip effect. The capacity utilization was countered by extra shifts, reactivation of idle resources and postponement of lower-priority orders [79]. Because of the corona crisis, global supply chains in particular, have come under criticism [60].

Partly triggered by the corona crisis, a global semiconductor crisis developed. During corona, many vehicle productions came to a standstill, and demand dropped. However, suddenly the industry picked up again, which led to a bullwhip effect. Due to the complex technology, semiconductor manufacturers need a certain amount of time to ramp up production again. OEM and the suppliers did not have this problem. The result was a phase shift. Semiconductor demand exceeded normal demand by far. The relatively long manufacturing times of about 170 days make it difficult to increase output in the short term.

One concept to cope with the above-mentioned challenges within supply chains is resilience. In [18], the authors define resilience in today's business world as an organization's capability responding to unexpected disruptions in order to restore normal operations. In [37] resiliencies defined as the characteristic of being adaptable to respond sustainably to sudden and significant changes in the environment in the form of uncertain demands. Many innovations with regard to supply chains, such as resilience approaches, have been investigated recently. Whereby the goal is often to increase the resilience of the network and therefore reduce total cost [69][78][128].

One of the most important resilience drivers is collaboration, which means joint risk mitigation of the partners, requiring mutual trust. Companies state that the outsourcing of production steps into collaboration-like organizational structures will likely increase in the future [118]. Concepts incorporating collaboration approaches like the resource sharing are enabled via recent developments like cyber-physical production systems, cloud computing and digital shadows [134]. These technologies support the integrated networking and the transparent data communication between every member of the supply chain and create new possibilities for networking and cooperation between different actors and stakeholders within a supply network [57][115].

The producer-consumer relationships in production networks are changing, giving room for increased cooperation in order to cope with such problems [76]. Cooperation and collaboration are not optional for manufacturing companies; it is a must if they want to remain competitive [2]. The authors of [8] distinguish between horizontal and vertical cooperation among enterprises, depending on whether they are at the same level of the supply chain in terms of value creation. Cooperation between supply chain actors is widely investigated in the literature from several perspectives: e.g., trustfulness [23] and robustness [116], but in these cases, the cooperation is always vertical between participants: the relationship between a producer and its supplier(s) is investigated. In the presented model, participants can have the same resource types and thus competitors of each other – they are on the same level of value creation (horizontal cooperation). Some of the methods developed for long-term customer-supplier relationships could be applied in resource sharing mechanisms (e.g., cost for penalizing inaccurate delivery time), but others have to be modified or rethought from the basics as the investigated problem is different.

Crowdsourced manufacturing was proposed by the International Electrotechnical Commission [66]. The main idea of the concept is to collaborate with each other by sharing resources via a *platform*. As mentioned in [77], for BTO companies often keep extra capacities to be able to meet order deadlines, and crowdsourcing can be an effective way to reach a high resource utilization level. This concept is also applicable for 3D printing, where offering resources in a Manufacturing-as-a-Service (MaaS) way is already in operation [34], and joining a platform could smoothen the demand fluctuations for resource offerors. This concept can also be a solution for companies having specialized, often expensive resources, facing a problem with utilizing them on a high level. As mentioned by [146], outsourcing and crowdsourcing help manufacturers – who are facing the challenge of multi-variety and variable-batch production orders – concentrate on their core business and share non-core business jobs with other companies.

An example of this type of collaboration is Swiss Virtuellefabrik [123], where SMEs focus on manufacturing unique products needing special equipment. The orders are distributed between the members by brokers, and the group of companies could not efficiently utilize their resources without working together. Other similar, manufacturing-as-a-service (MaaS) platforms are Xometry [137], MFG [95], and Fictiv [40]. They provide CNC, 3D printing, and injection moulding services in general. In the case of Xometry and Fictiv, the matching between customers and manufacturers is done by the platform, the customer is not able to influence the choice. Xometry even uses Machine Learning (ML) and Artificial Intelligence (AI) algorithms to determine the price promptly, based on previous jobs with its manufacturers, and the manufacturer is not known to the customer. In contrast, in the case of MFG, the customer has to compare e.g., the manufacturers, the prices, lead times, and choose the best option. Because of this, customer ratings and reviews are also available to support decision making.

Figure 2 shows the difference between demand fluctuation without and with being part of a resource sharing federation. In the first case, the company faces the costs of lost sales due to capacity shortage and lack of specific technology. In the second case, the company can "extend" its capacities by outsourcing some of the jobs to be able to complete more orders even from more customers, but it must face the additional outsourcing efforts, such as transportation and organization costs.

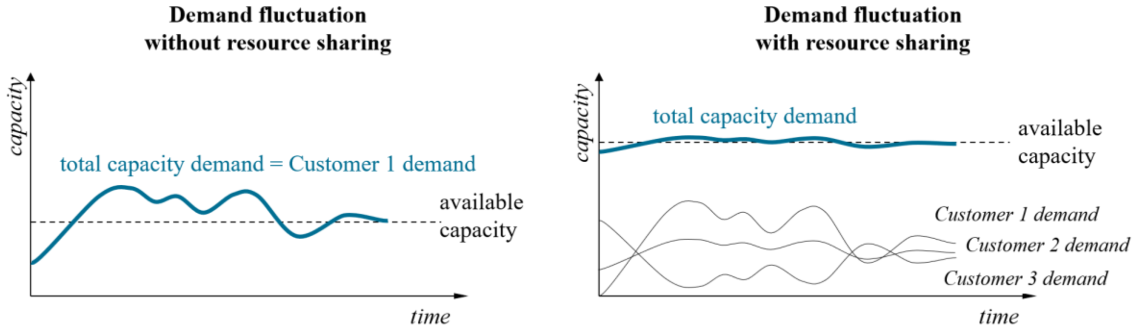

Figure 2. Demand fluctuations without and with resource sharing

At the company level, planning issues due to changes in demand are dealt with by Material Requirements Planning (MRP). The nature of demand can be deterministic, stochastic, seasonal, independent, or dependent. The resource sharing model presented in this thesis offers a solution when, due to unpredictable changes in demand, traditional planning methods no longer work effectively, and companies are forced to make quick decisions to meet the deadlines [41]. In this case, time and costs can be also saved by using a method capable of providing resource alternatives quickly. Fast decision making is very important mainly for supply chains producing innovative products with volatile and unpredictable demand. In comparison, in case of functional products, such as paper towels and light bulbs – where the demand is stable – the emphasis is placed on the effectiveness of the supply chain and cost minimization [42].

2.2. Resource sharing

Resource sharing can be defined as a cooperative action that has gained a lot of attention in recent years. Users of a shared resource receive the advantages of ownership, such as availability and use, while the disadvantages, such as investment costs and environmental impact, are reduced [8]. In case of fluctuating customer demands, the company may face the problem of idle capacities (lower demand than expected) or lost sales (higher demand than expected), as shown in Figure 3.

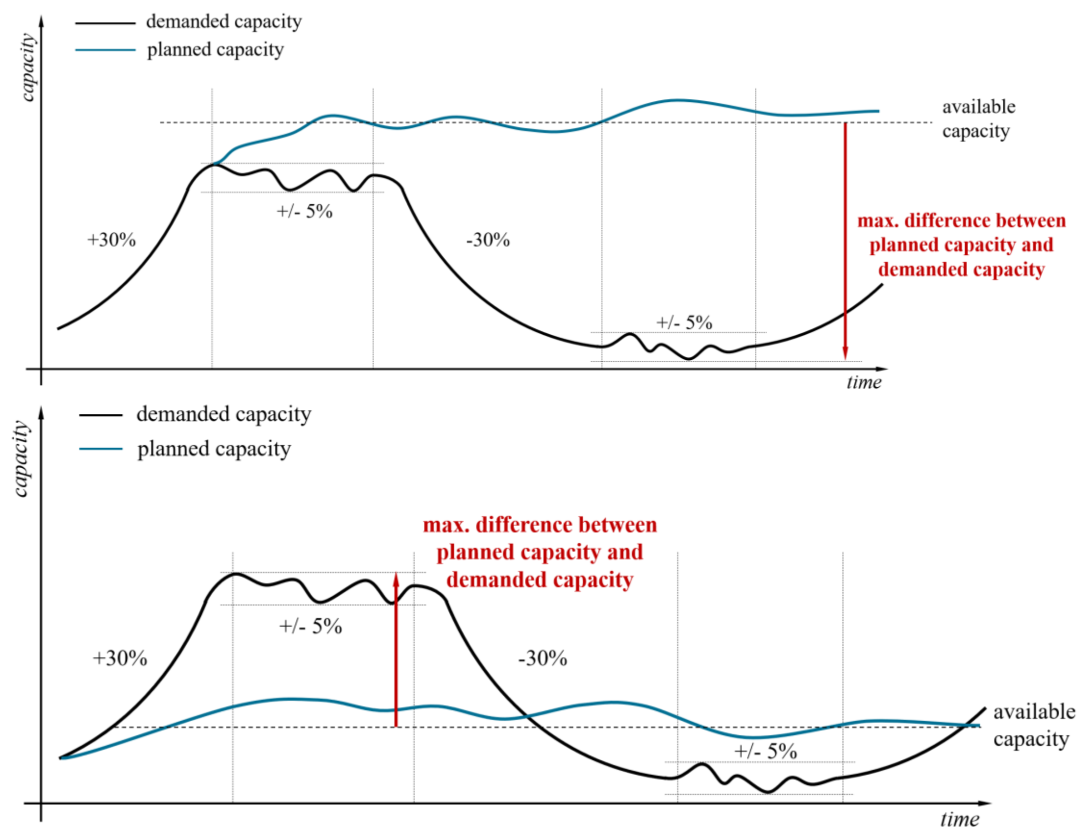

Figure 3. Idle capacities and lost sales in case of fluctuating demands

Table 1 summarizes the challenges solved by resource sharing.

Table 1. Challenges solved by resource sharing

Challenge	**Solution enabled by resource sharing**
Keeping extra resources in order to meet delivery deadlines of larger orders; but these may remain unused during less loaded periods	Sharing resources with each other: requesting them when having shortages and offering them when having surplus
Imprecisely predictable customer orders, disturbances in supply network causing fluctuating utilization of production system and difficult planning	
Fluctuating demand, underutilized capacities for companies having resources that can be used generally, e.g., laser cutting, 3D printing, CNC machines	Resource sharing platform where they can offer their resources, this way smoothening the demand and utilize their resources by receiving more orders

Buying specific equipment to be able to produce certain products, this way spending resources on tasks that are not the core business of the company	Outsourcing certain job phases to companies focusing on operating specialized equipment, and focusing on core business tasks that the company can complete with higher efficiency

The Product-Process Matrix (see Figure 4) was first introduced by Hayes and Wheelwright [54]. The process life cycle-rows of the matrix represent the process structure with increasing standardization towards the systemic form. The product life cycle columns represent the product structure going from great variety to highly standardized products. The resource sharing solutions discussed in this thesis are relevant for companies who are located at the two ends of the diagonal:

(1) They have general resource types (top left corner), such as 3D printers, CNC machines, laser cutting machines. In this case, defining jobs in a standardized way is relatively simple.

(2) They have specialized resources that are expensive to operate (bottom right corner), e.g., producing spare parts for the railway industry. In this case, as the machines produce only specific, highly standardized products, with already defined technical requirements that the machines are capable of.

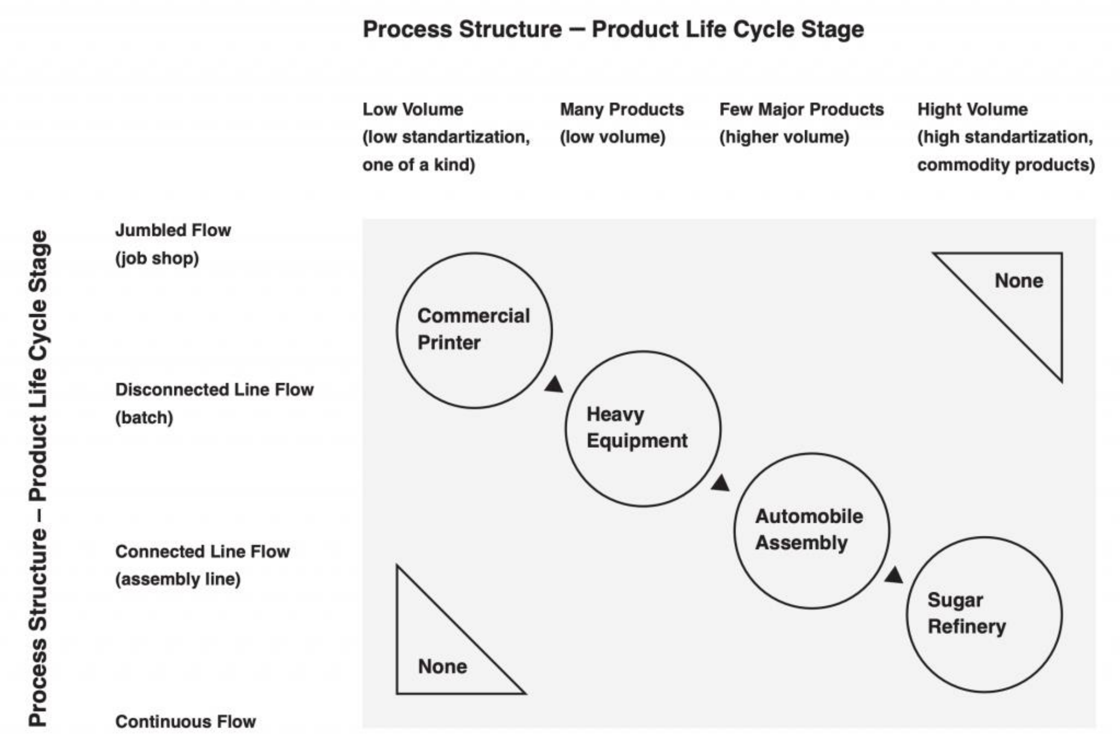

Figure 4. Process-Product Matrix [54]

Some examples of studies investigating resource sharing approaches with a sustainability aspect are presented in Table 2. Based on these examples, it can be stated that resource sharing is one possible and promising way towards sustainability – however, currently it is not a widely used solution.

Table 2. Sustainability aspects in resource sharing models found in the literature

Reference	**Summary/finding**	**Resource sharing between…**	**Sustainability aspect**
[146]	Balancing and reducing air pollutant emission of the whole region in China with resource sharing between production companies.	manufacturing companies	air pollution reduction
[104]	Analysis of how resource sharing – as a key practice in making transition to circular economy – can be facilitated and implemented. Different implementation strategies and transition pathways between them are also examined. A case study of nine Virtual Power Plants is conducted.	virtual power plants	circular economy
[139]	Multi-depot green vehicle routing problem is investigated implementing transportation resource sharing within the same depot and among multiple depots to minimize carbon emission and operating cost.	transportation vehicles	CO2 emission
[83]	Collaborative resource sharing improves supply chain performance and sustainability	theoretical study	general view
[110]	Blockchain-based smart contract for resource sharing companies, MILP model for optimal resource sharing and scheduling (measures: makespan, machine utilization, energy consumption, and reliability).	manufacturing companies	energy consumption in manufacturing

	Case study with 36 companies, manufacturing gearboxes.		
[45]	Resource sharing for intermodal transportation and performance measures are presented, and the model is tested with an agent-based simulation.	transportation vehicles	CO2 emission
[138]	Collaborative multi-center vehicle routing problem with resource sharing and temperature control constraints is solved with a hybrid heuristic algorithm.	transportation vehicles	environmental impact of transportation
[94]	Reducing environmental impact of logistics by sharing of operational capabilities, either by vehicle sharing, by vehicle capacity sharing, sharing warehousing or infrastructure sharing. Real case study (tested in a Portuguese city) is presented.	transportation vehicles	environmental impact of transportation

2.3. Agents and multi-agent systems

Agent, as an abstract concept, appeared as early as the 1960s, but it became more widespread in the 1990s. There are several definitions for the agent expression; an agent can be any independent entity (e.g., software, model, individual) [13]. According to [96], *"an agent is a computational system that is situated in a dynamic environment and is capable of exhibiting autonomous and intelligent behavior"* (see Figure 5)

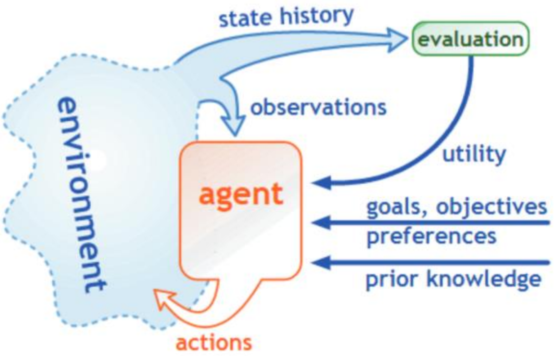

Figure 5. Agent interacting with its environment [96]

[102] classifies agents according to additional properties such as autonomy (initiative behaviour), cooperation, learning, mobility and the way of internal decision-making. The latter distinguishes between reactive agents, which react to the stimuli coming from their environment according to a set of "if-then" rules, and deliberative agents, which have a symbolic model of their environment, thus they can plan how to act. This behaviour is most often described by the BDI (Belief-Desire-Intention) model, where agents have a vision of the world (which may be wrong), goals, as well as plans to achieve their goals. In some applications (e.g., inter-company processes), they are characterised by honesty, transparency, and reliability.

Agents often operate in a multi-agent system. The architecture of multi-agent systems is highly customizable, which contributes to their wide range of applications. In [64] the characteristics are grouped into 5 categories:

(1) Intrinsic properties of agents, e.g., lifespan, mobility, cognitive level (from reactive to deliberative) or adaptive (permanent, teachable, self-learning).
(2) External properties of agents, e.g., social disposition, social autonomy (from independent to controlled), friendliness (cooperative, competitive or hostile), mode of interaction (direct or through some intermediary, only with other agents/environment or both, communication characteristics).
(3) System properties, e.g., homogeneity, structure (from hierarchical to democratic), freedom of implementation (from independent to controlled).
(4) Properties of the environment, e.g., familiarity, predictability, and degree of controllability from the agents' perspective. In addition, whether the whole past has an impact on future states or only the present. Furthermore, realism, i.e. whether the environment can change while the agent makes a decision.
(5) Properties of the framework, e.g., communication infrastructure and message protocol.

What are the benefits of a multi-agent system, and for what purpose they can be used? In [70], the authors define some characteristic settings helping to answer this question: no central control, decentralized data, the calculation is made in an asynchronous way, the agents have limited information or are not able to act alone to solve problems. This implies that they are capable of distributed problem-solving.

As there is no rigid central control, the system is robust: it does not collapse due to disruptions, changes, or the failure of an agent. It can return to a stable state through interactions between agents and with the environment; this way, it is possible to build complex systems. The complexity comes with emergent behaviour, i.e., the system has properties that are not the consequences of the individual agents' capabilities. This way, the system can solve tasks that are beyond the capabilities of the individual agents.

According to [132], *"Simulation is the representation of a system with its dynamic processes in an experimentable model to reach findings which are transferable to reality"*. Since the aim of this thesis is to investigate the performance of manufacturing companies and the dynamic resource sharing between them, simulation modelling as a method for testing and validating different methods seemed to be particularly suitable. By applying a simulation model, these mechanisms can be modelled in a realistic way, and the dynamics of the processes can also be managed.

For simulation modelling of manufacturing and logistics systems, in general, two main concepts are used: discrete event-based simulation (DES), and agent-based simulation (ABS). According to [20], ABS is suitable for systems with entities that frequently interact with each other, while DES focuses on simulating events and their relationships of the underlying discrete-event dynamic system. In this thesis, collaborative resource sharing mechanisms are modelled; thus, agent-based simulation was chosen as a tool for investigation to draw consequences and verify the functioning of the mechanisms.

2.4. Trust and reputation systems

For cooperating organizations, it is essential to be honest with each other and to have a strong commitment to the promises. By taking trust and reputation into account in decision-making, companies could be incited to keep their promises, e.g., complete an undertaken order in spite of noticing a more profitable option for using free capacities. They also can be forced not to bias information and to meet the job due dates because otherwise, they would worsen their own situation (after receiving a bad rating, they are less likely to win new jobs). Making decisions based on trust and reputation also enables to differentiate between partners who are reliable and who are not. Such a framework is driven by the promises and commitments for the future, given by the participants. The main internal testimonial of the framework is that one can believe the other's promises: if participants cannot count on these commitments, and they are not incited to keep the promises, the framework of cooperation is violated, and the efficiency of the distributed manufacturing system can decrease.

In this subchapter, the definition and classification of trust and reputation systems (TRSs), including features like, attack types and defense mechanisms, security issues and possible solutions, and case studies are presented from the literature, as well.

Based on the information collected, a requirement list is composed for TRSs applied in the manufacturing area, and the model used in resource sharing is also introduced. In the end, experiments are performed to investigate the effect of considering trustfulness together with the impact of lead time prediction accuracy in resource sharing.

2.4.1. Definition and classification

The various forms of trust may account for some of the apparent confusion on the concept of trust. Here the main goal is not to present all the definitions (as it has been done before in different surveys) but to mention some examples with the aim of making the concept of trust understandable.

More than 40 definitions have been collected by [85] from 1958 to 2009. The authors summarize the trust definitions in the following way: *"[...] trust relation implies the participation of at least two parties, a trustor and a trustee. The trustor is the party who places him or herself in a vulnerable situation under uncertainty. The trustee is the party on whom the trust is placed, who has the opportunity to take advantage of the trustor's vulnerability".* In Table 3, some other definitions are also mentioned from the literature.

Table 3. Trust definitions from the literature

Reference	Definition
[97]	Willingness to rely on an exchange partner in whom one has confidence.
[32]	An individual's belief or a common belief among a group of individuals that another individual or group (a) makes good-faith efforts to behave in accordance with any commitments, both explicit and implicit, (b) is honest in whatever negotiations preceded such commitments, and (c) does not take excessive advantage of another even when the opportunity is available.
[110]	A device to reduce complexity, a shortcut to avoid complex decision processes when facing decisions that carry risk.
[30]	Willingness to commit to a collaborative effort before you know how the other person will behave.
[72]	A personal and subjective phenomenon, based on various factors, some of those (such as personal experience) having more weight than others (such as second-hand information).
[93]	The willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party.
[63]	Trust is a complicated issue concerning the belief in trustfulness, integrity, competence, and dependability of the trusted system actors.
[112]	A psychological state comprising the intention to accept vulnerability based upon positive expectations of the intentions or behavior of another (neither a behaviour, nor a choice, but a psychological condition that causes or results from such actions).

As one can see, there is no generally accepted definition for trust: each work defines this concept from its own perspective. For example, different definitions exist for online commerce, sharing economy or supply chains. In addition, these definitions use expressions making them more obscure (because these expressions are not defined either) - e.g., interdependence, vulnerability and benevolence.

However, it is necessary to distinguish between trust and a concept that is very close to it: *reputation*. After reviewing various reputation definitions and models, it can be stated that the difference between trust and reputation can be phrased as follows: while trust is some kind of opinion based on *direct experiences*, reputation is based on *indirect interactions* and primarily used when no (or not enough) information is available from another party. This way, information can be obtained from a previously unknown entity, with the advantage of reducing interaction risk (among others). Here, some reputation definitions and models are mentioned to introduce the concept in a more detailed way.

In [92], the authors review computational reputation models for multi-agent systems. They define reputation as socialized trust that is propagated through a social network, while trust itself is a belief that is based on personal experiences. In [21], several reputation systems in practice are reviewed and compared. Ratings of merchants', products', customers', reviews' and reviewers' reputations are taken into consideration. Several definitions of reputation are provided in the mentioned paper: almost all of them are closely related to third-party opinions, especially first-, second- and third-hand ones, in this order of importance.

Collaborative filtering systems are similar to reputation systems as both of them collect ratings from community members. However, collaborative filtering is based on the idea that different people have different tastes. Consequently, if two users evaluate the same set of items in the same way, they can be classified into the same cluster. On the other hand, reputation systems assume that the ratings are insensitive to taste. If ratings subject to individual preferences are fed into a reputation system, then reputation scores can mislead because they reflect different tastes, not the different reliability of service providers [72]. In [114], the authors review computational trust and reputation models along the following classification dimensions, focusing on the functionality of the systems; the summary of their literature review is depicted in Figure 6.

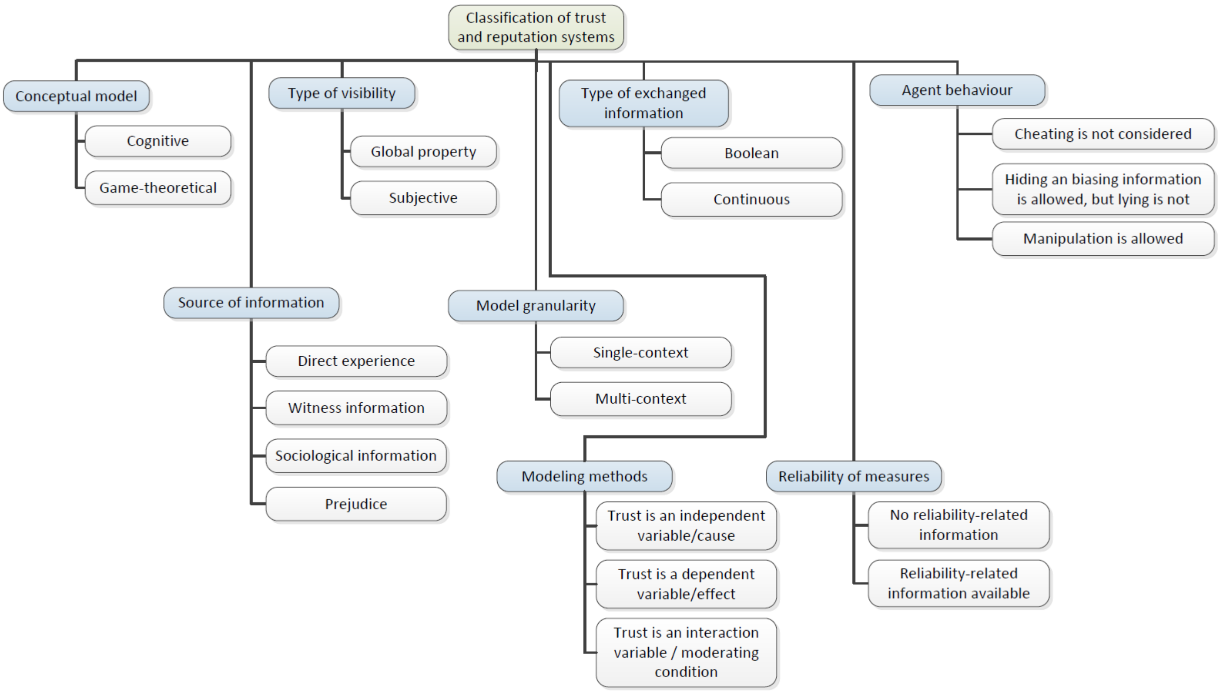

Figure 6. Classification of trust and reputation systems [114]

Trust relationships can be classified based on their type. We can distinguish between relationships based on the direction of the rating, the causes for building the relationship and the participants, see [47],[55],[65],[92] and [111]. The main relationship types are presented in Figure 7.

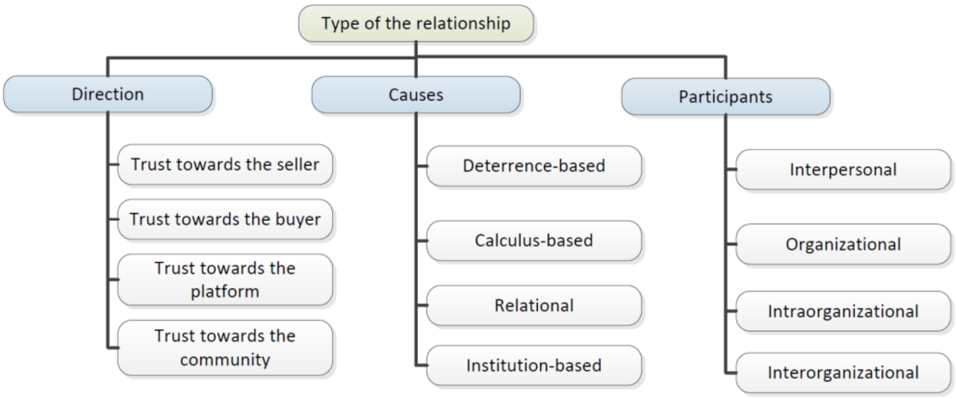

Figure 7. Types of trust relationships

Several settings can describe a trust relationship, including its phase, stage or level. In addition, the relationship is determined on the basis of the factors that influence it. In Figure 8, these are summarized based on [7],[32] and [98].

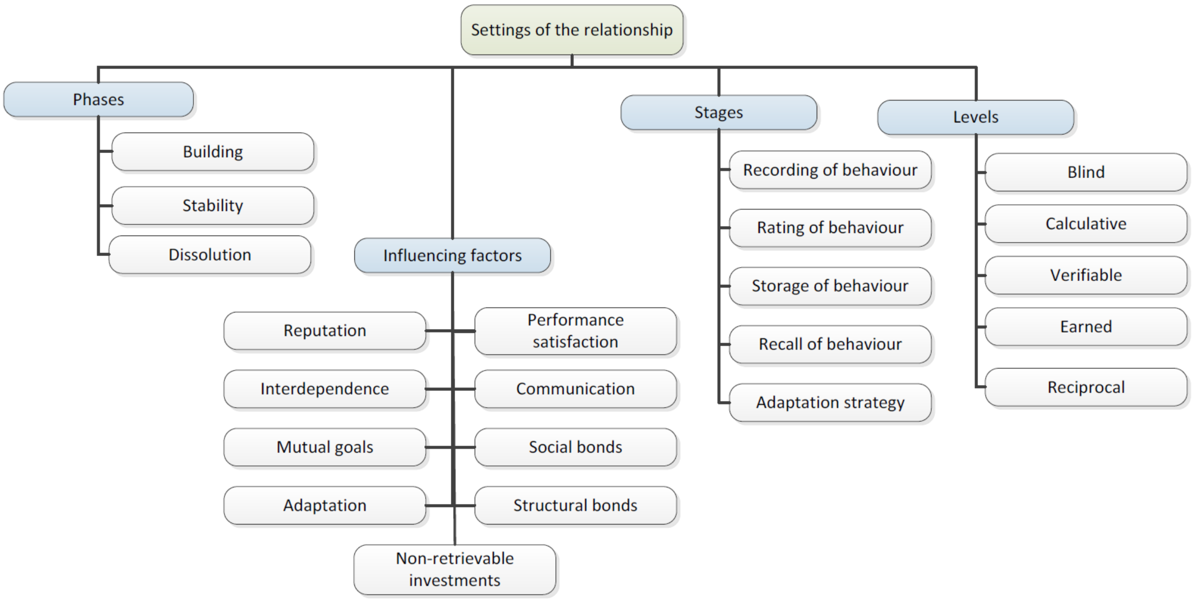

Figure 8. Settings of the trust relationship

2.4.2. Attacking types and defending mechanisms

Ratings of a given agent can be manipulated for selfish reasons by malicious or harmful agents. Based on [21], a malicious agent is an agent whose identity is unknown or undefined, or who is discovered to be untruthful and makes intentional mistakes and errors in order to disrupt the operation or business or provides misleading opinions to misguide the community. The detection of these (group of) agents and inciting them to be honest are challenging tasks. To detect malicious agents, data mining, pattern matching, behaviour monitoring and Just-In-Time reporting are useful tools. Based on [73] and [80], the different types of attacks against reputation systems are summarized in Figure 9.

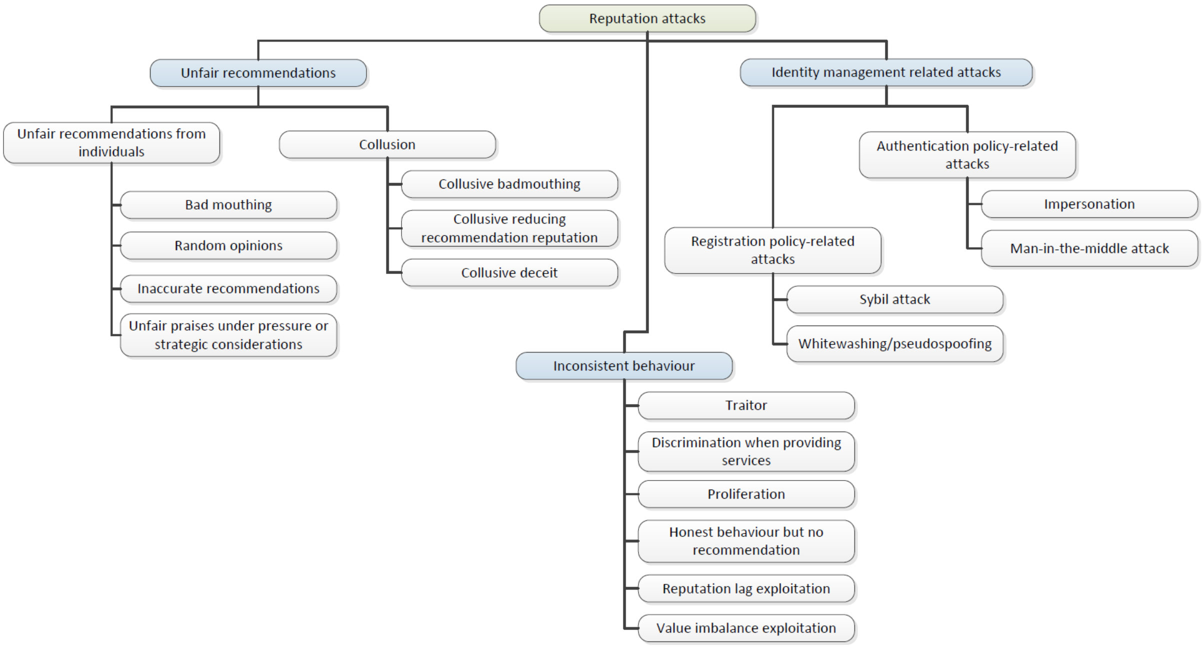

Figure 9. Attack types against reputation systems

Table 4 summarizes the inevitable conflicts presented by [80], in connection with the defence mechanisms developed by researchers against various attack types.

Table 4. Inevitable conflicts related to defence mechanisms in TRSs

Conflict			Suggested solution
Negative feedback sensitivity	vs.	Robustness against collective badmouthing	No suggested solution
Encouraging newcomers	vs.	Preventing whitewashing	Determining a default reputation value for newcomers
Resilience to oscillatory behaviour	vs.	Helping reputation restoration of previously misbehaving agents	Monitoring changes in behaviour
Performance	vs.	Accuracy	A larger history size improves the accuracy of reputation but requires storage space and computation time
Performance	vs.	Resilience to man-in-the-middle attacks	Reputation information redundancy
Considering only positive experiences in order to counteract badmouthing attacks	vs.	Resilience to collusive deceit attacks	Monitoring review patterns
Considering only negative experiences in order to counteract collusive deceit	vs.	Resilience to badmouthing attacks	Monitoring review patterns
Resilience to unfair recommendations (via similarity measures)	vs.	Considering honest recommendations which do not comply with the majority of recommendations	No suggested solution in the study
Encouraging recommendation provision (via rewards for recommendation)	vs.	Preventing random recommendations	No suggested solution in the study
Incentives for honest recommendations (via credit-based reward/punishment mechanisms)	vs.	Ease of development	No suggested solution in the study

2.4.3. Security issues and possible solutions

As a general fact we can state that information about trustfulness is highly confidential. Thus, a secure system should be designed to collect, store and update this information. Several researchers propose a blockchain-based system as a modern and secure solution. Although this thesis does not focus on the real implementation and security of the developed trust model, some relating examples investigating these issues are mentioned here.

PoRX (Proof-of-Reputation-X), a credit-based reputation incentive scheme for blockchain consensus of Industrial Internet of Things, introduced in [134], that rewards cooperative actions and punishes non-cooperative ones, confirmed by experimental results, too. *Fabrec*, a peer-to-peer network of manufacturing nodes proposed by [5], enables the participants to share data with the other participants, including the potential clients. Blockchain technologies and smart contract representations have been implemented and successfully tested in this case. In [88], the authors propose an order-driven trading service between manufacturers and customers, by introducing a reputation management system combining real-world reliability and feedback. Manufacturer rating classification and identification of malicious evaluators have also been implemented. The architecture is again blockchain-based. Simulation showed that the system meets the requirements of privacy, non-repudiation, anonymity, and fairness.

As one can conclude from the above-mentioned three examples, researchers see blockchain technology as the most promising solution for security issues and prevent malicious participants from harming others or the system itself.

2.4.4. Delivery time accuracy

Delivery time accuracy is essential in the case of manufacturing companies. In [1], the authors differentiate between two types of customers (agents): the first type accepts tardiness in the delivery of orders, the second type does not accept the tardy orders at all and assesses the tardy orders as failed. Authors of [51] mention that lower variance in delivery times improves delivery performance thus increases customer satisfaction among the existing customer base in the short term and can lead to new customers in the long term. They also define the delivery window as the difference between the earliest acceptable delivery date and the latest acceptable delivery date. In [48], Confirmed Delivery Date (CDD) is identified as the most important performance criteria. CDD means the reliability of a company in fulfilling the customer's order requirements. TRSs can also be used to motivate companies to keep their promises, especially in connection with delivery deadlines.

2.4.5. Case studies

Since TRSs for manufacturing resource sharing models have not been investigated by researchers yet, some examples are mentioned here from the topic of partner alliances and supply chains. Common projects are usually initiated by firms which are familiar with each other and have expectations of the intentions and possible behaviors of their partners. Hence trust obviously plays a prominent role in all alliances as it contributes to higher performance. Nevertheless is not easy, trust and reputation can be measured, and their impact also could be investigated between entities who are on the same or even on different levels of a supply chain or a network.

In the first case, trust is investigated between customer(s) and its suppliers. Here, the trust relationship is one-directional – ratings are only given by the customer(s), and they can use an aggregated rating to choose between different suppliers. In the second case, the participants could be each other's competitors, or they could be customers and suppliers for each other at the same time. Consequently, a specific entity could be rated by another one, and vice versa in two different interactions. This is what happens in a distributed manufacturing framework. In Table 5, some case studies are mentioned for both cases.

Table 5. Trust-related findings in partner alliance case studies

Reference	Findings
[36]	The complementary role between trust and control: the more you can trust your partner, the less control you need as the partner will behave as expected.
[52]	When firms enter multiple alliances over time, the transaction cost might decrease as they already know and trust each other.
[83]	Trust is positively related to alliance performance. Interdependence and inter-partner competition (behavioural uncertainty) increase the positive relation between trust and performance. Market instability and market unpredictability (environmental uncertainty) decrease the positive effect of inter-organizational trust on alliance performance.

Table 6. Trust-related findings in supply chain case studies

Reference	Findings
[17]	Trust promotes knowledge sharing, which improves the performance of the supply chain.
[46]	Trust contributes positively to supply chain management and is a powerful predictor of performance and competitive advantage.
[19]	Supply chain trust is a positive (negative) function of the number of uninfluenced (uninfluential) partners.
[111]	Three aspects of a network influence trust in supply chains: the number of uninfluenced partners, the number of influential partners, and the degree of interdependence.
[15]	Trust is positively, while asymmetric dependence is negatively associated with the success of supply chain projects.
[44]	The weighting in the computation of reputation values has a strong effect on which type of partners will be the dominant ones. Discount of reputation (caused by forgetfulness) strongly affects the relationships. Dominant market players are interested in increasing the influence of reputation in decisions.
[63]	Trust-based supplier selection increases the robustness of the network in comparison to price-based and random selection.
[144]	Negative connection between trust and the costs of negotiation and the level of conflict. Interorganizational trust has a greater impact on performance than interpersonal trust in the case of supplier relationships.

In the literature, one can find TRSs applied in different fields of the production and manufacturing area, too. The authors of [24] distinguish between five trust categories (competence, contractual, relational, indirect and negative), and investigate their impacts on information exchange processes in vertical collaborative networked organizations. In [23], a multi-criteria variable weights decision-making approach based on trust and reputation in supply chains is proposed. They put more emphasis on the detailed TRS, consider direct and indirect values and apply a time decay function for historical trust and reputation values as well, but ignore the resource constraints at the suppliers. In [142], a service satisfaction-based trust evaluation model for cloud manufacturing is presented, where the direct satisfaction, the friend recommendation satisfaction and the platform satisfaction were integrated into the comprehensive trust. This model also applies a time decay function and corrects trust values by using the service satisfaction volatility. The authors of [141] introduce a detailed TRS in Cloud Manufacturing. Here, direct, indirect, and third-party trust (which means relying on the opinion of independent and qualified third parties) are taken into consideration, and a time decay function is applied to historical transaction data. The model focuses on the trust evaluation model and takes several important aspects in connection with trust into account, but also ignores resource constraints. Nevertheless, the amount of available resources is an important aspect when investigating systems where the participants share resources with each other. A reliable participant could become overloaded and, consequently, other companies might choose a less reliable partner with free capacities instead of the reliable one which has no available resources.

2.4.6. Important aspects for TRSs in the production area

As one can see, trustfulness is beneficial in interactions from several aspects. For example, it increases performance and network robustness, reduces risks, negotiation, and transaction costs. In this section, the important aspects, possible requirements for TRSs that can be applied in the manufacturing area are discussed, based on the reviewed literature extended with novel thoughts, as well.

Trust types

The system has to support the different types of trust (towards the seller/buyer/platform/community, as depicted in Figure 7, as well as all the phases (building, stability, dissolution) and stages (recording, rating, storage, recall, adaptation) as shown in Figure 8.

Reliability and usability of results

Users must get useful and reliable data from the system: especially when own information is not available (e.g., about a previously unknown entity), the calculated reputation is a good estimation of the expected behavior of the partner. However, these values summarize other users' opinions, and while others might have given ratings based on different aspects, it could be misleading in certain cases. It is not totally sure that a participant is planning to make decisions based on the same aspects as the ones whose opinion is aggregated in the rating. Thus, using objective data about, e.g., performance, fulfilment, keeping deadlines or payment could be very useful even if some public ratings are available.

Findings presented in online commerce case studies could be applied to the manufacturing area, as well. Although a seller's reputation could have a significant effect on the selling price [62], as [13] mentions, sellers tend to reciprocate positive and retaliate negative feedbacks, which causes bias in the ratings.

As mentioned above, a TRS considering information from public reputation rankings can be useful also when direct experience is missing towards a specific partner. These rankings are created based on complex criteria. When planning to cooperate with a firm, not all these aspects have to be considered - however, the availability of this information helps to make better decisions when no other knowledge is accessible. These are also useful for, e.g., investors who are trying to obtain a general view of a firm at first.

Dealing with security, privacy and trustworthiness issues

The users could not get an advantage from being dishonest, and they must be motivated to report true ratings, and any attacks against the TRS have to be penalized. Trade-offs summarized in Table 4 and attack types mentioned in Figure 9 have to be considered. For example, how can a company share information with its competitor about the lack of specific resources? Blockchain-based solutions could be promising, as mentioned in [5],[88] and [134].

Motivation for companies to join and be active

When implementing a TRS, first, the most important task is to attract as many participants as possible. After reaching the "critical mass", a TRS has to provide services to its users motivating them to participate in the long term. Active participation also has to be motivated in order to avoid users taking advantage of the ratings without forming them. If there are not enough participants, it is not worthwhile for a user to devote energy to providing ratings about the others. In addition, the number of participants giving ratings determines that reputation has a positive correlation with sales price or with sales amount [143].

Time factor

To follow the dynamic nature of trust and reputation, a TRS should provide up-to-date information. A company has to be able to correct mistakes made in the past. This could be important if the specific company e.g., implements a change from the quality perspective. However, recent failures should occur in the records, too [21]. Discount of reputation (caused by forgetfulness) strongly affects the relationships: the TRS has to cope with this issue, also.

Weighting the ratings

As stated in [44], weighting the different aspects in the computation of reputation values has a strong effect on which type of partners will be the dominant ones. Dominant market players are interested in increasing the influence of reputation in decisions. Furthermore, ratings given based on a specific interaction should be weighted relative to each other's according to the magnitude of the transactions. The system cannot be manipulated by boosting own reputation with high ratings from many small-valued transactions or by value imbalance exploitation [73].

Handling interdependent jobs

A job completed with a delay by a partner, can occur that a company must change its already planned works or reschedule its production, to be able to perform the next manufacturing step. A TRS designed for the manufacturing and supply chain area should also cope with this: it is an open question and has not been studied yet. How should a calculation based TRS handle this situation? Who gets and exactly what penalty?

Easy implementation

A TRS must be flexible enough and compatible with the common platforms and enterprise resource planning software.

2.5. Cost model for resource sharing

Cost is a critical factor in the success of production, especially in today's competitive market and companies which are unable to provide detailed and meaningful cost forecasts have a distinct disadvantage. Therefore, when considering whether a company should join a resource sharing federation, the cost and benefit aspect is of particular importance. Appropriate cost forecasts estimating these effects in advance are necessary. According to the authors in [1] a cost forecast is: "*The prediction of the probable costs of a project or effort, for a given and documented scope, a defined location, and point of time in the future.*". The conceptual bases for this cost forecast are cost theories establishing the relationships between costs and their determinants. The mathematical formulation of the cost hypotheses takes place via so-called cost functions, which enable the forecast of the cost amounts. Due to the complexity of a production system, it is not possible/feasible to set up a single cost function. Instead, it is necessary to formulate several (partial) cost functions for subareas and combine them [117]. Generally, two techniques for cost forecasting can be used: qualitative and quantitative techniques. Qualitative techniques are the following:

- Based on data from the past, the cost of a new product is estimated.
- Historical products are examined for similarities with the product to be evaluated in order to generate an estimate of the costs or at least a basis for such an estimate if there are similarities [100].

Quantitative techniques are:

- The product is broken down into its components and their production processes. A subsequent analysis then evaluates the costs of these elements and adds them up.
- The sum of the resources required in the production process is formed. These methods promise more accurate results than qualitative techniques but involve more effort [100].

Within this thesis, a quantitative break-down approach is taken, which determines each process step in the overall production process, including indirect areas and assigns costs to them. The processes also include non-productive efforts such as setup times. In addition to the activities, the material costs are also included in the evaluation, whereby the focus is clearly on the activities and their process times. It is a very accurate way of determining costs, which can be applied late in the product development process [9][74][100][103][140]. To determine the relationship between the determinant and the cost level, three steps are necessary [56]:

(1) determination of the factors influencing the cost level,
(2) grouping of the cost-influencing factors, and
(3) formulation of the functional connection between the cost and the factors.

In subsection 9.8, platform-based resource sharing is compared with systems not sharing resources, from the financial perspective. For this, a detailed structure of the relevant cost types is necessary. Within supply chain management, various so-called cost structures were developed for this purpose. Most approaches focus on a differentiation based on the organizational units or activities similar to the SCOR-Model [121]. In [106], manufacturing cost (including direct material, direct labor and overhead production), administration cost (including order handling and planning), warehouse cost, distribution cost (including inbound and outbound transportation), capital cost, and installation cost are distinguished. Authors of [130] differentiate labor cost, inventory holding cost, order cost, lost sales cost and theft cost. In [135], the authors divide the costs into raw materials, production labor, production expenses, production overheads, finance and service, personal and administration, and distribution costs. Other researchers also consider, e.g., the visibility of occurring costs and distinguish between visible and invisible costs. Whereby visible costs can be directly quantified into monetary terms, and invisible costs are referred to as hidden opportunity costs [25]. In [118], the authors develop a supply chain cost model including penalty costs, e.g., for the failure of delivery.

2.6. Novelty of the trust-based resource sharing approach

As already mentioned, resource sharing between manufacturing companies has already been investigated from different aspects. Table 7 summarizes the models taken from the literature to compare them with the approach presented in this thesis. In the table, the letters in the third column mean the following:

a) trustfulness is not considered,
b) requests are not divisible,
c) resource constraints of participants are not considered,
d) service requesters and offerors are two separated groups,
e) costs and incomes are not considered (financial model is missing).

Table 7. Summary of the resource sharing related literature

Reference	**Focus of paper**	**Main differences compared to the resource sharing approach presented here**
[22]	Matching costs for participants and the platform	a, b, c, d
[23]	Multi-criteria decision-making algorithm	c, e
[26]	Stability of request-offer matching in crowdsourced manufacturing	a, b, c, e
[27]	Strategy proofness of resource matching in crowdsourced manufacturing	a
[75]	Details of resource sharing algorithm	a, b, e
[116]	Robustness of capacity allocation	a, e

As one can see based in Table 7, the main difference in the proposed mechanism compared to the others already presented in the literature is the inclusion of trustfulness in resource sharing, the consideration of resource constraints of the participants, and the financial model. It is also important to highlight that the purpose of the trust and reputation model presented in this thesis is to show how considering trustfulness can increase the efficiency of resource sharing and the performance of the cooperating partners.

3. Direct exchange-based resource sharing

In this chapter, a resource-sharing mechanism is introduced where the collaborating companies communicate directly with each other about outsourcing certain jobs. First, expressions used in the model description are defined for easier understanding. Then, the model and the communication mechanism are described, and the calculation of available resources and the decision-making logic are also detailed.

3.1. Basic definitions of the model

A *company* has a certain amount of different types of resources. It can communicate with other companies (offer its resources and send resource requests to others when having extra resources or shortages). On the basis of its decision mechanism, it can choose the best from the received offers.

A *federation* is a group of companies. Companies are allowed to enter or exit the federation at any time: the entry condition is to accept the interaction protocol and pay the entry fee. Collaboration from the model perspective is only possible between federation members.

The *Federation Center* (FC) manages entries and exits from the federation, updates the list of federation members, and calculates reputation values for each member.

A *lead company* is the one that receives the customer order from outside the federation. The nomination is used to highlight the company that is consolidating different jobs (insourced and outsourced).

Companies receive customer orders from outside the federation. One *order* represents several – in this stage of research, independent – *jobs*, which are determined by their resource requirements: type (e.g., CNC machine), quantity (e.g., 3 pieces), the number of products in the job, and the earliest start time and due date. To complete an order, all the jobs included in it must be completed by a) the company that received the order, or b) a company for which the job was outsourced. The following assumptions are taken: to fulfil a job, the resource load of the job has to be provided, which is calculated by multiplying the required resource quantity with the difference between the due date and the earliest start time. This means, with more available resources, the job could be completed in less time.

A *request* is sent to ask for free resources when having shortages (missing resource type or amount): it consists of the technical information (anonymized CAD files, material, tolerances, and manufacturing method) of the product to be manufactured – which determines the machine type(s) that the manufacturer have to own. A request also contains the capacity requirements of the job to be outsourced and the information about whether the requested capacity can be divided, and if yes, the maximum number of fragments the job can be divided into. There is also a minimum reputation value for the companies whose offer can be accepted to complete a specific request (trust and reputation is detailed in Chapter 5).

An *offer* includes the resource type, time interval, and the amount of offered capacity. It is also important to define a minimum amount of capacity that can be used if a requester company does not need all of it. Offers are sent to requester companies in response to a request.

For example, an order can be to produce 100 windows, and specific types of equipment are needed to manufacture the glass, metal and plastic parts. One job is to manufacture the 100 handles, and another is the production of the glass plates. If a company that received the order has a shortage from a certain resource type required to keep the delivery deadline or receives an order that requires a specific resource type that the company does not have (e.g., 3D printer to manufacture a complex part), it sends a resource request to the other federation members.

A *contract* is an agreement between two companies by accepting a matching offer-request pair. It describes the time interval when one company uses the other company's capacities, the payment, possible penalty and cancellation conditions. A company may terminate an already signed contract or withdraw an offer or a request already announced: this is penalized from trustfulness and from a monetary perspective as well, as it is discussed in the following chapters.

3.2. Model description

The flowchart of the agent interaction applied in the model is presented in Figure 10. When an agent receives an order from outside the federation (1), it performs the capability check (see Section 3.3). If the result is true, the agent schedules the job for itself based on the earliest job start time and due date – and performs it between these two time points. If the result is false, the agent checks the federation member list (updated after each entry or exit by the FC) and sends requests to all the other agents of the federation immediately (2). It is necessary to send the request to all the federation members because agents do not have any information about each other's resource types or amounts. After receiving an order, agents perform the capability check on their own production plan and send an offer to the requesting agent if the result is true – otherwise, they send a reject message about offering their resources. The requesting agent expects some kind of answer from each of the other agents in one model time unit – an offer is technically a feedback that the offering agent is able to complete the specific request.

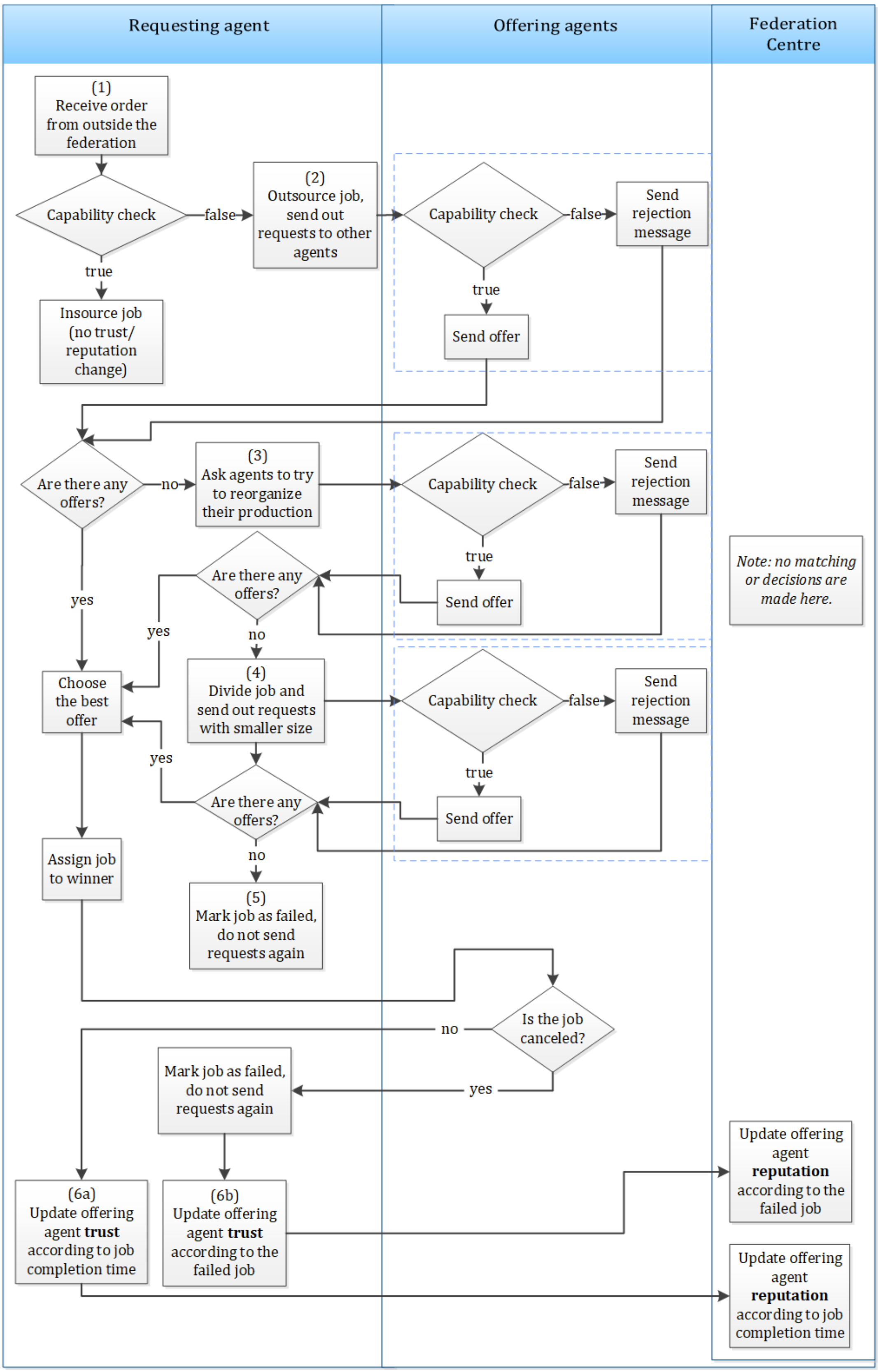

Figure 10. Swim lane of the distributed resource sharing model

If the requesting agent does not receive any offer from the other agents in the specified time window, it sends out the request again, and in parallel, asks all the other agents to try to reorganize their production with the aim of completing the specific job (3). They check their production and free resources again and if it is possible to complete the job after reorganizing, and they send back an offer or a reject message. If there are still no offers, the requesting agent divides the job into equal parts (number of parts is determined by the requester), sends out its parts separately as requests, and waits another time unit for offers (4). If the requesting agent does not receive offers for some of the parts, it marks them as "failed", and does not try to send request(s) associated with this job again (5). If there is at least one offer after steps (2), (3) or (4), the requesting agent chooses the best (or the only) offer or offer combination and assigns the job to the winner(s). In Figure 10, the frames in blue dashed lines are the same steps that the agent performs when receiving a request at different phases of the interaction.

A job being finished or cancelled, the requesting agent updates the winner agent's (subjective) trust value, and the FC updates the winner's (public) reputation value depending on whether the agent completed the job or cancelled it (6a and 6b). If the job is completed, trust and reputation values change according to the lateness in the due dates (detailed in Chapter 5). If the job is cancelled, trust and reputation values are recalculated by assigning a zero value to this unsuccessful interaction. In this case, the requesting agent does not try to find a new offer for this job and marks it as failed; similarly, as in step (5). If a (part of a) job is marked as failed, the requesting agent does not try to send it out again.

3.3. Calculation of available resources

In the model, one crucial point is how the companies determine their capability for being able to complete a job, taking their work in process (WIP) or already planned work into account. In Figure 11, a simple example is shown, where job_i and job_j are already planned jobs with t_i and t_j processing intervals (i.e., given by start and end times), and r_i and r_j resource intensities. The overlapping time intervals with t_{new} is $o_{i,new}$ and $o_{j,new}$. Here, the area of the hatched rectangle is equal to the possible maximum resource load the agent can provide in the t_{new} interval. The agent subtracts the overlapping area of the two light blue rectangles (planned jobs) from the area of the hatched rectangle (maximum resource load), and if the difference is higher than the area of the green rectangle (new job), the agent has enough resources to perform job_{new}. To have a manageable model, setup time is also included in the processing interval.

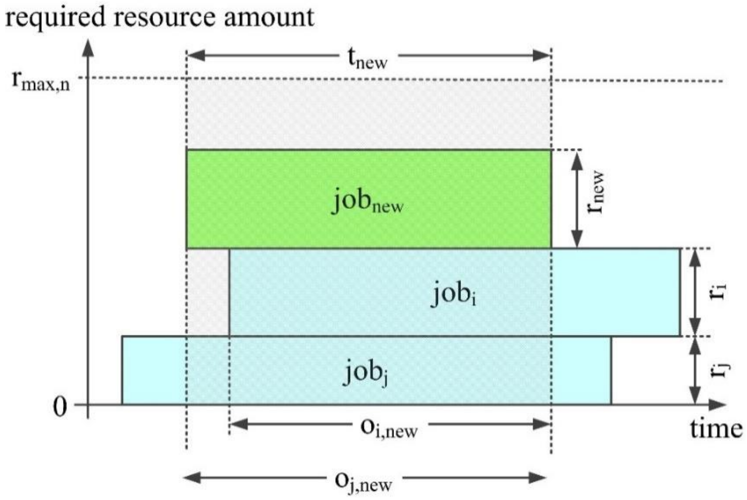

Figure 11. Calculation of available resources in case of a new job.

3.4. Choosing between offers

It should be highlighted that in the presented model, choosing between offers is done by the individual companies. They decide between offers based on price, the actual trustfulness rating of the offeror. In the case of each received offer *o* (sent by company *B*, about job *j*), requesting company *A* determines a fitness value ($F_{j,o}^{A,B}$) by calculating the weighted sum of the actual rating of B ($rating_B$) and the price of the offer sent in connection with *o* ($price_{j,o}^{B}$). Weights $w_{rating}{}_{j}^{A}$ and $w_{price}{}_{j}^{A}$ (both are positive values) are determined by the requester company; this way they can be modelled with a preference of price or trustfulness. $rating_B$ is the weighted average (weights are company-specific and constant) of cumulated *trust* and *reputation* scores which are calculated based on Eq. (5).

$$\boldsymbol{F_{j,o}^{A,B} = w_{rating}{}_{j}^{A} \cdot rating_B + w_{price}{}_{j}^{A} \cdot price_{j,o}^{B}} \qquad \text{Eq. (1)}$$

As one can see, the model allows multiple offers for a job from a single company. This may occur if the offeror company can carry out the work with, for example, resources with different operating costs, or resulting in different quality, and offers them at separate prices.

4. Platform-based resource sharing

In this chapter, the platform-based resource sharing mechanism is introduced: basic concepts are discussed similarly to the previous chapter. The communication mechanism and the role of the platform are also introduced here.

4.1. Definitions and model description

In the description of the platform-based resource sharing mechanism, the same definitions are used as in the case of the direct exchange-based mechanism. When a definition of an expression is different, this is highlighted separately.

In this case, a *company* is sending the offers and information about free resources to a central platform instead of directly to the other companies.

The *platform* is a central unit providing its services to a group of collaborating manufacturing companies which are the members of the federation. Companies are allowed to join or quit the federation at any time. It has a larger role in the collaboration of companies than the Federation Center: its functionality is discussed in detail in Section 4.2.

Offers are sent regularly to the platform by the federation members, and *requests* are also sent to the platform instead of directly to each other. In this case, a request sent to the platform also contains information about the number of (not necessarily equally sized) fragments it can be divided into.

From the *contract* perspective, it is also signed through the platform to be able to track changes that companies may make compared to it.

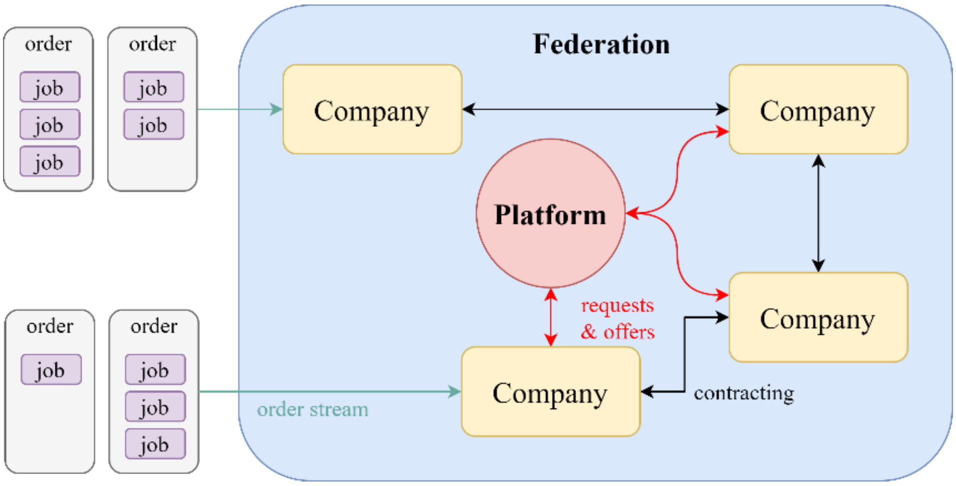

Figure 12. Platform-based resource sharing

In the platform-based resource sharing mechanism, the calculation of available resources and the decision between offers is made in a similar way as described in Sections 3.3 and 3.4. Due to the advanced matching logic of the platform (it can combine offers), the only difference in the companys' decision-making logic is that they determine the same fitness value as introduced in Eq. (1) for each offer and use their weighted average to calculate the fitness of a specific combination. Here, offer weights are determined based on the resource load (requested time multiplied by the resource quantity) of the offer, this way taking a larger offer with a larger weight into account.

One novelty of the resource sharing approaches presented in the thesis is that collaboration is not between geographic locations of the same company or between different manufacturing service providers (as in Cloud Manufacturing). In contrast, the collaborating partners are on the same level of value creation, and their role (resource offeror or requester) depends on the specific interaction. Here, Build-To-Order (BTO) companies, who are members of the federation, receive orders from customers outside the federation (see Figure 13). From an information flow perspective, the platform is in the center of the federation and receives resource requests and offers (that companies can send in case of having underutilized resources) to match them: this way, helping companies with shortages and extra resources also. The platform could also combine offers from different companies to fulfil a request (this way, a job could be completed by different companies), and sends the list of appropriate offers or offer combinations to the company that received the order from the customer. The presented approach is placed between decentralized and centralized production methods because the decision making (choosing between resource offers) is made by the participants locally, but the distribution of resources is supported by a central platform whose role is to pre-filter and combine resource offers from which the requester can choose from.

Figure 13 depicts the material and information flow between the participants of the federation. The costs (blue colour) and incomes (green colour) are also visualized in this figure and are discussed in detail in Chapter 9.

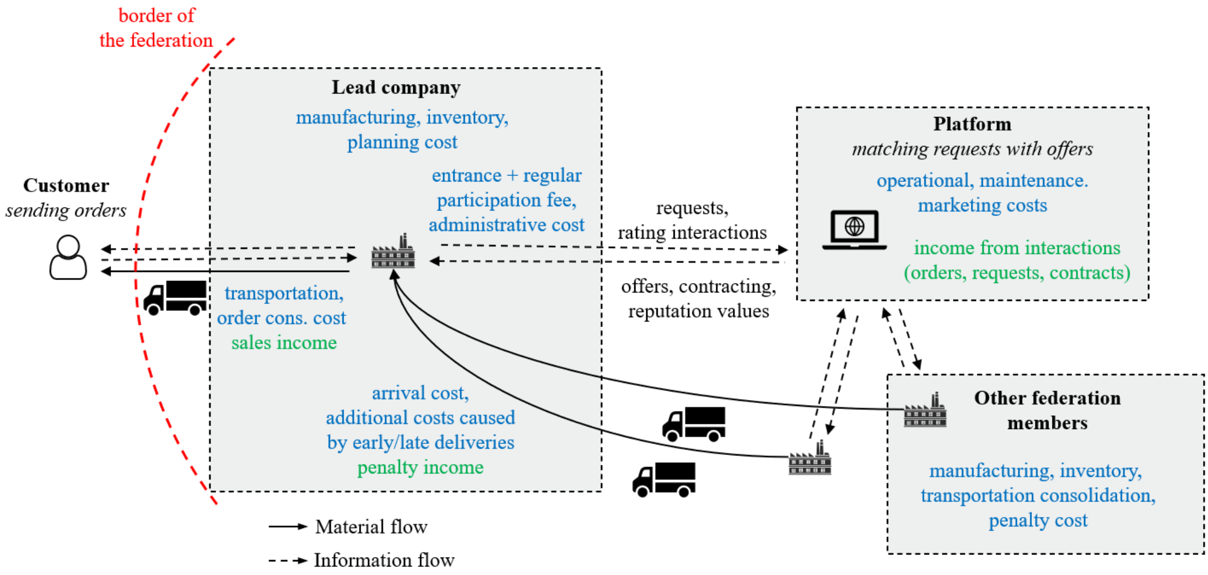

Figure 13. Material and information flow in the federation with the costs and incomes of the participants

One can distinguish between the initial and the operation phase in connection with a resource sharing platform. At the initial phase, the main goal is to attract as many participants as possible to establish matching of resources (provide the necessary number of offers for the requesting companies and vice versa). This thesis focuses on the operation phase, the platform must ensure continuous communication and interactions between participants.

4.2. Platform functionality

The most important goal of the platform is to provide a request-offer match for outsourcing and capacity sharing companies, supporting better resource utilization of its members. It provides access to a continuously changing capacity sharing community; the platform operates as a dynamic augmentation of the company, which can be extremely useful in the case of fluctuating and unforeseen customer orders. It can recommend offers coming from previously unknown companies, providing the requesting participants a higher chance to find the best (if any) offer. It minimizes the disclosure of capacity-related information to prevent companies from querying and planning on the basis of all of the other companies' capacities. Sustainability also could be addressed by recommending offers with lower environmental impact (closer partner, fewer logistics), and, as it has a global view of the members, by optimizing logistics as well. It could happen that a single offering company does not have the required number of resources to satisfy a request: in this case, the request could be fulfilled by fragmenting and combining capacity offers from different companies.

It is important to mention that the goal for the platform is not to select the best solution but to pre-filter the offers that are meeting the resource constraints of a request, by integrating reputation values as well. This way, the platform limits the communication and decision space, and expands it at the same time since it can find suitable offer combinations. Companies could also set a minimum reputation value below which they do not want to receive offers. The platform considers all the offers sent by companies above this reputation level in the matching process but leaves the decision to the requester company, who takes the prices into account in decision making, as well.

A company might outsource a job even though it has the appropriate capacity to finish it, if it notices that an offer made through the platform is less expensive than using its own capacities. It is assumed that the companies are honest with each other and with the platform as well: they do not try to manipulate the system by sending false messages or providing false trust values.

The communication mechanism of platform-based resource sharing can be seen in Figure 14 (the blocks related to rating the partner’s performance are discussed in detail in Chapter 5.).

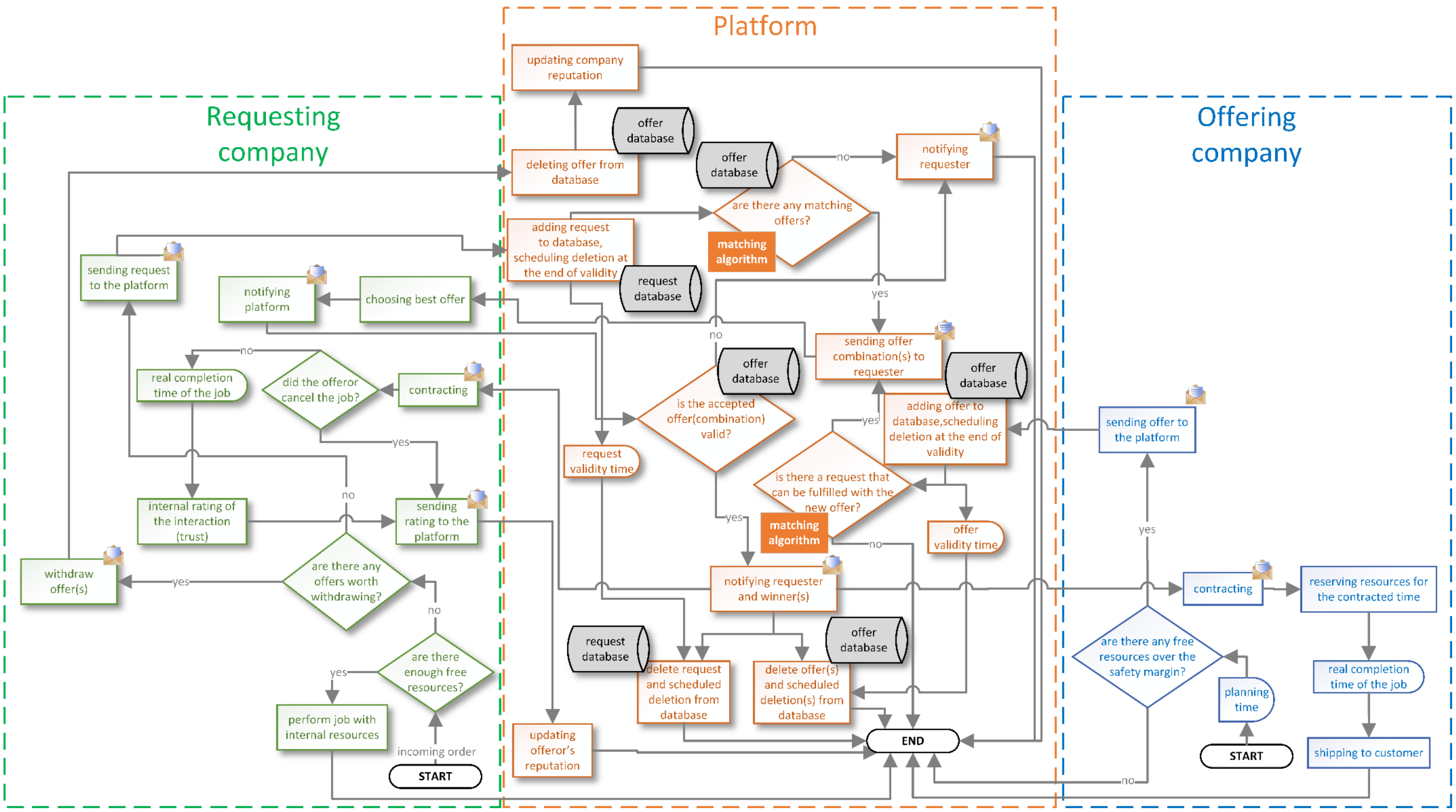

Figure 14. Communication mechanism for platform-based resource sharing

The matching logic of the platform operates based on the logic introduced in Section 3.3: the platform tries to cover the resource load (area of the rectangle in Figure 11) of the incoming request with the suitable offers (same resource type, at least partly overlapping time interval, minimum reputation level is reached by the offeror).

The aim of the matching function is to find (combinations of) offers that are suitable for the request and create all the appropriate offer combinations. As Figure 15 depicts, the goal is to fulfil the request with the least possible number of offers in order to reduce administrative and logistics costs; in the model, the maximum number of offers that the Platform can combine is three. If the platform finds at least one appropriate combination, sends it (or them) to the requester who is going to choose based on its own preferences. If there is no match, the platform adds the request to the list of not matched requests and will try to find a suitable offer for it later.

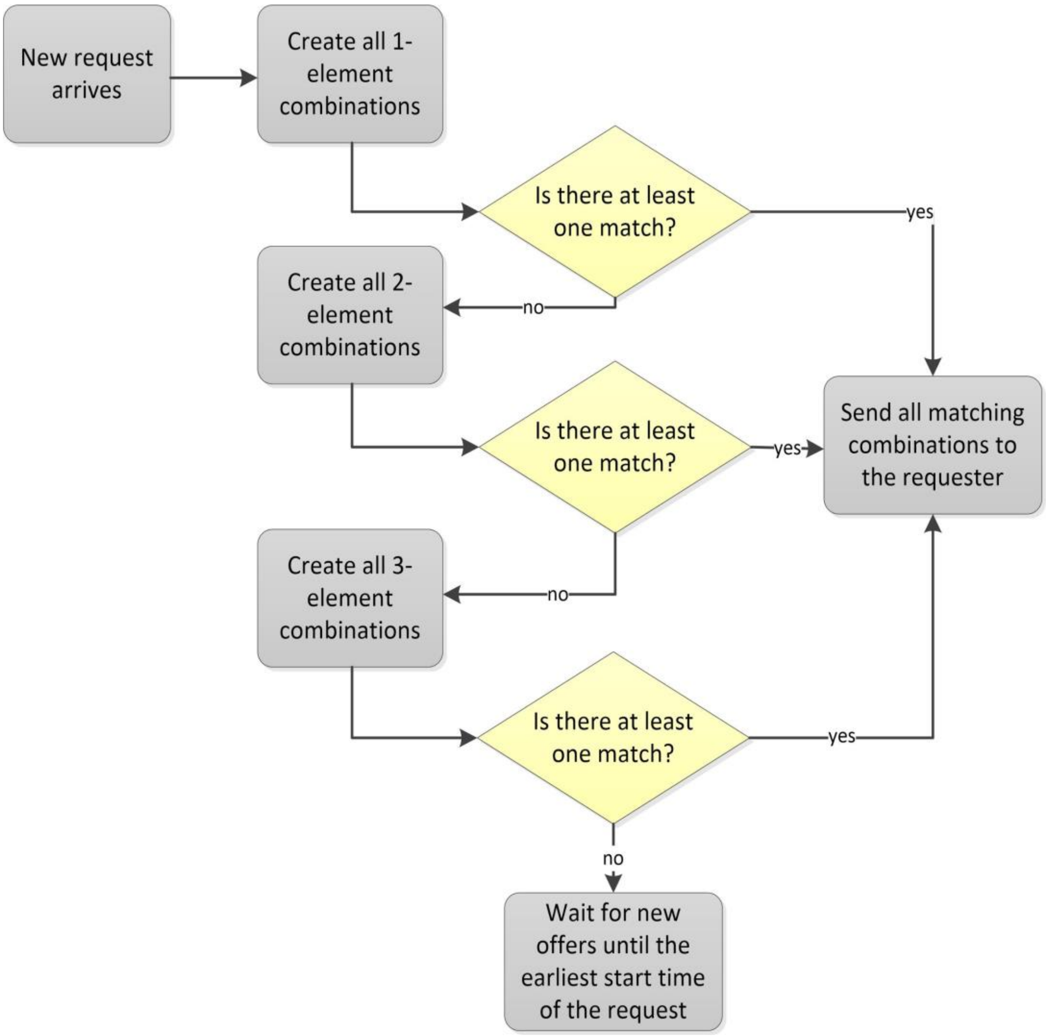

Figure 15. Combining offers with the aim of finding a match in the platform

The communication mechanism depicted in Figure 14 is implemented in the agent-based simulation model, which is described in Chapter 6, where a very detailed description of the whole process is given.

For the proper working of the presented model, some additional assumptions have to be made. It is assumed that each company is able to create good enough forecasts about its production, enabling to make decisions based on them – e.g., by using a simulation model. In addition, compatibility and quality check of companies/equipment are important for quality assurance purposes: production plants must ascertain the quality of the resources shared by other companies in order to use them with minimal risks. Safety margins for the resources are considered as well: companies do not offer all of their free resources. They might have different KPIs and decision criteria – however, the local decision-making mechanism of the companies is not in focus, only a simple function is provided. The platform and the companies have pre-defined minimum response times, as well as a maximum time allowed to react to a message coming from the platform. Minimum response time is necessary because each company should create forecasts to make the right decision, and the platform also has to run its matching algorithm. If a company does not react in the maximum response time, the platform assumes that it does not want to accept any of the offers. The platform also checks the validity of offers; it can happen that the platform has sent a message about a specific offer to more than one company, and all of them are planning to accept it. In this case, the first one who sends the acceptance message gets the resources, and the second receives a message about invalid offers. Then, the match-making mechanism tries to find another offers for the second company.

5. Trust and reputation model

In the presented new model, the main purpose of taking trustfulness into account is to show it can improve the efficiency of resource sharing and the performance of the cooperating partners. As previously mentioned, in the manufacturing area, the most important aspects are keeping promises and delivery deadlines – these are the main focus points of the presented model, too.

In the resource sharing mechanism, trustfulness is included by using two ratings calculated on the 0..100 scale, called

1. ***trust*** (which is an internal rating about a specific partner, similar to an own opinion) and
2. ***reputation*** (which is a public rating, aggregating all the ratings sent by partners about a specific company, and updated by the central unit in the federation).

The bases of the ratings are:

a. the percentage inaccuracy with the deliveries, and
b. contract cancellations and offer withdrawals.

When companies are sharing resources directly with each other, both ratings are considered in decision-making. In contrast, when applying the platform-based method, the identity of the offeror is revealed only after choosing the winner offer(s); companies rely only on the reputation values provided by the platform. This is necessary to prevent companies from discovering each other's resources with the aim of creating a competitive advantage. (Note: a more detailed comparison of the two model is provided in Chapter 8).

5.1. Considering delivery accuracy

In [51], the authors define the delivery window as the difference between the earliest acceptable delivery date and the latest acceptable delivery date, as shown in Figure 16. Here, ${t_{sa}}_j{}^B$ is the start and ${t_{ea}}_j{}^B$ is the end of the acceptable delivery time interval of the part of job j that was outsourced to agent B, and ${t_{so}}_j{}^B$ is the start, ${t_{eo}}_j{}^B$ is the end of the on-time delivery interval where no penalty is issued, neither in terms of trustfulness nor cost perspective.

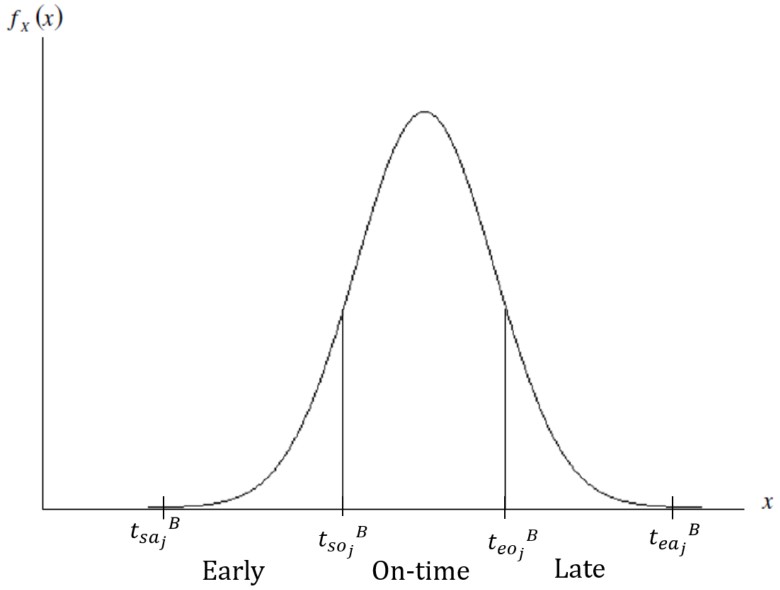

Figure 16. Delivery window based on [51]

In the presented trust model, the $r_{t_r,j}^{A,B}$ score given in time point t_r about a specific interaction (between lead company A and resource offeror company B, about job j) is calculated on the following way. If the delivery is *on-time*, $r_{t_r,j}^{A,B} = 100$, i.e., company B gets the maximum possible rating. If the delivery is *early* or *late* (arrives inside the *delivery acceptance* interval, but outside the *on-time* interval), the rating of the interaction is computed based on the extent of earliness or lateness that is nominated with δ_j. If the delivery is early, $\delta_j = t_{so_j}{}^B - t_{d_j}{}^B$, if it is late $\delta_j = t_{d_j}{}^B - t_{eo_j}{}^B$. In these cases, the rating is calculated as follows:

$$\text{if } \boldsymbol{\delta_j < L_j} \quad \boldsymbol{r_{t_r,j}^{A,B} = 100\left(1 - \frac{\delta}{L_j}\right) \cdot \gamma \cdot \mu_j} \ \textbf{where}$$
$$\boldsymbol{0 < r_{t_r,j}^{A,B} < 100,\ 0 < \gamma < 1,\ 0 < \mu_j < 1} \qquad \text{Eq. (2)}$$

$$\text{if } \boldsymbol{\delta_j \geq L_j} \qquad \boldsymbol{r_{t_r,j}^{A,B} = 0} \qquad \text{Eq. (3)}$$

Where L_j is the length of the job in time, γ is the penalty factor applied on the federation level to penalize inaccurate deliveries to a greater extent, and μ_j is the quality factor that makes it possible to rate the delivery accuracy and the quality of the resource offeror's work about job j. If the delivery arrives outside the acceptance interval, the job is marked as failed, and a rating equal to zero is given.

5.2. Choosing between resource offers

Choosing between offers is based on trustfulness and price; thus, the trust and reputation scores are cumulated before choosing from different resource offers. To assign smaller weights to older feedbacks, a modified exponential smoothing is applied, similarly to different trust and reputation systems. The $w(t_r, t)$ weight – that is assigned to a score given in time point t_r to calculate the cumulative score in time point t – is calculated as follows, where θ is the decay factor used to affect the shape of the function:

$$w(t_r, t) = \frac{\theta}{\theta + (t - t_r)} \qquad \text{Eq. (4)}$$

The cumulative score (both trust and reputation) in time point t is calculated according to Eq. (5), where all the scores given earlier are included:

$$\varphi^{A,B}(t) = \frac{\sum_{t_r \leq t} w(t_r, t) \cdot r_{t_r,j}^{A,B}}{\sum_{t_r \leq t} w(t_r, t)} \qquad \text{Eq. (5)}$$

The score described in Eq. (2) and Eq. (3) is created after each interaction, and 1) sent to the central unit of the federation to be able to update the offeror company's reputation based on Eq. (4) and Eq. (5), and 2) the internal trust value can be also updated based on this score on the lead company side. When choosing between offers, a company could take the cumulative trust and cumulative reputation (provided by the central unit) of the offeror company into consideration, also, in addition to the price of the offer.

This way, a company could be penalized for bad performance, but its low rating also could be changed in the long term, in case of improvement in delivery or quality accuracy. It can also happen that a company withdraws an offer or a request that has been already sent to the platform before matching. This is penalized from the trust and reputation perspective with a percentage decrease (model parameter) of the cumulated value. Companies can also cancel a contract which is penalized in the same way.

Another important aspect is the honesty of federation members. Here it is supposed that the companies are providing ratings honestly and do not try to influence other companies' decision making by giving lower ratings to a partner than it deserves.

6. Agent-based simulation model

To validate the model and to be able to run experiments, companies and the platform are modelled with agent-based simulation in AnyLogic. In this chapter, the different agent types, their parameters, building blocks and functions are introduced in detail. The additional Java classes (e.g., requests, offers, equipment) used in the model are described in the Appendix.

In the model, only discrete resources are considered, and orders are processed by companies on a First Come First Served (FCFS) basis. Setup times are not explicitly modelled; they are included in the processing times of the jobs. Distance between companies and their different resource types are also considered, but modelling of maintenance periods, shifts, premises, and buffers are neglected. As mentioned above, it is assumed that the participants are honest and do not try to manipulate other agents by sending false messages. Security issues of the communication mechanisms are also not focused points in the model.

In general, the model operates based on the communication mechanism depicted in Figure 14. When running the model, all companies – modelled with agents – receive an order stream generated within the model. According to their available resources, the companies insource or outsource a specific part of the order, send messages to the platform, and perform different functions triggered by specific events (for example, the offer-making function is called when some resources are released). The platform is also modelled as a separate agent, which communicates with the federation members.

According to [96], in an agent-based model, each agent must have one or more clear goals to be accomplished through specific actions, driven by decisional rules. These are also detailed in this chapter. In the presented simulation model, the following agent types are present:

- Main agent (no communication, only the frame of the simulation model)
- Company agents
- Platform agent.

6.1. Main agent and model parameters

In Figure 17, one can see the Main agent, which is the frame of the model. Functions in connection with initializing the model, general model parameters, statistics variables, diagrams and the built-in Geographic Information System (GIS) map, where the agents are placed, are included here. At the beginning of the model run, an *OnStartup* function is called to initialize the model: it creates a population of company agents using their parameters (that is discussed in the following section) inserted into an external database.

oMF
root.oMF

facilities
Facility [10]

connections
not connected

Initialization

modelPath
C /ADAM/SZTAKI/AnyLogic Model...

eq_types
{20}

f
File name ../Facilities.xls...

WIP_effect
false

general_delay
0.2

federation_size
10

Model parameters and functions

default_trust
80

alfa
0.1

setAlfa

order_frequency
1.5

avg_order_interval
20

avg_order_capacity
800

eq_num
20

offered_int
20

offer_rep_change
1

default_rep
80

gamma
2

avg_order_delay
3

order_delay_dev
0.2

order_interval_dev
0.25

order_capacity_dev
0.3

completion_dev
0.5

delay_sigma
0.1

penalty
0.9

planninghorizon
2

Statistics

failedRequests
{0}

sum_orders_generated
3,530

sum_insourced
2,024

sum_partlyoutsourced
0

sum_outsourced
1,506

sum_loadCanceled
384,304.762

nocap
1,018

noeq
488

sum_message
12,033

notify_winner
1,916

notify_OMF
1,301

offer_sent
7,310

cancelling
37

statistics
501 samples [49.933...64.037]. Mean=57.402

sum_partlyCanceled
37

sum_completed
2,565

sum_failed
202

facility_coll
{10}

fed_cap_sum
175,401.6

fed_cap_util_current
98,299.083

sum_delay
173,529.085

avg_delay
94.721

sum_capreleases
1,832

reqID
7,311

Figure 17. Main agent interface in AnyLogic

The following model parameters were defined that can affect the model behaviour. When the value of a parameter is generated from a truncated normal distribution, the difference between the lower and upper bounds of the distribution from its mean is half of its expected value.

General parameters:

- **fin_interval** – length of the "on-time" time interval described in Chapter 5.1
- **acc_interval** – length of the "acceptance" time interval described in Chapter 5.1
- **avg_prodNum / prodNum_dev** – average (expected) value / deviation of the truncated normal distribution from which the number of products in a job is generated.
- **completion_dev** – deviation of the truncated normal distribution from which the real processing intervals are generated for *insourced jobs* (in percentage).
- **delay_sigma** – deviation of the truncated normal distribution from which the real processing intervals are generated for *outsourced jobs* (in percentage).

- **planninghorizon** – interval in model time units; the companies check whether they have free resources until the end of this interval and send offers to the platform accordingly.
- **offered_int** – interval in model time units; the companies send offers about their free resources to the platform with this length.

Order generation parameters:

- **avg_order_interval / order_interval_dev** – average (expected) / deviation value for the truncated normal distribution from which the *processing times* of customer orders are generated.
- **avg_order_capacity** / **order_capacity_dev** – average (expected) value / deviation for the truncated normal distribution from which the *resource loads* of customer orders are generated.
- **avg_order_delay** / **order_delay_dev** – average (expected) value / deviation of the truncated normal distribution from which the difference between the time point of the order arrival and its earliest start time is generated.
- **order_complexity** – number of jobs included in an order (to complete the order, all the jobs have to be finished in time).
- **eq_num** – number of equipment (resource) types that are present in the model; used to generate random customer orders.

Parameters for investigating the effect of trustfulness:

- **default_trust**, **default_rep** – initial value for trust and reputation for all the companies. As it is discussed in Chapter 5, the computation of these ratings is based on former interactions, thus an initial value has to be defined for both of them.
- **gamma** – γ, used to calculate new reputation values according to Eq. (2)
- **penalty** – penalty factor; the reputation value of a company is multiplied with this after cancelling an undertaken job.
- **offer_rep_change** – the reputation value of a company is multiplied with this number if it withdraws an offer before matching.

Parameters for investigating financial aspects:

- **initMoney** – initial money for modelled companies.
- **distanceAC** – average distance between lead company and customer is kms to calculate transportation costs.
- **entryFee** – one-time entry fee for companies that are joining the federation.

- **regularFee** – regular participation fee for federation members.
- **offerCost** – cost of sending an offer to the platform.
- **requestCost** – cost of sending a request to the platform.
- **contractCost** – cost of contracting, incurring for the requester company.

In Figure 17, with orange background one can also see several statistical variables that are used as an input to create diagrams for the experiments. During the model run, data is continuously collected, and in the experiments, the change of some KPIs over time, and some average values are also represented to compare different scenarios.

6.2. Company agent

The goals of this agent type are to:

- Maximize its own resource utilization and profit.
- Minimize penalty costs occurring because inaccurate delivery.

Action items are to:

- Send offers and request to the platform (or in the direct exchange-based case, to other companies).
- Select external resources according to own preferences (e.g., lowest price, highest reputation).
- Rate other companies based on their performance.
- Create and continuously update own production plan based on incoming orders, accepted offers and available resources.

Each company is modelled with a separate agent, as a company is an autonomous decision-maker who can communicate with other companies and make decisions on its own. It has several parameters in the model, as one can see in Figure 18 with purple background – as mentioned above, these parameters are imported from an external database on model startup.

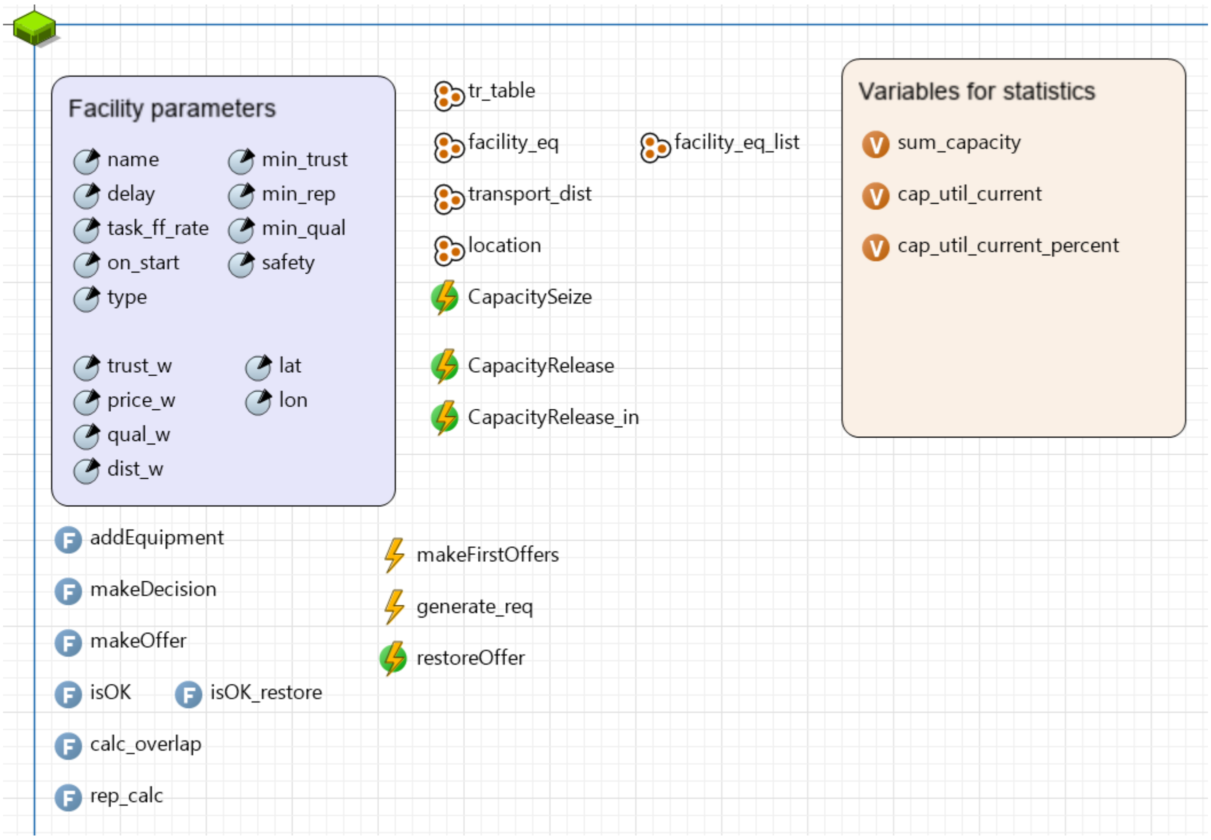

Figure 18. Company agent in AnyLogic

- **name** – name of the company
- **order frequency** – time interval (in model time units) elapsed between customer orders coming from outside the federation for the agent.
- **delay** – difference between the length of the time interval required to fulfil a customer order, and the expected value of the truncated normal distribution from which the *real* time interval is generated for the jobs the specific company accepts. This parameter is the main cause for delayed job completions that are penalized from trustfulness and cost aspect, too.
- **task_ff_rate** – task fulfilment rate, which means the percentage of jobs that the company does not cancel on average.
- **min_rep** – minimum reputation value for offers that the company accepts.
- **safety** – safety margin for offering resources. E.g., 20% means the company offers max. 80% of its resources and keeps 20% for internal jobs (even if they remain unused at the end).
- **lat / lon** – latitude and longitude coordinates of the company location.
- **trust_w** – weight of trustfulness in the weighted sum of decision-making function.
- **price_w** – weight of price in the weighted sum of decision-making function.

- **qual_w** – weight of quality in the weighted sum of decision-making function.

The following parameters were only used in specific test experiment types that are not detailed in this thesis – however, it is possible to use them for further investigations:

- **dist_w** – weight of distance in the weighted sum of the decision-making function.
- **min_qual** – minimum quality level considered by requesting companies when choosing from offers.
- **min_trust** – minimum trust level considered by requesting companies when choosing from offers.
- **on_start** – marks if the company joins the federation at the beginning of the simulation run or not. Used for experiments where the effect of a company entering/exiting later was tested.
- **type** – used for specific experiments to mark different company types, where companies with different preferences (e.g., preferring trustfulness in contrast to price) were investigated.

In the model, a company agent can call several functions that are triggered by specific events (e.g., message or customer order arrival), which are introduced in detail in the following.

Generating customer orders

The incoming customer order flow is modelled by regularly generating orders for each company through the **generate_req** event. During this event, an **Order** instance is generated, and it is investigated (using the functions described earlier) whether the company can perform the jobs in the **Order**, or not. A **Request** is generated for each job in the **Order**, and for each **Request**, the company performs the **isOK** function (described below) to determine whether it has to outsource it or not. It can happen that the company can complete only a part of the request; in this case, a job is divided into two **Request** instances; one of them is insourced, and the other is outsourced.
The time interval between generating requests and their other parameters is determined by the general simulation parameters, described in the **Main** agent.

addEquipment is called when initiating the simulation; it adds the specific equipment (resource) types of the company to a LinkedHashMap data structure called *company_eq,* in order to reference it in an easier way (a specific equipment instance can be reached by its name).

Choosing between offers

makeDecision is called when the company receives a message from the Platform, containing offer(s) in response to a request sent earlier. The function argument is a *MatchingOffers* instance, which contains all the matching offers or offer combinations, and the output is the winning offer combination. For decision-making, a company uses a weighted sum of trustfulness and price to determine the best option. (Note: distance could also be used to decrease the environmental impact of logistics: this way, for example, by giving this factor a higher weight, more environmentally friendly agents could be defined. In fact, any function could be applied here – for example, a more complex multi-criteria decision-making approach could also be used.)

Offering capacities to the Platform

makeOffer is called regularly, it checks whether the company has free resources until the end of the interval *planninghorizon* defined in the main agent. The function iterates through the available resources and sends the offers to the Platform, if any, by taking the safety margin into consideration. Checking resources means calling the *isOK* function, which is detailed in the next paragraph. Technically, when sending the offer, the company inserts a virtual request (which load equals with the offered load) to its production plan (which consist of already undertaken requests) to avoid accepting an order coming from outside the federation that require the already offered resources. This virtual load is marked with a unique ID to delete it in an easier way if the offer is out-of-date or some part of the offered capacity is accepted by another company. In the latter case, the virtual load has to be replaced with the accepted load.

Checking available resources

isOK function is used in the decision whether a job in an incoming customer order can be fulfilled by the receiving company or not. By iterating through the planned requests using the **calc_overlap** function (detailed below), it returns an **IsOKAnswer** instance which contains information about the amount of work that can be insourced and the amount that has to be outsourced.

It can happen that a customer order cannot be fulfilled using available resources, and in this case, offers that are already sent to the Platform may be withdrawn. When checking available resources, these functions are also checking, and restoring these offers, if necessary (this also comes with a reputation loss).

calc_overlap calculates and returns the "overlapping" resource load between an incoming customer order and the already undertaken or planned requests on the basis of the logic depicted in Figure 11. Technically with this function, the already planned jobs can be summarized for the requested time interval and subtracting this load from the maximum load of the specific resource type returns the available resources for the requested interval. This way, it can be decided whether the company has enough resources to fulfil a request or not.

Starting and ending interactions

In the simulation model, dynamic events are **CapacitySeize**, **CapacityRelease** and **CapacityRelease**_in (_in means releasing resources used for jobs performed only internal resources). These events are changing the number of available resources when working on a job starts or ends. This is used for continuously recording resource utilization levels. When a job is completed, the financial interactions – detailed in Chapter 9 – are also performed in the above-mentioned functions.

Variables **sum_capacity**, **cap_util_current** and **cap_util_percent** are utilized to create statistics about the actual resource utilization levels of the companies, which is one of the most important KPIs of the system. These values are always changed when seizing or releasing capacities.

After releasing capacity in connection with offers accepted by another federation, the reputation value for the specific interaction is calculated, too – this is the input for the **rep_calc** function in the Platform agent.

Collections named **company_eq** and **company_eq_list** are applied in storing the Equipment instances that the specific company owns. The difference between the two collections is that the first is a LinkedHashMap, which allows reaching an equipment based on its name (key element class is string); the second one is an ArrayList used for easier iteration through all equipment types for specific functions.

6.3. Platform agent

The goals of this agent type are the following:

- Generate income during operation paid by companies.
- Receive and handle all incoming requests and offers from companies.
- Provide up-to-date reputation values to companies for easier decision-making.

Action items are:

- In case of an incoming request: not to select the best solution but to pre-filter the offers that are meeting the resource and reputation constraints and send them to the requester.
- In case of an incoming offer: to try to find matching requests by checking already received ones, in addition, to continuously monitor the incoming requests for a possible match.
- React to incoming messages (offers, requests) as quickly as possible and update the databases containing requests and offers continuously.
- Penalize companies which are withdrawing offers or already undertaken jobs, to create a reputable environment that the companies can use for planning.

To reach the above-mentioned goals and to execute the action items, the Platform agent performs pre-defined functions when receiving a message (request, offer, acceptance, job cancellation, offer withdrawal). Objects included in the Platform agent can be seen in Figure 19 and is discussed in the following paragraphs.

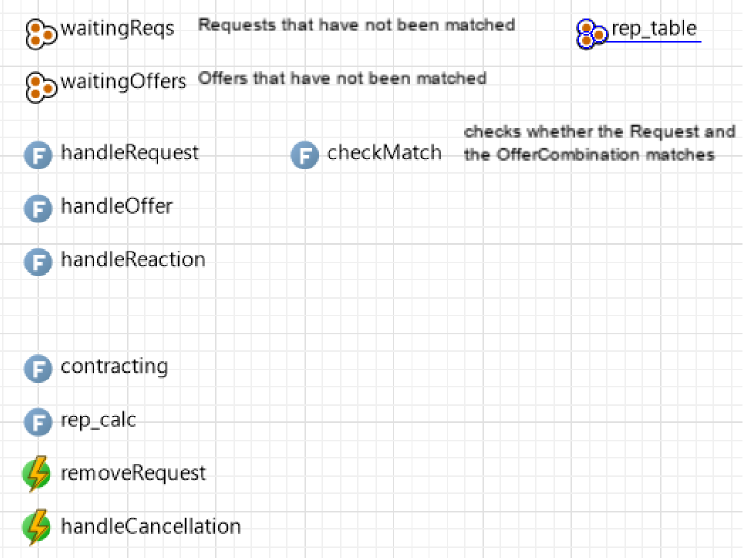

Figure 19. Platform agent in AnyLogic

Handling incoming requests

The **handleRequest** function is called when the Platform receives a request from one of the federation members: the flowchart of the function can be seen in Figure 15.

Handling incoming offers

Through the **handleOffer** function, when the Platform receives an offer, it also checks all the unmatched requests in order of their arrival, with the aim of finding one that could be fulfilled with the new offer. It might happen that the new offer does not fulfil any of the unmatched requests, but combining the new offer with other waiting offers does. Technically the same **handleRequest** function is called for each unmatched request each time an offer arrives – this places an additional computational load on the Platform, but also means one of the main advantages of joining it: a higher chance of matching.

Check matching offers

The **checkMatch** function iterates through the **OfferCombinations** created by the **handleRequest** function and checks whether the requested resource load could be fulfilled by the offers or not (marked with a yellow rhombus in Figure 15). This is made by calculating the overlapping areas between the offered resource loads and the requested load, similarly, as depicted in Figure 11.

Accepting offers

When the Platform receives an acceptance message from one of the companies (because one offer combination has been accepted by a requesting federation member), it checks whether the accepted offers are valid or not. It might happen that one offer is suitable for more than one request at the same time; in this case, the company that sends the acceptance message faster is able to accept the specific offer. The other(s) receive a message that the specific offer is not valid, and the Platform tries to find an appropriate one instead.

Contracting

If the offers in a specific **OfferCombination** instance (combination of appropriate offers) that have been accepted by a requester company are all valid when the Platform receives the acceptance message about them, the Platform performs the **Contracting** function. Accepting an **OfferCombination** means that all the offers in that are necessary to complete the request but does not mean that each of the offers are necessary entirely. The contracting function checks what resource load is necessary precisely and chooses the last offer from the **Offercombination** – from this one, only the necessary amount is contracted. For the remaining amount, an offer is generated to the database automatically.This function also iterates through all offers in the winner **OfferCombination** and schedules the **CapacitySeize** and **CapacityRelease** events that are changing the amount of available resources. In the simulation model, whether an offer is withdrawn or not is decided here according to the company parameters.

Reputation Calculation

Based on the reputation value given by the requesting agent about the offering one in connection with a specific interaction, the reputation value for the offering agent is updated. The calculation is made based on the method described in Chapter 5.

6.4. Modelling reliable and non-reliable companies

In the model, to mimic real life's stochastic nature as accurate as possible, real job processing times are determined by truncated normal distributions in order to simulate lead time prediction inaccuracy and generate a difference between expected (promised) and real job completion times. For example, if the expected completion time (i.e., the difference between the due date and earliest start time) is 20 model time units, then the mean of the distribution is 20 model time units. The deviation of the distribution is one of the parameters for some of the experiments, especially in the cases where the effects of prediction accuracy is investigated. In this case, when the expected (promised) processing time is equal to the mean of the distribution, the company is called *reliable.*

The above-mentioned case is visualized in Figure 20. In green colour. In the model, non-reliable companies exist, also: the distribution from which real processing times are generated for these companies is shifted with Δ. This parameter is the extent of the agent's non-reliability, given in the percentage of expected job completion time. For example, in the case shown in Figure 20, Δ = 20% for the agent marked with red colour, as the difference between the mean of the original and the shifted distribution is 4 time units. On one hand, these two parameters – deviation and Δ – determine how accurate is the lead time prediction of a company. On the other hand, as mentioned earlier, non-reliability could also come from cancelling jobs, and withdrawing offers that were already sent to the platform.

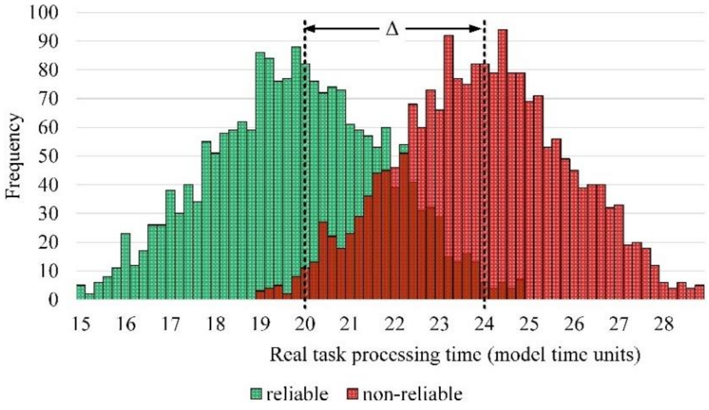

Figure 20. Generating real processing times for jobs in case of reliable and non-reliable companies

7. Simulation experiments – trustfulness in resource sharing

In order to validate the operation of the proposed mechanisms and to investigate the effect of considering trustfulness in resource sharing, a high-level multi-agent-based simulation model was built in AnyLogic, in this case for the *direct exchange-based mechanism*. In most of the experiments, the resource sharing mechanism described above has been implemented with ten company agents, with the aim of testing and validating the proposed protocol. As one can see in Table 8, six non-reliable (C01 to C06 marked with red background) and four reliable (C06 to C10 – marked with green background) company agents were considered. In the case where not only ten agents were investigated, the agents in Table 8 were duplicated (as detailed in the description of the specific experiment). In the experiments, 20 different resource types existed; the types and amounts are also listed in Table 8 (for example, C01 company agent has 10 units from resource type 1). Two of the non-reliable agents (C01 and C02) have all the twenty resource types, and the others have only ten of them.

Table 8. Resource capacities of the modelled companies

Resource type	**Company**									
	C01	**C02**	**C03**	**C04**	**C05**	**C06**	**C07**	**C08**	**C09**	**C10**
1	*10*	*10*	*10*		*10*				*10*	*10*
2	*12*	*12*	*12*		*12*				*12*	*12*
3	*10*	*10*	*10*		*10*				*10*	*10*
4	*12*	*12*	*12*		*12*				*12*	*12*
5	*10*	*10*	*10*		*10*				*10*	*10*
6	*8*	*8*	*8*		*8*				*8*	*8*
7	*9*	*9*	*9*		*9*				*9*	*9*
8	*10*	*10*	*10*		*10*				*10*	*10*
9	*11*	*11*	*11*		*11*				*11*	*11*
10	*12*	*12*	*12*		*12*				*12*	*12*
11	*10*	*10*		*10*		*10*	*10*	*10*		
12	*12*	*12*		*12*		*12*	*12*	*12*		
13	*10*	*10*		*10*		*10*	*10*	*10*		
14	*12*	*12*		*12*		*12*	*12*	*12*		
15	*10*	*10*		*10*		*10*	*10*	*10*		
16	*10*	*10*		*10*		*10*	*10*	*10*		
17	*12*	*12*		*12*		*12*	*12*	*12*		
18	*10*	*10*		*10*		*10*	*10*	*10*		
19	*12*	*12*		*12*		*12*	*12*	*12*		
20	*10*	*10*		*10*		*10*	*10*	*10*		

The model has several parameters; thus, for easier understanding, Table 9 contains the varied parameters, and Table 9 contains the parameters that were fixed in all experiments.

Table 9. Varied parameters in the experiments

Notation	Description	Unit
a_r	number of reliable agents	pcs
a_n	number of non-reliable agents	pcs
d	number of parts the agents can divide the requests into	pcs
f	agent flexibility	%
tr	the model includes trust (1) or not (0)	-
rep	the model includes reputation (1) or not (0)	-
$\boldsymbol{\theta_{tr}}$	decay factor for trust values	-
$\boldsymbol{\theta_{rep}}$	decay factor for reputation values	-
t_{order}	order interarrival time	model time unit

Table 10. Fixed parameters in the experiments

Notation	Description	Value	Unit
Δ_r	difference between the mean of the real processing interval and the original processing interval, in the case of *reliable* agents	5	%
Δ_n	difference between the mean of the real processing interval and the original processing interval, in the case of *non-reliable* agents	20	%
γ	penalty factor in case of delayed jobs	0.8	-
x_r	job cancelling rate in case of *reliable* agents	2	%
x_n	job cancelling rate in case of *non-reliable* agents	20	%
t_{avg}	average interval size of the jobs received from outside the federation	40	model time unit
r_{avg}	average amount of required resources for the jobs received from outside the federation	1000	-
u	resource unit price	100	-

In Table 11, the performed experiments are introduced, with the bounds of the varied parameters, which are marked with a grey background. If there are no grey background cells in the column of an experiment, that means a specific scenario –

detailed in the experiment description – was investigated. In the case of most of the experiments, the simulation was run for 750 model time units: after this period of time, the measured KPIs did not change significantly. In Experiment (8), the effect of an unexpected negative event is investigated, and here it was necessary to run the simulation two times longer than in the other cases to show the system changes.

In the experiments, the *normal* (without considering trustfulness) and the *advanced* (considering trustfulness in decision-making) model were compared. Resource unit price was considered in all cases: when no trust or reputation values were considered (normal model), agents made decisions based only on this static feature. In the other cases, a weighted sum of resource unit price and trust/reputation values were calculated and associated with a certain offer during evaluation. In the experiments, these weights were equal. As one can see from Table 11, resource unit prices are also equal for these experiments in order not to influence the difference between reliable and non-reliable agents. This way, when applying the normal model, the agents send equally priced offers. Thus, the first received offer will be the winner.

Table 11. Experiment parameters

	Experiment ID							
Parameters	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
a_r	4	4	4	4	4	4..40	4	4
a_n	6	6	6	6	6	6..60	6	6
d	3	1..10	3	3	3	3	3	3
f	0..40	20	20	20	20	20	20	20
tr	1	1	0/1	0/1	0/1	0/1	1	1
rep	1	1	0/1	0/1	0/1	0/1	1	1
θ_{tr}	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
θ_{rep}	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
t_{order}	2.5	2.5	2.5	1..18	2.5	2.5	2.5	2.5
sim. time	750	750	750	750	750	750	750	1500

Since there are several stochastic parameters, fifty simulation runs were executed for each parameter set, and the average of the results is visualized in the diagrams. In addition, the confidence intervals on 95% confidence level are also mentioned (except Experiment (7) and (8), where a single simulation run is depicted in the figures). The experiment results are presented mainly using two measures:

- *Average job lateness,* which means the sum of differences between the job due date and the real completion time for each job which was completed by the federation during the simulation run, divided by the number of mentioned jobs. If a job is finished earlier than the due date, the difference is negative – it decreases the average.
- *Average resource utilization*, which is computed by averaging the resource utilizations for all the resource types a specific agent has, in each model time unit.

Average job lateness indicates the performance of the federation as seen from outside – this value is important for outsider companies who are sending orders to the federation. Average resource utilization is important for the companies inside the federation: they are trying to maximize the utilization of their resources. Traditional metrics – for example, throughput or WIP – are not used here because these two depend on the frequency and size of the incoming orders and are not characteristic of the performance of the federation. The aim of the federation is not to increase the throughput but to increase the service level: to complete as many received requests as possible on time. If the federation performs better, it won't be able to complete much more jobs because the order stream is fixed in the simulation model.

7.1. Experiment (1) – Agent flexibility

Agent flexibility is the parameter determining the number of additional resources that a company can use in case of reorganizing its production, given by the percentage of the original resource amount. For example, a company with 20% flexibility means it can offer 120% of its original resources ($r_{max,n}$) in the processing interval of a specific job, after being asked to reorganize its production. In Figure 21, one can see the percentage of jobs that were:

- insourced (the agent had the specific resource and carried out the job by itself),
- completed after sending out (reorganizing or dividing was not necessary),
- completed after reorganizing,
- completed after dividing, or
- marked as failed.

In these experiments, companies divided the requests into 3 parts (if there were no offers after reorganizing). In the case of divided jobs, when visualizing results, the original job amount was considered: if an agent divided the job into three parts, but only one of them was completed, 1/3 was added to the "completed after dividing" category. One can see in Figure 21 that the rate of jobs completed after reorganizing increases in line with the company's flexibility, and the rate of jobs completed after dividing decreases. The ratio of failed jobs remains almost the same, which means that in the described system if a company cannot find an offer after asking the other companies to reorganize their production, it will find an appropriate offer after dividing the job. The question may be asked: why does a company try to ask the others to reorganize if dividing always solves the problem? Why doesn't it divide the job immediately? The answer is that if a company assigns a job to one partner, it can choose the one with the best parameters (unit price, trust, reputation), and does not have to compromise with agents having weaker features. Besides, dividing jobs causes additional costs in reality (for example, transportation). In the following series of experiments – as one can see in Table 11 – 20% flexibility was considered at all companies. As mentioned, fifty simulation runs were performed for each parameter set – the confidence intervals on 95% confidence level are between 0.14 and 0.17 percent of all received jobs in case of each job completion type.

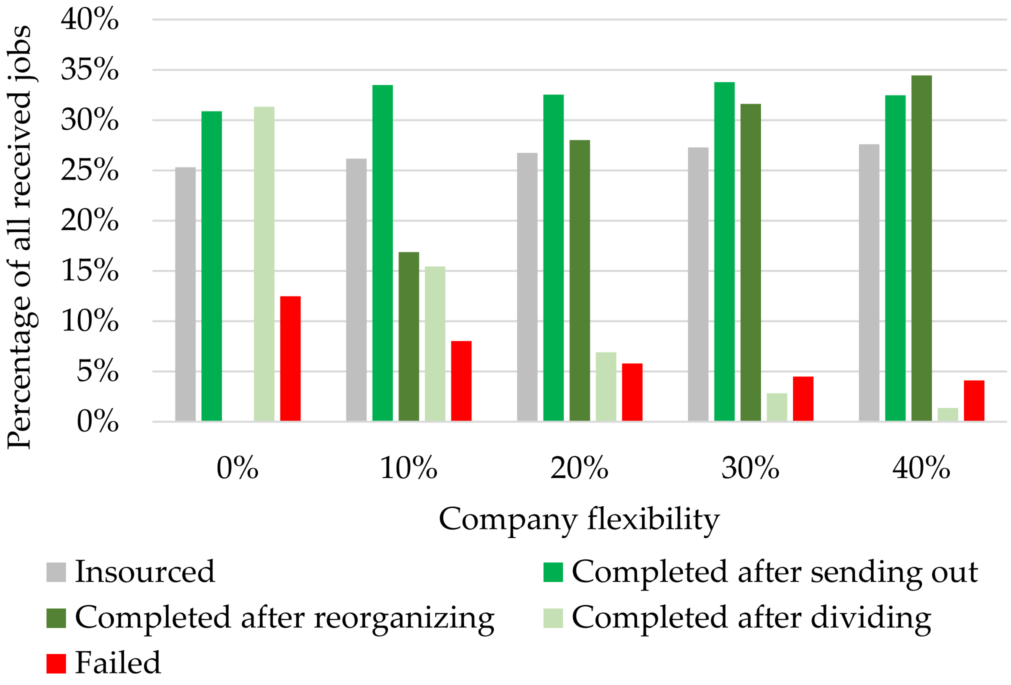

Figure 21. Effect of company flexibility on types of job completion

7.2. Experiment (2) – Dividing jobs

Some experiments were performed to determine the appropriate number of parts the agents can divide the requests into. As one can see in Figure 22, no dividing leads to a higher rate of failed jobs (15%). In the following experiments, dividing jobs into 3 parts was set because after this value, the ratio of failed jobs remains the same (6%). Here, the confidence intervals on 95% confidence level are between 0.13 and 0.18 percent of all received jobs in case of each job completion type.

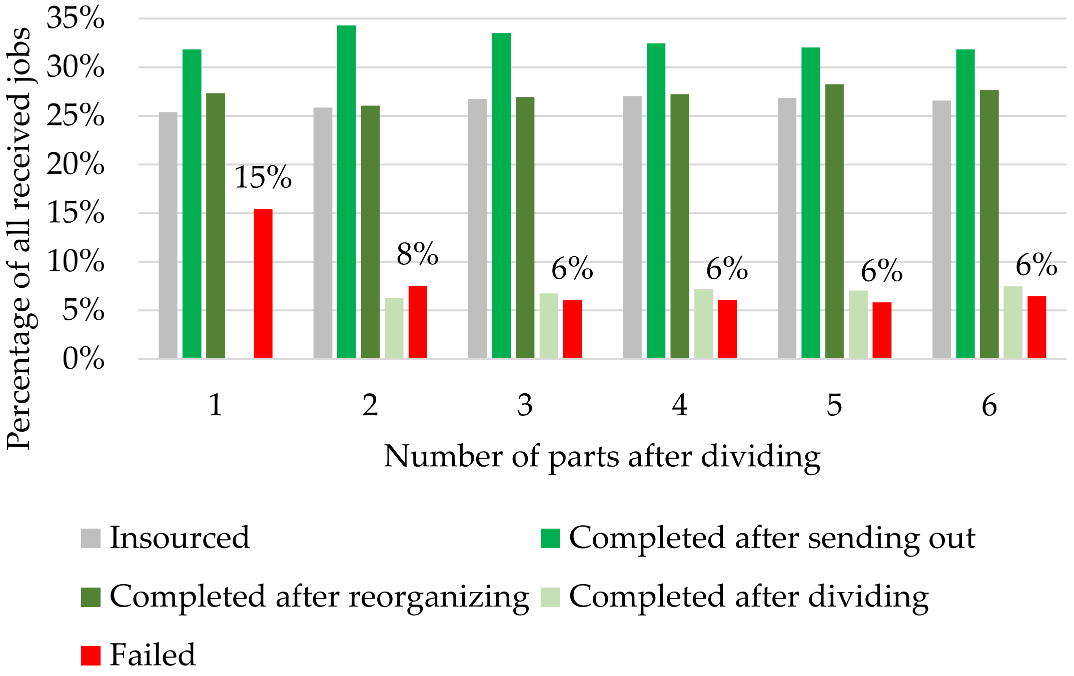

Figure 22. Effect of dividing jobs on types of job completion

7.3. Experiment (3) – Effect of considering trustfulness in decision-making

In this experiment, the effect of considering trust and reputation was investigated. Four different cases were simulated: agents could use both trust and reputation values, one of them, or none of them, to choose the best offer. According to the results, when making decisions between offers, it is worth taking at least one of them into consideration, because the average delivery inaccuracy is lower in these cases (Figure 23). According to the experiments, in the models in which agents take reputation values into account, the federation performs better than in the other cases where only trust values or none of them were considered. There is only a little difference between considering only reputation and considering both trust and reputation (the difference is smaller than the deviation of the results). The reason for this is that reputation more accurately determines the reliability of an agent because it is calculated on the basis of a higher number of jobs, as it is formed by interaction with all the other agents. The confidence intervals on 95% confidence level are between 0.03 and 0.04 model time units in the case of each model type.

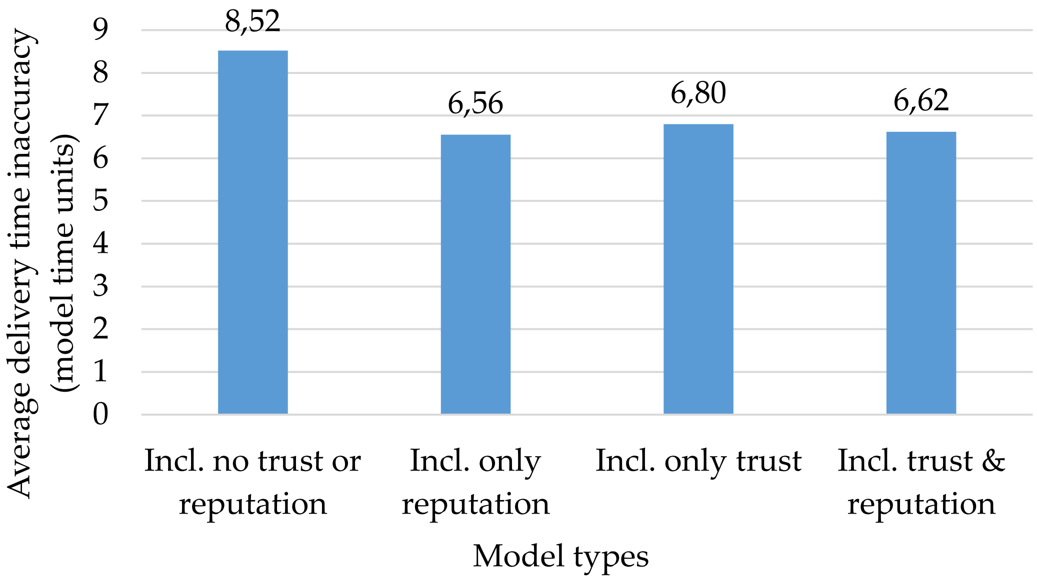

Figure 23. Effect of considering trust and/or reputation on average job lateness

7.4. Experiment (4) – Effect of incoming order frequency

In this series of experiments, the effect of the change in the order interarrival time – which can be interpreted as the load of the federation – was investigated. The question was how the overall system performance changes if the federation members receive orders (with the same processing interval and amount of required resources) more often; therefore, the load of the companies increases. Order interarrival time is a constant parameter in a specific simulation run, meaning that each company agent receives an order at the same time, and this is repeated periodically.

In Figure 24, this parameter was changed between 1 and 18, and the simulation was run applying the normal and the advanced model, as well. The average job lateness is visualized in Figure 24 in each case for the normal (including no trust or reputation) and the advanced (including trust and reputation) model, and the difference between them is also shown. The confidence intervals on 95% confidence level are between 0.8 and 2 percent of the average job lateness value in each case.

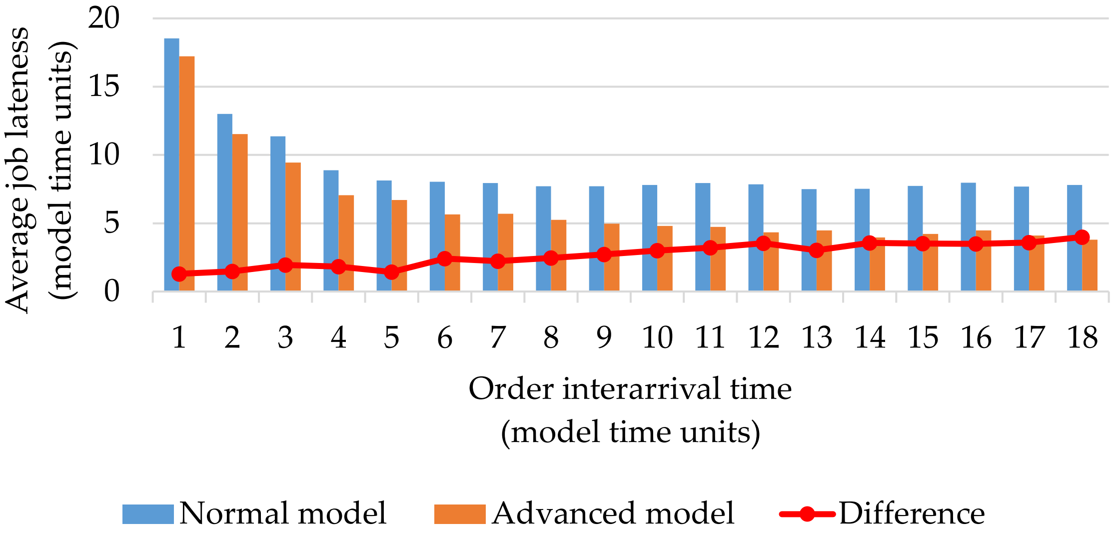

Figure 24. Effect of changing the incoming order time period on average job lateness

As one can see in Figure 24, as the arrival time between two incoming orders increases (in other words, the load of the federation member decreases), the average job lateness decreases as well in the case of the advanced model. This is because in a federation where the agents consider trust and reputation in decision making, the agents who are working faster and more reputable win more jobs; thus, the average job lateness is lower. In the case of the normal model, when the order interarrival time reaches around 5 model time units, average job lateness decreases slower than before. This is because here is the point where the resources of the agents become fully utilized, and if the agents receive orders less frequently, they will always have enough resources to finish jobs with lateness max. 4 model time units. In the normal case, all the agents have the same average utilization rate for their resources because here, the decision making is made based on the response time that is uniformly distributed. Lateness does not reach 0, because even reliable agents' real job completion times are generated from a normal distribution and thus include some variation.

In the case of the advanced model, the level mentioned above is around 6 model time units: after this, if the load of the federation decreases, the average job lateness decreases much slower than before. If the average resource utilization is investigated in the case of the advanced model, this is not the point where the resources of the reliable agents get fully utilized, but the point from which the difference is gradually decreasing between the resource utilization of reliable and non-reliable agents; more and more jobs are performed by the non-reliable agents if the load of the federation is increased.

7.5. Experiment (5) – Differences in resource utilization

In this experiment, the two model types are compared based on the average resource utilization of the agents. In Figure 25, the non-reliable C01-C06 companies and the reliable C07-C10 companies are participating in the federation. As shown in Table **11**, all the other parameters were the same in the two scenarios. In the case of the normal model, all of them have resources utilized between 40% and 50% (since the decision making is based on response time), while in the advanced case, the higher utilization of reliable companies' resources is clearly visible. The confidence intervals on 95% confidence level are between 1 and 2.5 percents of the average resource utilization values in each case.

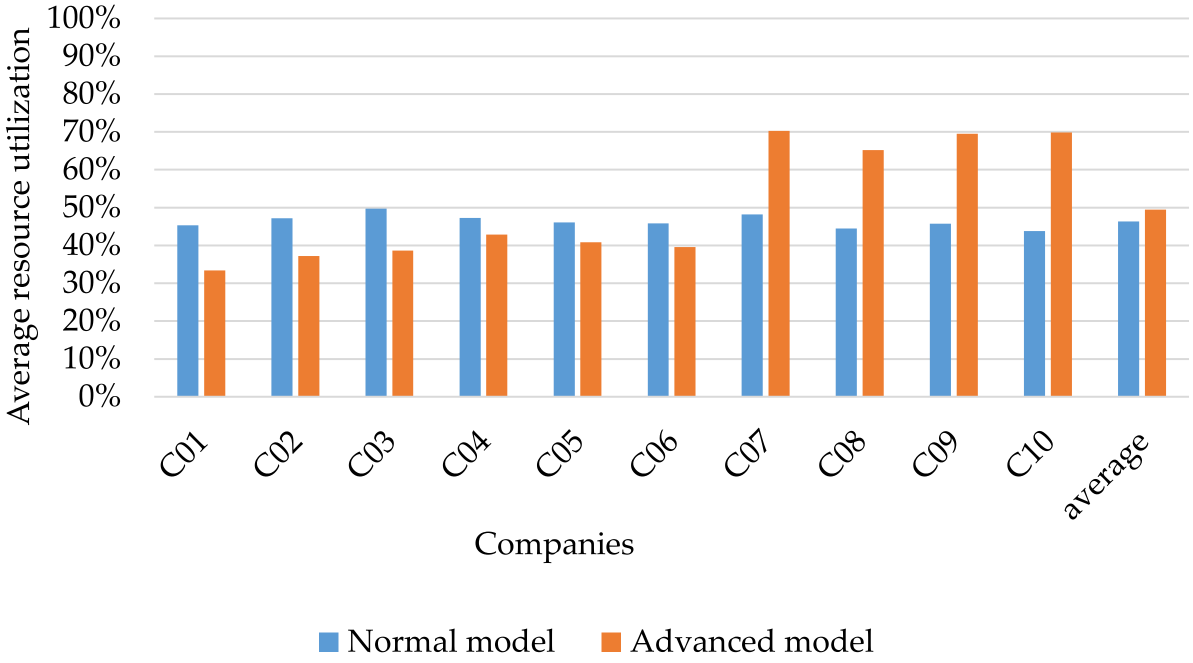

Figure 25. Average resource utilization in the normal and advanced model

7.6. Experiment (6) – Federation size and trustfulness

In Experiment (6), the federation size was increased from 10 to 100 companies, and the average job lateness was compared in the case of the normal and the advanced model. Figure 26 shows the results: as the federation grows, the difference between the two models gets larger: trust and reputation have more effect on the average job lateness. The confidence intervals on 95% confidence level are between 1 and 2 percent of the average job lateness values in each case.

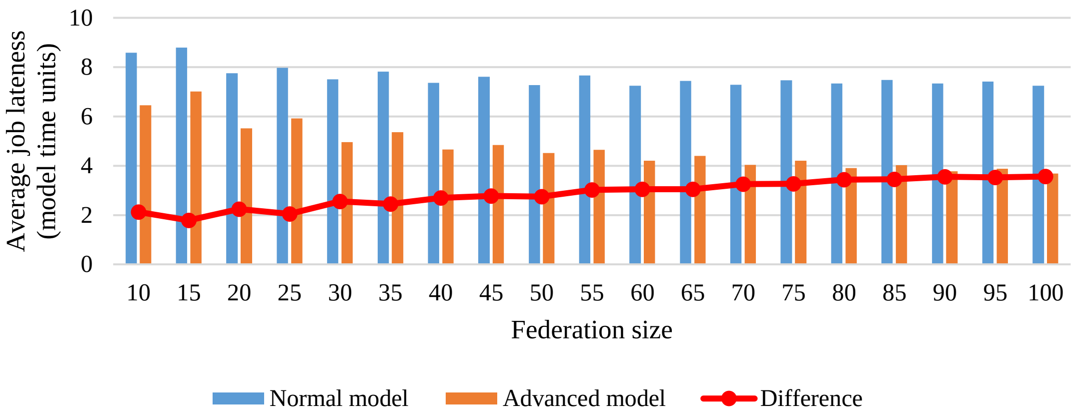

Figure 26. Effect of considering trustfulness in different federation sizes

A little fluctuation can be noticed in Figure 26 between federation sizes which are divisible by 10, and the other values. This is because in this experiment when increasing the number of members in the federation, the 10 company agents that were introduced in Table 8 were duplicated – for example, in the case of 50 agents, 5 agents had the same parameters as C01. In the other cases, when the number of members is not divisible by 10, non-reliable agents similar to C01-C05 were added to the federation, this way increasing the rate of the non-reliable agents and increasing the average job lateness, too.

7.7. Experiment (7) – Change of trustfulness in time

Here the change of reputation is investigated during the simulation run, applying 10 company agents included in Table 8. As mentioned, C01-C06 are non-reliable ones, C07-C10 are the companies that are reliable. The difference between them is clearly visible in Figure 27 – the non-reliable agents are marked with bright colours and the reliable ones with darker colours. The reason for comparing companies is that reputation more accurately determines the reliability of an agent since it is calculated based on a higher number of jobs, as it is formed by interaction with all the other agents.

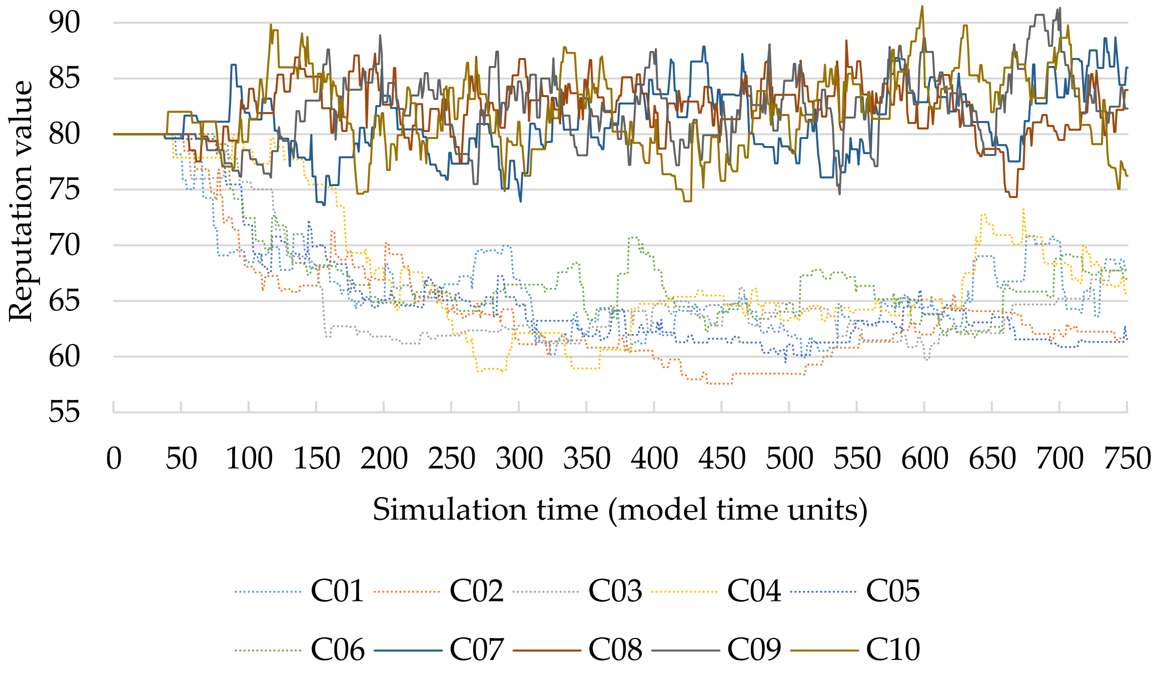

Figure 27. Change of reputation values during the simulation run

In Figure 27, at the beginning of the simulation, all agents are on the same reputation level (80) until the first jobs are not finished. After then, the two groups are diverging from each other: the reliable agents' reputation values are changing between approximately 75 and 90, and the non-reliable ones are between 55 and 70. The values are fluctuating due to the stochasticity of job lateness, but the boundary between them is clearly visible. This experiment has the same parameters as the one visualized in Figure 25: as one can see, the difference between the reputation values influences resource utilization, as well. Reliable agents with higher reputation can win more jobs, resulting in a higher resource utilization.

7.8. Experiment (8) – An unexpected negative event

In this experiment, the effect of a sudden change in trust and reputation values was investigated. In reality, due to some political or economic news or other unexpected external events, the community's opinion could suddenly change in a negative manner about a certain company. The effect of such an event is simulated by decreasing the reputation value and all the other company's trust values about the reliable C09 to 10, at model time 300. The results can be seen in Figure 28, where the reputation of all agents and the resource utilization of C09 are visualized: it lasts around 100 model time units for C09 to reach approximately the same reputation level as it reached before. This recovery time can be influenced by θ decay factor – according to Eq. (4). This way, a trade-off between appreciating the positive long-term performance and penalizing the temporary bad performance can be set in the model. As one can see in Figure 28, the resource utilization of C09 is not increasing along with the reputation because this unexpected negative event also affects the trust values. The recovery time in the case of trust values is influenced by θ decay factor according to Eq. (4) the similar way as reputation values were affected.

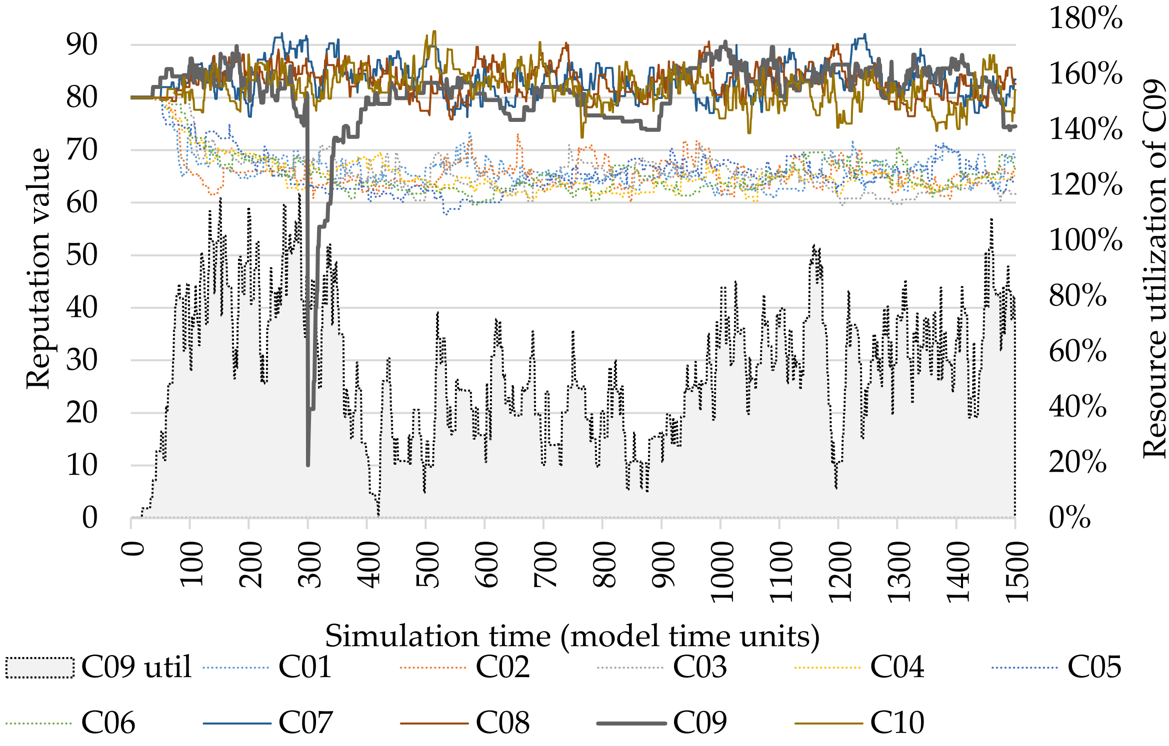

Figure 28. Effect of an unexpected event on reputation values and resource utilization

Figure 29 shows all the other agents' trust values in connection with C09 during the simulation run: as it can be seen, subjective trust values are increasing much slower than the public reputation after the event with a negative effect. Reputation is influenced by all the finished jobs. Thus, all the interactions finished by C09 can increase this value – therefore, it increases faster according to the agent's performance. In contrast, trust values can increase based on fewer jobs, only if there was an interaction between C09 and the specific agent. This way, when the agents are evaluating offers sent by C09, they calculate with the high reputation and the much slower increasing trust values. That is why resource utilization of C09 reaches the original level after hundreds of model time units after the event with a negative effect. In this case, re-running the simulation naturally leads to different diagrams due to the stochasticity of some model parameters – but the trends were the same as presented in each experiment.

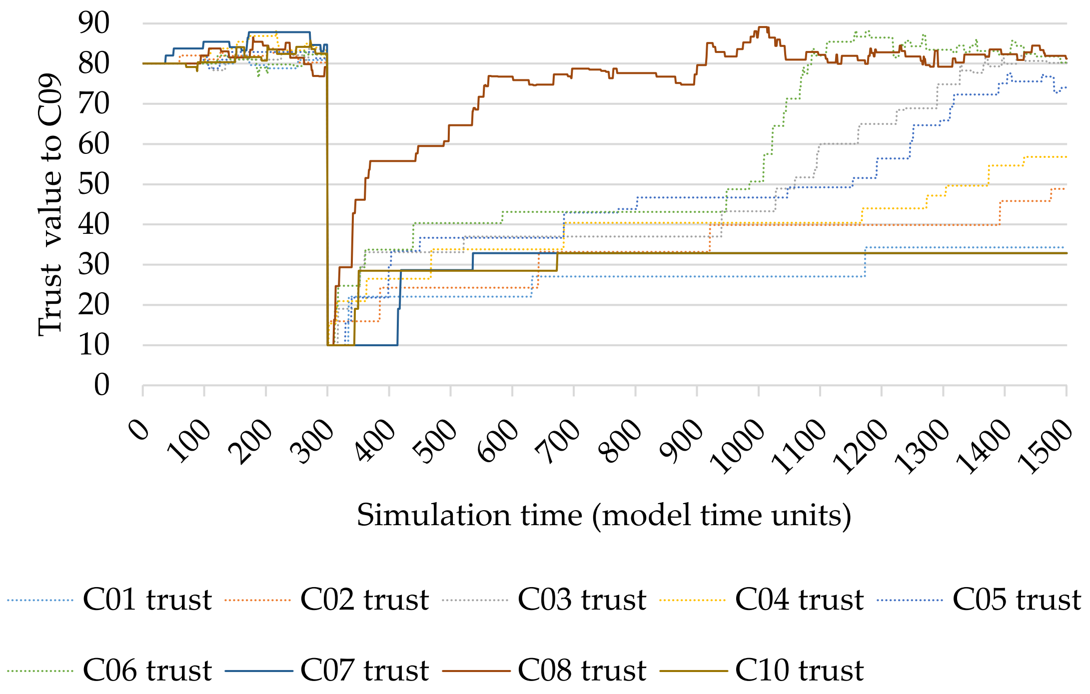

Figure 29. Effect of an unexpected event to trust values to a company agent (C09)

7.9. Effect of lead time prediction accuracy

Manufacturing *lead time* is one of the most important KPIs for companies that are striving to meet deadlines. In addition, accurate lead time prediction is the key to successful production planning and control [53]. One can find extensive literature in connection with lead time calculation and prediction; however, not in terms of resource sharing. Applying more complex tools for lead time prediction that support quasi-real-time decision making, such as machine learning or data analytics, combined with simulation models, has only started in recent years. These tools can be used to cope with fluctuating reject rates, unexpected tasks and events [107].

Determining lead times accurately has a strong effect on keeping job deadlines, which is an essential pillar in collaboration. Collaborative resource sharing only works efficiently if companies can count on their partners' promises, such as finishing an undertaken work in time. In this case, the effect of lead time prediction accuracy is tested using agent-based simulation in the *platform-based resource sharing mechanism.* It was investigated how the performance of collaborating partners changes if they could predict lead times of their jobs with different accuracy. Based on [122], resource utilization has a strong effect on lead times, which also depends on variability. Higher resource utilization level causes longer and less predictable lead times, as working on different jobs in parallel increases the complexity of production planning. In order to investigate this, experiments are performed to examine the effect of decreasing prediction accuracy when operating under a higher load.

It is important to highlight that here the focus is not on different lead time prediction methods. The aim is to show the difference between cases 1) when lead time is more accurately determined and partners could rely on each other to a greater extent, and 2) when lead time prediction is not accurate, and failures in keeping deadlines may require changing existing production plans. The novelty of the results presented here is the consideration of lead time prediction accuracy in collaborative resource sharing, which is unique in the literature.

The accuracy of lead time prediction could influence the production plans of the participants, as described below. One can distinguish between two types of jobs that are performed by a company:

- customer orders coming from outside the federation are completed using its own resources, and
- an offer sent to the platform is accepted by another company.

In both cases, the company estimates the lead time of the specific job and inserts it into its production plan. If the real finish of a job is delayed compared to the estimated date, it can overlap with: (1) already offered resources that were sent to the platform earlier: these must be withdrawn to finish the job, or (2) already undertaken jobs that could only be started later and may be finished with additional delay (causing reputation loss, also). If a job is finished earlier than planned, it may occur storage capacity shortages, as the products are waiting for delivery.

Effect of lead time prediction accuracy on service level and resource utilization

As mentioned, real processing intervals are determined by normal distributions in the model to consider lead time prediction inaccuracy. In the first experiment, the effects of changing the deviation of the distribution (from 0% to 50% of the expected value) and the Δ value (from 0% to 50 % of the expected value) were investigated. These two parameters determine the lead time prediction accuracy of a specific company. To compare different scenarios, service level was applied as a KPI. It is calculated by measuring percentage lateness in the case of all completed jobs, subtracting it from 100%, and recording the average of these values after each simulation run.

In this experiment, each company had the same deviation and Δ parameters but different equipment types. Completing jobs inaccurately results in lower reputation values, which directly affects the agent's performance, as decision-making between offers is made based on reputation. The results of the experiment can be seen in Figure 30. The simulation was run for 500 model time units, but the results are visualized by neglecting the first 30 time units (ramp-up phase).

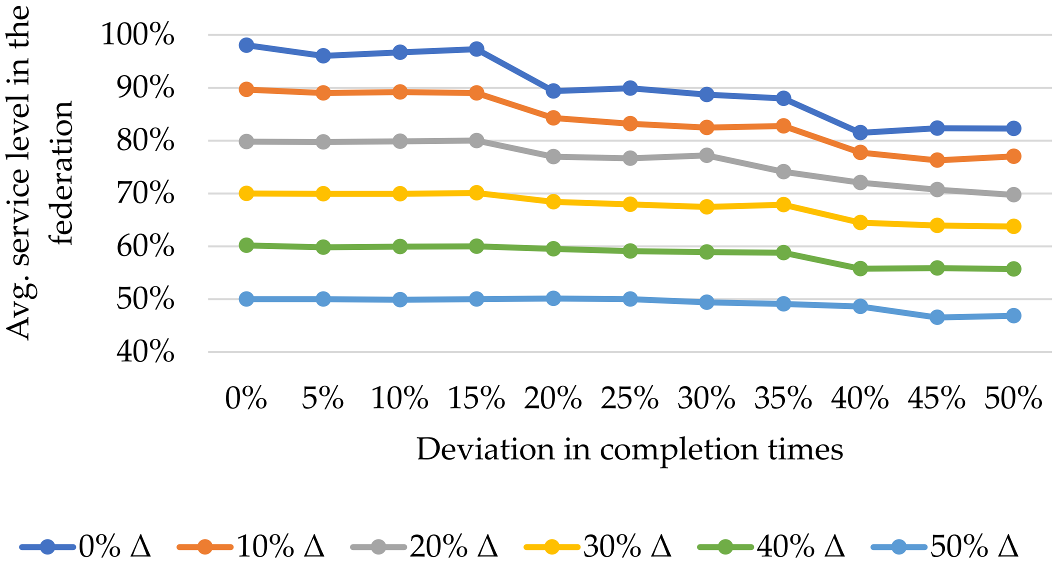

Figure 30. Effect of deviation in completion times and Δ on average service level.

Based on the results, the following remarks could be made. As one could expect, the higher the Δ value, the smaller the average service level is in the federation. When increasing the deviation in the real completion times (leaving the expected value unchanged), the service level decreases. The extent of decrease depends on Δ: smaller Δ causes a higher decrease on service level. Therefore, in the case of agents that are less likely to be late (more reliable agents, smaller Δ), the deviation of their lead time prediction has a higher effect on service level than in the case of non-reliable agents (higher Δ).

Taking resource load into consideration in lead time accuracy

As mentioned above, resource utilization is an important factor that influences the accuracy of lead time prediction. In this experiment, companies with higher resource utilization can predict manufacturing lead times less accurately, in order to simulate more realistic scenarios. To visualize the results in a more transparent way, 10 companies are forming the federation in this experiment. Out of them, 4 are reliable (Δ = 0%) and 6 are non-reliable (Δ = 30%). The deviation of real processing times is determined by dividing the actual resource utilization level of a company by 2. This means if a company starts working on a job, and in that time point its resource utilization level is, e.g., 80%, the real processing time of the specific job is determined by a truncated normal distribution with a deviation equal to 40% of the expected processing time (in case of reliable agents). Of course, the mean of the distribution is shifted with 30% of the expected processing time in the case of non-reliable agents.

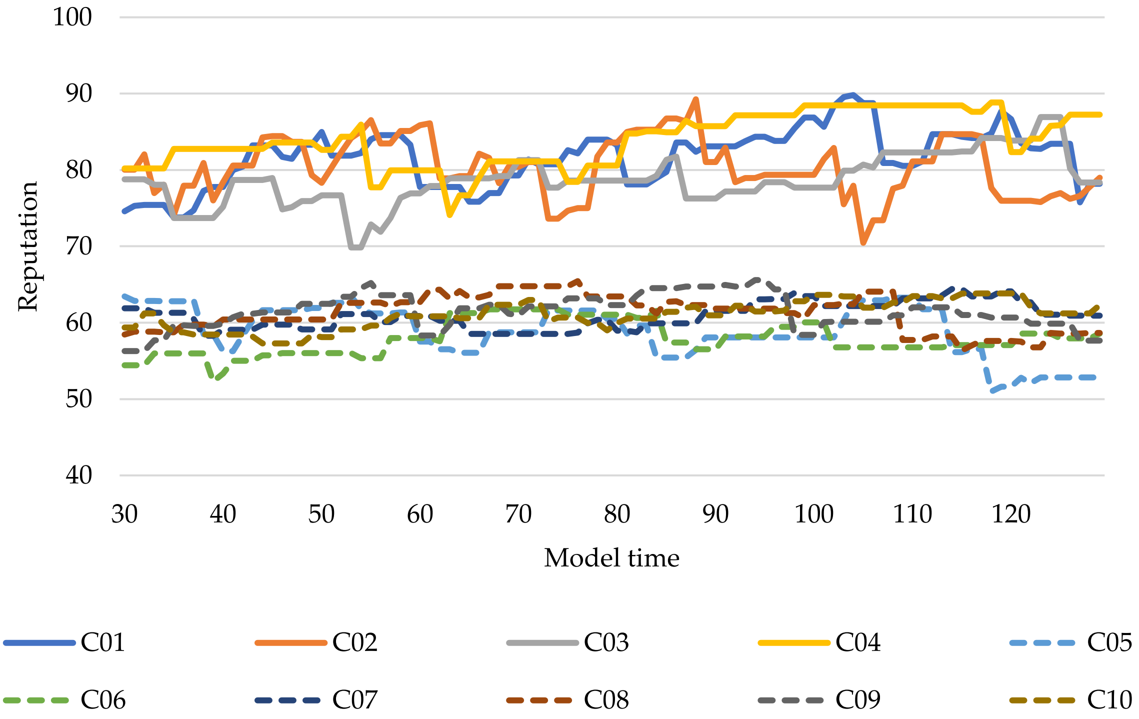

Figure 31. Change in reputation in the original scenario

In this experiment, the change in reputation values is visualized in the case of the original scenario (10% deviation in real processing times for all agents) and in the case of taking resource utilization level into consideration in the abovementioned way (Figure 32). In the figures, reliable agents are marked with solid lines, and non-reliable ones are shown with dashed lines. The results are visualized between model time units 30 and 130 (100 model time units), in order to exclude the ramp-up phase, similarly to in the previous experiment. In this case, it was unnecessary to run the simulation for a longer time, as the main findings are the same after this time interval. Based on Figure 31, one can see that neglecting the resource load of companies results in a clear separation of reliable and non-reliable companies. In general, reliable companies are between reputation levels 75 and 85, while non-reliable ones are between 55 and 65. However, for reliable companies Δ = 0%, they also have a deviation in their real processing times, this way they do not reach the 100 level.

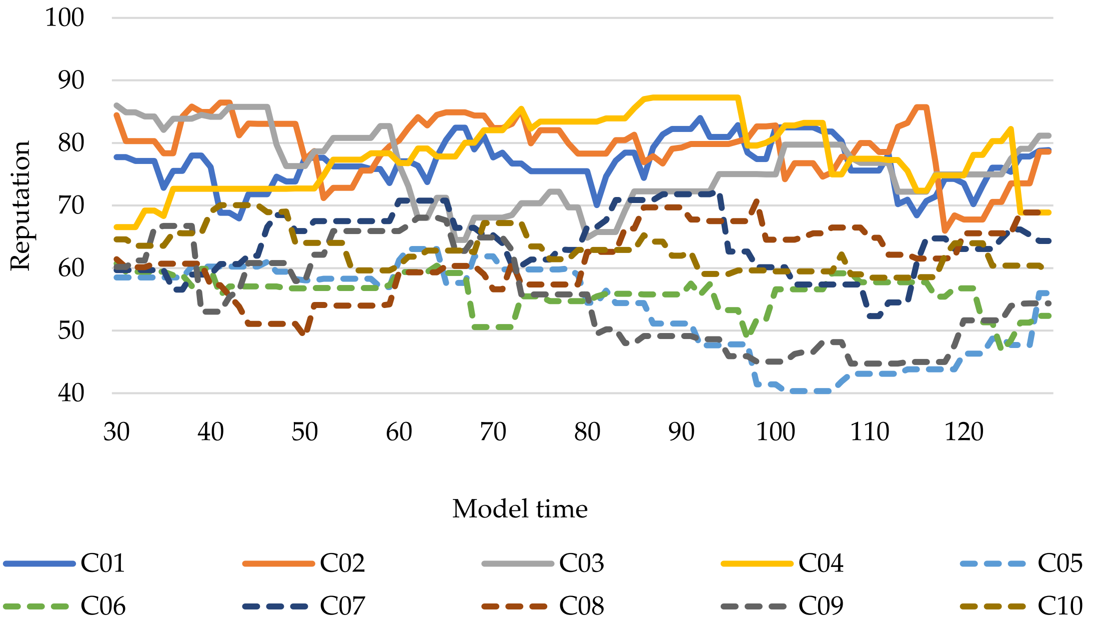

Figure 32. Change in reputation when taking resource load into consideration in lead time prediction accuracy.

In Figure 32, one can see that the difference is not that clear between reliable and non-reliable companies when taking resource load into consideration in determining lead time prediction accuracy. In general, reliable agents are reaching a higher reputation level, but non-reliable ones sometimes have similar values. Another observation is that the reputation values are fluctuating to a greater extent in the case of non-reliable agents compared to the previous case. When neglecting resource load, the values are in a zone with a width 10. In contrast, in this case, they are changing between 40 and 70. The fluctuation is a little higher for the reliable agents, also. This can be explained by the changing deviation of the processing intervals: when reaching a high reputation level, the company wins lots of jobs, which causes a high load on its production system and lead time prediction inaccuracy as well. This way, the load is balanced between participants who are members of the resource-sharing federation.

7.10. Conclusions of considering trustfulness in resource sharing

Based on the experiments, if trust and reputation are considered in decision making (advanced model in case of direct exchange-based mechanism), the system performs better than in the case when offer evaluation is based on a static parameter only (normal model). The difference between the advanced and the normal model (in other words, the impact of considering trustfulness) depends on the federation load: if the participants are highly overloaded and some of the agents are forced to work together with non-reliable partners (because they want to complete the received orders), the difference is smaller than in case of a less loaded federation, where there is the opportunity to choose a more reliable partner to work with. The results have also shown that the higher the number of federation members is, the higher the impact of trustfulness is on the federation's performance. It was also presented that considering the trust and reputation of the participants affects the utilization of their resources as well: reliable agents' resources are utilized on a higher level. The effect of an unexpected negative event has been tested, too: it is easier to build up a good (public) reputation than to recover from bad trust values in the applied model; and this causes the low utilization level of resources for a relatively long time after the negative effect, too. Results presented in this thesis suggest that including trust and reputation in a manufacturing resource sharing mechanism really makes a difference in the performance of a federation containing manufacturing companies.

Agent-based simulation experiments have also shown that deviation in lead time prediction could strongly affect the average service level in the resource sharing federation: higher deviation causes worse performance. The decrease in performance also depends on the reliability of companies: if they are more likely to finish jobs in time, increasing the deviation of lead times results in a relatively higher performance decrease. The effect of taking the resource load of companies into consideration in lead time prediction accuracy was also investigated: companies with higher load could predict lead times with lower accuracy. This way, reliable and non-reliable companies could be balanced in terms of reputation values.

8. Comparison of platform- and direct exchange-based resource sharing

Based on the examples from the literature mentioned in Chapter 2, it seems that resource sharing generally improves the participant's resource utilization level. In this chapter, two different ways on how it is more valuable to share resources are investigated and compared: 1) by using a direct exchange-based mechanism(Chapter 3) or 2) by joining the intermediate platform (Chapter 4) that is responsible for matching requests with the offers. By modelling the same set of companies loaded with an order stream having the same parameters, a fair comparison is made that is unique in the literature. The performance of the mechanisms is assessed considering three different viewpoints: a) average resource utilization, b) average service level, and c) communication load. Both the compared approaches consider trust-related aspects as well (based on the trust model introduced in Chapter 5), which is usually neglected in resource sharing case studies or investigated without taking capacity constraints into account.

8.1. The two models

As mentioned in Chapter 3, in the case of the direct exchange-based model (referred to as "Model A" in short in this chapter) – participants communicate and coordinate their actions directly with each other: they are sending resource requests to all the other organizations who are part of the federation. When receiving a request, all the participants being able to complete the request are applying for the given job and send offers to the requester.

In the case of the second model introduced in Chapter 4 ("Model B"), the participants are sending their resource offers and requests to a central platform, which does the matching: it sends the appropriate offers to the requester company from the already received ones. Since the platform is aware of all requests and offers, it can combine several offers to complete one request.

In Model A, as companies are communicating directly with each other and thus know the identity of the offerors, they can calculate with trust and reputation. In Model B, the identity of the winners becomes known only after the decision was made, to prevent companies from obtaining information about each other's free resources. Thus, subjective trust values cannot be considered in decision-making, only reputation, which is calculated in the same way as in Model A.

In the comparison, both trust and reputation are cumulated values calculated based on ratings given by the requester company about each interaction, as discussed in

Chapter 5. For calculating and updating reputation values and ensuring their public availability, a central unit is necessary in both cases; nevertheless, its main role is different in the two models. In Model A, it manages the entries and exits from the federation and updates the public reputation rankings. In Model B, besides these activities, the companies send all the resource offers and requests to the platform, which dynamically matches them and manages the contracting, as well.

When having resource shortages, companies send requests to all the other federation members (Model A) or to the platform (Model B). A request contains all the resource requirements mentioned in the case of a job, and in Model B, the maximum number of fragments the request can be divided into. In Model A, when a company receives a request, it checks its already planned works for the future. If the appropriate resources (same type, sufficient quantity in the required interval) are available, it sends an offer to the requester, which can choose the best based on its preferences.

In Model B, companies send offers to the platform regularly. Receiving a request, the platform checks whether there is a match between the new request and the active offers. If the platform receives an offer, it checks whether some of the non-fulfilled requests can be completed with the new offer (or by combining the new offer with the earlier ones). If the platform finds a match, it notifies the requester about all the possible offer combinations fulfilling the request.

In Model A, the companies are exchanging resource information about themselves directly with each other in case of each request or offer. In Model B, information is shared only with the platform, and, in addition, the requester company knows only the identity of the winner(s) after choosing from the offers. In the first case, companies have to trust everyone else in the federation; while in the second case, it is enough to trust the platform.

In Model A, a company tries to divide the offer into a feasible number of equal-sized fragments in case of receiving no offers for the whole request, as it does not know the free capacities of the others and cannot adjust the request sizes accordingly. In Model B, the requester can set the maximum number of offers that can be combined to fulfil the request, and the platform sends all the possible offer combinations accordingly.

Other aspects are the computational and communication load of matching. In Model A, requests are sent to all the companies in the federation, and only the companies that have the proper resources to fulfil the request, send an offer. In the case of fragmented requests, each fragment is treated separately: companies must check their available resources for each of them and send separate offers to the requester. In contrast, in

Model B, requests and offers are sent only to the platform, which performs the matching. Requests do not have to be divided into pre-defined fragments, because the matching function can combine the received offers based on the maximum number of fragments. Faster reaction time can be achieved since a company does not have to wait for several answers, it is enough to get feedback from the platform. However, due to offer anonymity, companies cannot calculate with trust values (reputation values are available in both cases).

Naturally, Model B is more vulnerable in the case a problem occurs with the platform. In such a case, the whole resource sharing process fails, and the companies could try to contact each other directly. In Model A, only the reputation values are lost, and subjective trust values are still available. Loss of access to the updated member list of the federation affects both cases. A substantial difference between the two approaches is that only Model B provides the possibility to optimize on a global level, as the platform is aware of all requests, offers and contracts created in the federation.

The differences between the two approaches are summarized in Table 12.

Table 12. Comparison of the direct exchange- and platform-based approach

Comparison aspect	**Model A**	**Model B**
Role of the platform	Managing entries and exits, updating reputation values	Same as in Model A + request-offer matching, managing contracts
Communication, information sharing	Directly with each other	Only with the platform
Anonymity	Companies know the offer parameters and the identity of the offerors	Companies know the offer parameters without sender identity. The identity of the winner(s) turns out after choosing from offers
Dividing requests into fragments, if one offer cannot fulfil it	Requesters outsource all the fragments separately	The platform divides the offers based on the parameters of the whole request
Computational load	Requesters receive only the suitable offers	The platform checks all the received offers and requests
Decision-making	Reputation and trust are considered	Reputation is considered, trust is not
Vulnerability in case of problems with the platform	Reputation values are lost, but the resource sharing process can operate	Matching function of the platform fails, companies can share resources directly (same as in Model A)
Optimizing on a global level	Not possible	Possible

8.2. Simulation experiments – comparison of the two models

To compare the two models, experiments with agent-based simulation were performed in AnyLogic: the simulation model was created the same way as described in Chapter 6.

Regarding the input, in both models, the same companies with the same resources were loaded with the same average order size and arrival rate. The most important parameters of the experiments are presented in Table 13. For the parameters determined by using a truncated normal distribution, the mean and sigma values are included in the table (for the constant ones, sigma is 0). In these cases, the difference between the lower and upper bounds of the distribution from its mean is sigma/2. The parameters relevant in only one of the two models are marked with a superscript.

In the experiments, out of 10 companies, 6 are *non-reliable,* and 4 are *reliable*. The job completion times are also normally distributed: for non-reliable companies, the mean of the distribution is shifted, creating a lower chance for them to finish the undertaken jobs in time. In the experiments, 20 different resource types were initiated: one company had 10 to 20 types of them with the amount 8 to 12.

As indicated in Table 13, each company received an order every 1.5 days, and required 8 resource units for 20 time units from a specific resource type. To reduce the administrative costs of contracting and the computational load of the platform, matching is only possible between exactly one request and a maximum of three offers in Model B. Similarly, requests can be divided by requester companies into three parts in Model A. In Model B, companies sent offers to the platform about their free resources with a look-ahead for the next 40 time units. In all cases, the experiments were run for 500 time units: based on observations, the investigated KPIs do not change after this time in an unexpected way. Since some of the parameters are normally distributed, 10 experiments were performed for each parameter set, and the average of the values is presented in the following paragraphs. The three examined KPIs were *average resource utilization, average service level* of federation members and *communication load* (defined below in detail). In both cases, 48×10 experiments were performed while increasing the number of federation members from 3 to 50 companies, with the aim of investigating the effect of federation size on the KPIs.

Table 13. Input parameters for experiments.

Parameter	Mean	Sigma	Unit
Initial reputation^{A,B} and trust^{A}	80	0	-
Incoming order arrival rate	1.5	0	1/mtu*
Incoming order length	20	5	mtu
Incoming order resource quantity	8	2.7	-
Max. number of offer fragments	3	0	-
Planning horizon^{B}	40	8	mtu
Simulation time	500	0	mtu

* model time units

8.2.1. Average resource utilization

To compare the federation performances in the two models, resource utilization of the federation members was sampled in every time unit. The average of these values after each simulation run was calculated by neglecting the first 30 values out of 500 (run-up phase). The confidence intervals on 95% confidence level in the case of the remaining 470 values obtained from a specific simulation run were between 2~3% (4~5% in federation sizes under 10). As one can see in Figure 33, in both models, average resource utilization increases until the federation size reaches 10. The difference between the two models is approx. 30% and slowly increasing with the number of federation members. This is mainly due to the effect of the platform, which finds matching offers for requests in a more efficient way and divides requests into fragments adjusted to already received offers. In contrast, in Model A, companies try to outsource equal-sized fragments, which limits the solution space.

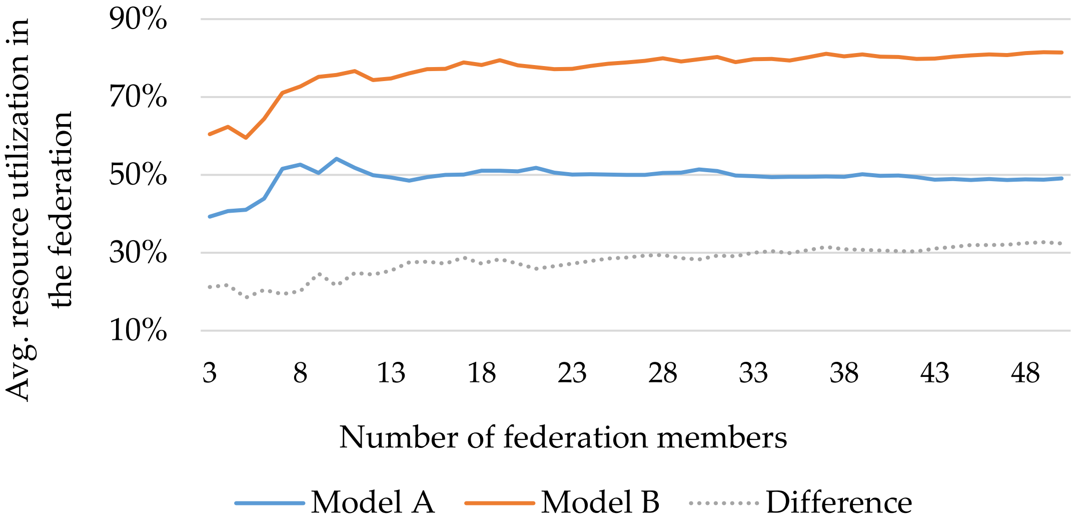

Figure 33. Difference between average resource utilization in the two models.

8.2.2. Average service level

The performance of the resource sharing approaches could be measured by comparing their ability to find more reliable partners to work with. Outsourcing jobs to more reliable partners results in less delay in job completion times: average service level is used to highlight this setting. The results show that Model A performs better: after reaching federation size 10, the difference between the two approaches is approx. 4~5% (see Figure 34). The confidence intervals on 95% confidence level are between 0.1%~1%. Since the average resource utilization is much lower in Model A, the reliable partners are not as much loaded as in Model B. Thereby, the companies have a better chance of choosing a more reliable partner in the first case. This means a kind of trade-off between resource utilization and service level: if the goal is to maximize resource utilization, offers from non-reliable partners have to be accepted as well; but if the aim is to reach a high service level, a limit could be defined for trust/reputation values of possible partners. This setting is also included in the models, but its effect is out of scope for this dissertation. The cause of the non-monotonous trend in the results is that increasing the number of federation members was done by creating companies with the same parameters as the first 10 in the same order. For example, in the case of 16 federation members, 12 were non-reliable and only 4 reliable.

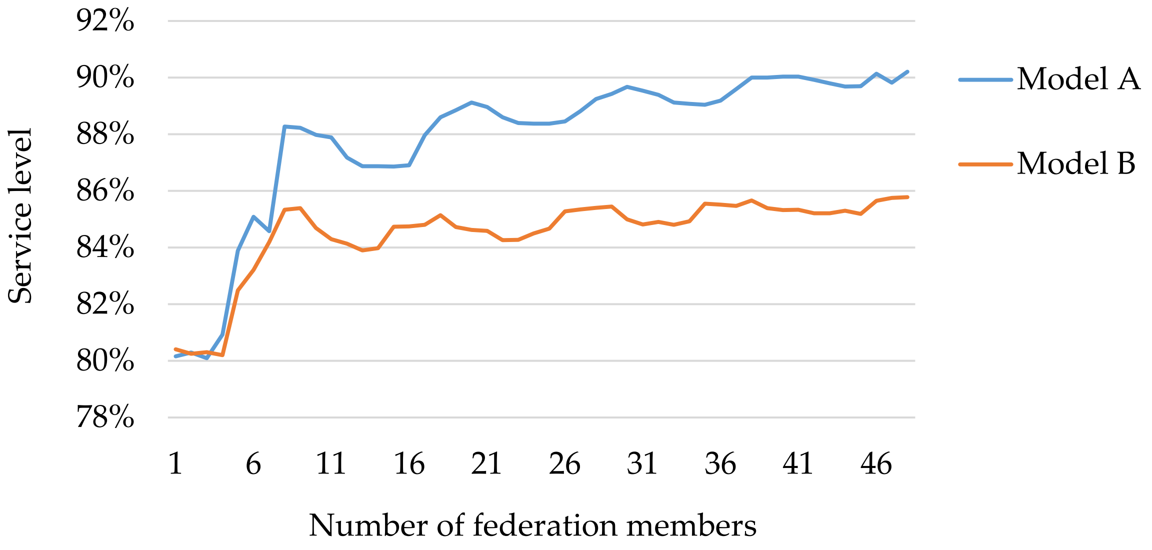

Figure 34. Difference between service levels in the two models.

8.2.3. Communication load

In the third experiment, the number of messages was investigated. As mentioned, the two models use different communication mechanisms. Message types in Model A are the following:

- *original/divided requests* (sent to all federation members),
- *offers* (sent in response to a request),
- *notifying winner(s) and loser(s)* in response to offers,
- *contracts* between the requester and offeror directly,
- *contract cancelling* messages, and
- *ratings* about contracts.

In the case of Model B, only *original requests* and *offers* are sent to the platform. Only the winners are notified after matching – the offers (and requests) that are not matched are deleted automatically from the offer database when they expire. *Notifying winner(s)*, *contracting*, furthermore *cancelling* and *rating contracts* are all made through the platform.

Figure 35 highlights the communication load, which increases heavily in Model A, even more than linearly when the federation grows. A second-order polynomial trendline was fit to the results in the case of Model A, and a linear trendline in the case of Model B. For each experiment, the difference between the two models depends on the parameters of the incoming orders (arrival rate, resource quantity, length) and the parameters of the companies, also.

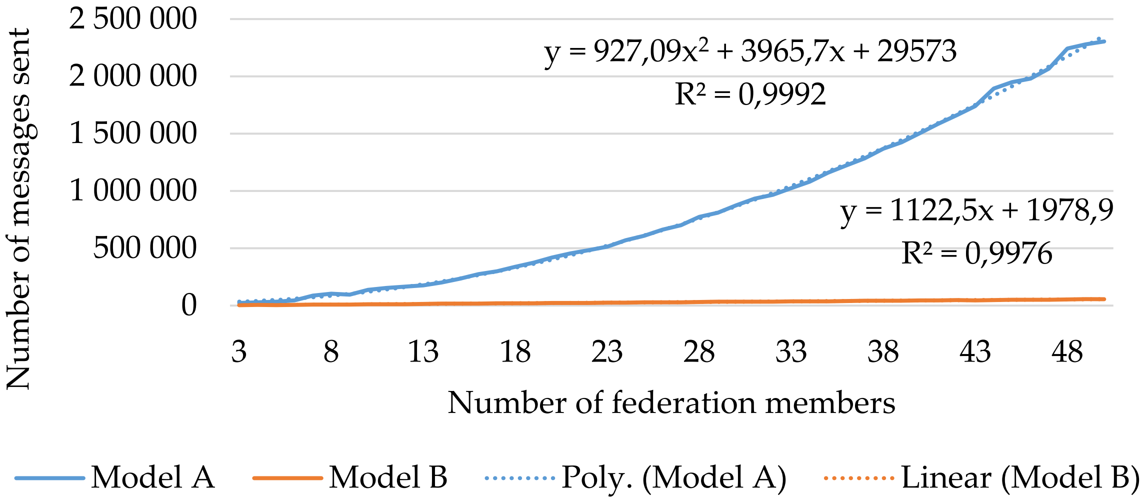

Figure 35. Difference between communication load in the two models.

8.3. Conclusions of the comparison

In this chapter, two manufacturing resource sharing approaches were compared: Model A, where resource sharing is arranged by a direct exchange-based mechanism, and Model B, where resource requests and offers are matched by an intermediate platform. Agent-based simulation experiments have shown that the difference between average resource utilization is approx. 30% in favor of Model B, due to the more complex matching logic of the platform. However, the average service level is approx. 4~5% higher in Model A, since lower resource utilization causes that reliable companies have more free capacities that can be requested by others. Communication load is one order of magnitude lower in Model B, because the companies are sending messages only to the parameter instead of each other.

It is important to mention that while a platform-based solution could improve the federation's performance from different aspects, it does not take away the possibility of autonomous decision-making from the participants. In addition, a platform can limit the decision space by selecting the appropriate offers and widen it at the same time in useful directions (for example, with combining offers).

9. Financial model for platform-based resource sharing

In this chapter, the calculation of the cost and income types applied in the platform-based model is introduced. When calculating these items, only the costs incurring and incomes generated for the resource sharing based federation members and the platform are considered, focusing on the period when a company is a member of the federation. To help understand the formulas, the meanings of the notations are summarized in Table 16 of the Appendix.

The main goal for the financial model is to verify the platform-based resource sharing method by investigating the incomes and costs of the participants. Although [28] determines the performance of the supply chain based on the profit it makes, in this thesis the aim of the financial model is to investigate the detailed financial elements of the participants, instead of evaluating the performance of the whole federation from the financial perspective.

After detailing the financial model, platform-based resource sharing and a non-resource sharing companies are compared from a financial perspective. Two financial aspects are particularly relevant: benefits due to resource sharing, e.g., higher resource utilization and additional efforts caused by transport and management costs of the platform. Existing cost structures found in the literature were adapted to the specific use cases while missing important aspects of resource sharing, thus, based on the cost structures mentioned in subsection 2.5, I created a new financial structure accommodating the relevant aspects for the comparison (see Figure 36).

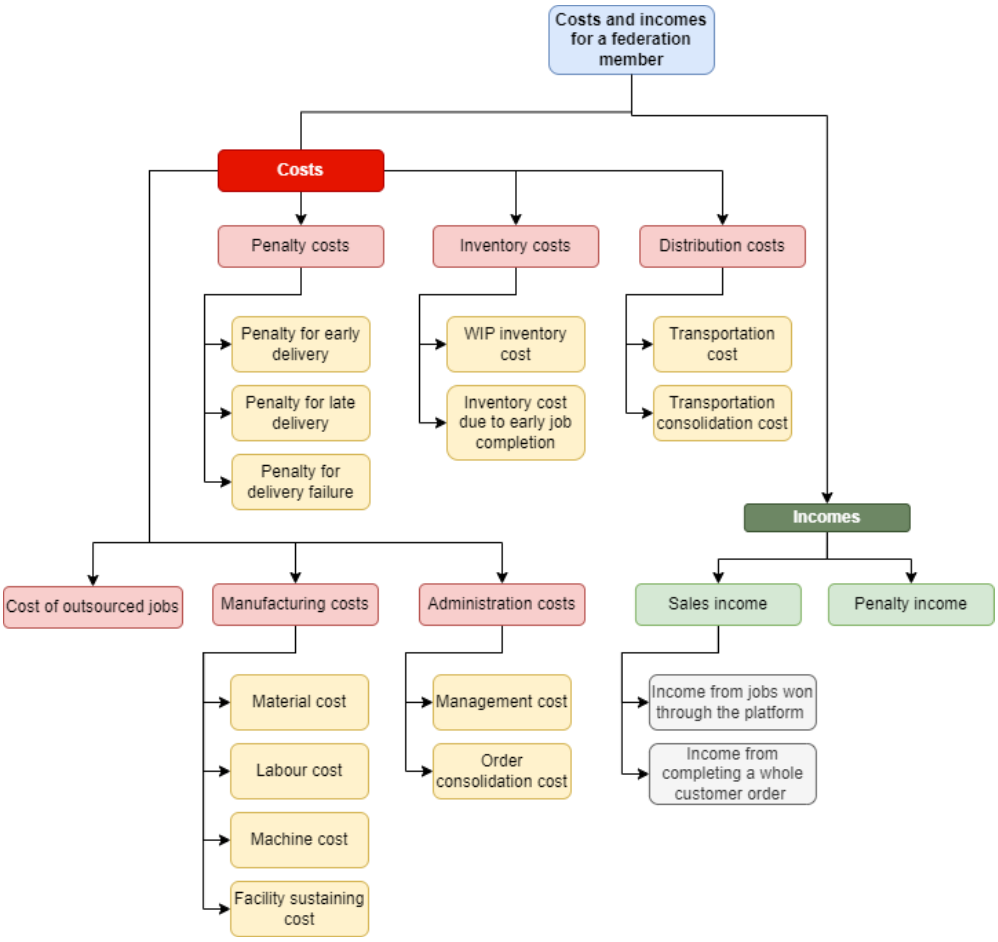

Figure 36. Financial structure for a federation member in platform-based resource sharing

For each cost and income type, a hypothesis is derived, and a corresponding function is formulated within the following subsections. The main cost types are the following:

- Manufacturing costs include all the costs related to manufacturing the requested number of products: the company must pay for the materials, the labor and the machine costs, and, in addition, facility sustaining costs are also considered.
- Administration costs consist of two parts: (1) management costs, which mean all the costs paid to the platform (entrance fee, regular participation fee, sending and accepting requests and offers), (2) order consolidation costs incurring for the lead company who is managing all the jobs in an order.
- Penalty costs are defined to compensate additional costs incurring for the lead company due to early delivery (additional inventory cost), late delivery (sales opportunity loss), and delivery failure (lost profit).

- Inventory costs can occur (1) when storing WIP products during production and (2) in case of early job completion, before delivery.
- Distribution costs cover the transportation costs (between customers and lead companies) and transportation consolidation costs (between federation members, in case of outsourced jobs).

Naturally, there are some simplifications in the model: e.g., as mentioned in the resource sharing model descriptions, companies do not try to find a suitable offer after the deadline of a customer order. In reality, this would cause backlogging costs, which is not included in the model yet.

9.1. Manufacturing cost

In [118], the authors distinguish between four cost types regarding manufacturing costs:

- unit-level activity costs (e.g., material, machine, and labour costs),
- batch-level activity costs,
- order-level activity costs, and
- facility sustaining costs.

Manufacturing costs in the presented model are introduced by taking this approach into consideration. For simplicity reasons, batch- and order-level manufacturing costs are neglected because it would raise the question of the optimal batch size that is not in focus here. For a part of a job j that was outsourced to company B or insourced by itself, the manufacturing cost $c^B_{manuf_j}$ can be calculated by adding the $c^B_{material_j} \cdot \alpha^B_j$ material costs (material cost per product multiplied with the number of products) to the $(c^B_{machine_j} + c^B_{labour}) \cdot t^B_j$ machine and labour costs per time unit multiplied with the time that is required to complete an order:

$$c^B_{manuf_j} = c^B_{material_j} \cdot \alpha^B_j + (c^B_{machine_j} + c^B_{labour}) \cdot t^B_j \qquad \text{Eq. (6)}$$

For company B, the total manufacturing cost for the time interval it was a member of the federation for the xth time ($C^B_{manuf_x}$) includes the c^B_{sust} facility sustaining costs multiplied by the T_x time interval for which the company is a member of the federation for the xth time; and the manufacturing costs of all the jobs that it has completed during T_x.

$$C^B_{manuf_x} = c^B_{sust} \cdot T_x + \sum_{j=1}^{J} c^B_{manuf_j} \qquad \text{Eq. (7)}$$

9.2. Inventory cost

In the presented financial model, inventory costs may occur due to three main reasons. The first of them is when a company must store the products during manufacturing (WIP inventory cost):

$$c_{WIP,j}^{B} = \frac{\alpha_j^B}{2} \cdot c_{inv_j}^{B} \cdot t_j^B \qquad \text{Eq. (8)}$$

Where $c_{WIP,j}^{B}$is the total WIP inventory cost for the part of job *j* that was outsourced to company B, $\alpha_j^B$is the number of parts that have to be stored, $c_{inv_j}^{B}$is the inventory cost per product in job *j* for company B for one time unit, and t_j^B is the manufacturing time of the specific job part. As a simplification, it is assumed that the number of manufactured products grows steadily during the manufacturing time, and in this case, the average number of stored products is half of the final number of them.

The remaining two causes for additional inventory costs are based on the early completion of a job. The first is when a resource offeror delivers the finished products to the lead company earlier than the start of the on-time interval of the specific job. In this case, inventory cost incurs for the lead company, which recovers this cost by penalizing the resource offeror with the same amount. As mentioned below in the *Penalty Cost* subsection, this cost is calculated by multiplying the number of products in the specific part of the job that was outsourced to company B by the inventory cost of agent A for one product in job *j* for one time unit and by the time unit for which the products have to be stored:

$$c_{inv_{j,B}}^{A} = \alpha_j^B \cdot c_{inv_j}^{A} \cdot \left({t_{d_j}}^{B} - \, {t_{sa_j}}^{B}\right) \qquad \text{Eq. (9)}$$

The second reason is when the lead agent finishes an *insourced* job earlier than the start of the on-time interval determined for it and must store the products until the whole order is finished. In this case, the additional inventory cost incurring due to the insourced job is calculated similarly as in Eq. (9).

9.3. Penalty Cost

In supply chain relationships, companies must be motivated to deliver accurately. For BTO companies, which are operating a Just-In-Time (JIT) production system, excessive inventory is not an option, and inaccurate delivery times can cause serious space and cost problems [49]. In such cases, delivering products too early and too late also must be penalized somehow. The most common way of forcing suppliers to be more reliable and to compensate the customers for the costs coming with early, late, or failed delivery is issuing a penalty cost. Authors of [50] also state that exact, cost-based performance measurement is a key aspect in connection with delivery times and measuring delivery performance, and it is closely connected to reliability. For example, in the automotive industry, Saturn levies fines of $500 per minute against suppliers who cause production line stoppages [43]. Chrysler fines suppliers $32,000/h when an order is late [106].

Authors of [39] distinguish between four penalty cost types (the same approach is used by [49]):

(1) Penalty cost for delivery failures, which is a fixed cost, proportional to the number of delivery failures.
(2) Penalty cost caused by the sales opportunity loss, proportional to the number of products not delivered.
(3) Penalty cost to compensate extra working time that is spent to re-produce the safety stock to its nominal level.
(4) Cost of safety stock, which is proportional to the buffer size.

Here, the (1) and (2) abovementioned approaches are combined, and at the same time, focus is placed on only those penalty types that are relevant in the case of BTO companies who are sharing resources via a platform. Penalty cost types (3) and (4) are not really applicable for BTO companies operating a Just-in-Time production system, because they do not have a safety stock.

From [39], cost type (1) could be applied to failed deliveries due to job cancellation or delivering outside the time window, as shown in Figure 16. In reality, a failed delivery can cause the failure to fulfil the whole order; thus, the lead company could also lose trust towards the customer, too.

Cost type (2) is for late deliveries to compensate the sales opportunity loss of the customer – but this does not include the loss of trustfulness towards the company that the customer is delivering late and a penalty that the lead company may pay for late delivery. In [39], authors do not include any penalty costs related to early delivery; however, in [35] it is also highlighted that additional costs arise in connection with holding too early delivery (inventory costs. They present a model including nonlinear early and late delivery costs (holding/backlogging issues for the customer), which are taken as a product of a linear function of delivery lead time and a nonlinear function of the delivery lot size. They state that the efficiency of a supply chain network is greatly influenced by the reliability of the supply process and highlight that the success of a supply chain lies beneath the proper timing of delivery of goods to the intermediate parties. They introduce a delivery tolerance period, where no penalty must be paid, and mention that delivery time inaccuracy could come from inaccuracy of production lead time or transportation time. Here the focus is placed on the sum of these two and assume that the earliness or lateness with deliveries is the offeror's responsibility no matter what the cause is. Nevertheless, early delivery is not only generating additional inventory costs: in the case of food or medicine, the quality of the products could decrease during the storage period – which also has to be included in the penalty cost [3].

Taking the referenced models into consideration, by combining and extending them, the penalty costs applied in the presented financial model are the following, calculated on the basis of the delivery time ${t_{d_j}}^B$.

If ${t_{d_j}}^B < {t_{sa_j}}^B$ or ${t_{d_j}}^B > {t_{ea_j}}^B$, the delivery is *failed,* and the penalty cost is computed by multiplying the P^o profit that could have been created by selling the products (calculation is described in the *Sales income, profit* subsection) with the $c_{fail_j}^A$ failed delivery factor for company A (see Eq. (10). Here all the additional costs, such as trust loss caused by the failed delivery, are included in $c_{fail_j}^A$. It could also happen that a resource offeror company cancels a job even though it has signed the contract. For cancellation, the same penalty is issued, but it is penalized from the trustfulness perspective, also.

$$c_{pen_j}^B = P^o \cdot c_{fail}^A \qquad \text{Eq. (10)}$$

If ${t_{sa_j}}^B < {t_{d_j}}^B < {t_{so_j}}^B$, it is an *early delivery,* here $\alpha_j^B \cdot c_{inv_j}^A$ (the number of products multiplied by the inventory cost per product) and the possible $\alpha_j^B \cdot c_{qual_j}^A$ quality reduction costs (number of products multiplied by the quality reduction cost per product) of lead company A for the early time interval is incurred to resource offering company B as a penalty. In the case of products whose quality does not decrease over time, $c_{qual_j}^A = 0$. Here it is assumed that the offerors do not store the products that were finished early but ship them immediately to the lead company.

$$c_{pen_j}^B = \alpha_j^B \cdot (c_{inv_j}^A + c_{qual_j}^A) \cdot ({t_{so_j}}^B - {t_{d_j}}^B) \qquad \text{Eq. (11)}$$

If ${t_{eo_j}}^B < {t_{d_j}}^B < {t_{ea_j}}^B$, it is a *late delivery,* and $\alpha_j^B \cdot c_{loss_j}^A$ sales opportunity loss of lead company A for the late time interval is issued to resource offering company B as a penalty:

$$c_{pen_j}^B = \alpha_j^B \cdot c_{loss_j}^A \cdot \left({t_{d_j}}^B - {t_{eo_j}}^B \right) \qquad \text{Eq. (12)}$$

If $t_{so_j}{}^{B} < t_{d_j}{}^{B} < t_{eo_j}{}^{B}$, the delivery is *on-time*; thus, the penalty cost is equal to zero:

$$c^{B}_{pen_j} = 0.$$

Eq. (13)

9.4. Distribution costs

According to [71], the transportation cost of an order depends on the size of the shipped batch (larger batch has a lower cost per unit), but inventory costs occur when creating too large batches as the products have to be stored until the batch size is reached. This way, an optimal batch size can be calculated. In the referenced paper, three main cost types are differentiated regarding transportation: arrival cost, inventory cost and delivery cost. Here, the focus is not on determining the optimal batch size and simplifying transportation costs, which depend mainly on the distance and transport type, the number of products delivered, and the specific company's administrative costs. Also, as one could see above, inventory costs are treated separately from shipment costs in the presented model. Regarding distribution costs, *transportation cost* (which incurs when the lead company ships the products of a completed order to the customer) and *transportation consolidation cost* (which incurs when the members of the federation are shipping the products of a specific job to each other) are distinguished.

9.4.1. Transportation cost

Transportation cost $c^{AC,o}_{transport}$ for an order o is computed by multiplying the d_{AC_o} distance between lead company A and customer C, with the $c_{transport}{}^{o}$ cost factor for a specific transport type per product, and the α^{o} number of products in the specific order. In addition, a fixed shipment sending $c_{shipment}{}^{A}$ cost incurs for the lead company A for each order:

$$c^{AC,o}_{transport} = d^{\mathrm{AC}} \cdot c_{transport}{}^{o} \cdot \alpha^{o} + c_{shipment}{}^{A}$$

Eq. (14)

9.4.2. Transportation consolidation cost

Transportation consolidation cost is calculated similarly to transportation cost, but for shipment of products in case of outsourced jobs between federation members: d^{AB} distance between lead company A and resource offeror company B is multiplied by the $c_{transport_j}$ cost factor for a specific transport type per product and by the α^{B}_{j} number of products in the specific part of job *j*. In the case of each outsourced job, $c_{shipment}{}^{B}$ fixed shipment sending cost also incurs for company B. For lead company A, only the fixed $c_{arrival}{}^{A}$ arrival cost incurs in case of each outsourced job part.

$$c^{B,j}_{tr.cons} = d^{AB} \cdot c_{transport_j} \cdot \alpha^{B}_{j} + c_{shipment}{}^{B}$$

Eq. (15)

$$c_{tr.cons}^{A,j} = c_{arrival}{}^{A} \qquad \text{Eq. (16)}$$

Of course, transportation consolidation costs do not appear in the case of insourced jobs. Nevertheless, the higher number of parts the platform separates a job (because one company does not have the required resource load), the more transportation consolidation cost incurs for the companies.

9.5. Administration costs

9.5.1. Management cost

Management costs cover the different cost types that incur for federation members related to the Platform and the planning costs when sending requests and offers:

(1) *entrance fee* that incurs when the company is joining or re-joining the federation for the nth time (c_{entr}),

(2) *participation fee* that has to be paid regularly after a certain time unit (c_{part}),

(3) *administrative costs* (paid to the Platform) of sending one offer ($c_{offer,a}$), one request ($c_{request,a}$) and establishing a contract ($c_{contract}$), multiplied by the number of offers, requests, and contracts (${n_{offer}{}^{A}}_{x}$, ${n_{request}{}^{A}}_{x}$, ${n_{contract}{}^{A}}_{x}$) sent to the platform during the time that the company was the member of the federation for the xth time,

(4) *resource planning costs* of sending offers ($c_{offer,p}$) and requests ($c_{request,p}$).

The $c_{man_x}^{A}$ total management cost of federation member A for the time it entered the federation for the xh time, is calculated by summarizing the entrance fee, the participation fee multiplied with the T_x time interval since when the company is a member of the federation for the xth time, and the abovementioned administrative costs:

$$c_{man_x}^{A} = c_{entr} + c_{part} \cdot T_x + \left(c_{offer,a} + c_{offer,p}\right) \cdot {n_{offer}{}^{A}}_{x} + \left(c_{request,a} + c_{request,p}\right) \cdot {n_{request}{}^{A}}_{x} + c_{contract} \cdot {n_{contract}{}^{A}}_{x} \qquad \text{Eq. (17)}$$

9.5.2. Order consolidation cost

Order consolidation cost is incurring for the lead company as a cost for administrating the consolidation of different jobs and job parts in connection with each order. The total order consolidation cost for lead company A is calculated by summarizing all the consolidation costs for the jobs that are part of the orders that company A received:

$$c_{ord.cons}^{A} = \sum_{j=1}^{J} c_{ord.cons_j}^{A} \qquad \text{Eq. (18)}$$

9.6. Incomes of the participants

In this section, the following income types for the federation members are introduced: *sales income*, *penalty income* and *order completion income*.

9.6.1. Sales income, profit

In the model, sales income is received after completing an outsourced *job*, and lead companies also receive an income after a *whole order* is completed and the finished products are delivered to the customers.

The first type of sales income in connection with a specific job j is paid by the lead company to the resource offeror company and means the selling price of the products that were delivered (${i_{sales_j}}^B$) to the lead company. It is calculated by multiplying the sum of ${c_{manuf_j}}^B$ manufacturing cost and $c_{WIP,j}^B$ WIP inventory cost of the job, by the $1+p^B$ profit margin of company B (if its profit margin is 10%, the company receives the 110% of these costs as sales income).

$$ {i_{sales_j}}^B = ({c_{manuf_j}}^B + c_{WIP,j}^B) \cdot (1+p^B) \qquad \text{Eq. (19)} $$

Profit generated by company B from completing (a part of) job j:

$$ P_j^B = ({c_{manuf_j}}^B + c_{WIP,j}^B) \cdot p^B \qquad \text{Eq. (20)} $$

Before selling the whole order to the customer, first lead company A pays for the outsourced jobs. Here it is assumed that the profit generated directly from these jobs, in general, is equal to zero, as after paying for the manufacturing, logistics, inventory, administration costs – even if the partner company works less expensively – it is not realistic to gain a considerable profit directly from these interactions. Consequently, the lead company can generate direct income mainly based on the additional value of the insourced parts. Nevertheless, income is also generated from consolidating all the jobs and delivering the whole order to the customer or assembling the parts that other partners provided. This income is proportional to the α^o number of products in the order, and also with factor i^o additional income per product. In total, the ${i_{sales}}^{A,o}$ sales income received by lead company A after selling order o is equal to the sum of the income that is proportional to the size of the whole order ($\alpha^o \cdot i^o$) and the income that is generated by the insourced jobs.

$$ {i_{sales}}^{A,o} = (1+p^A) \cdot \left(\alpha^o \cdot i^o + \textstyle\sum_{j=1}^{J}({c_{manuf_j}}^A + c_{WIP,j}^A)\right) \qquad \text{Eq. (21)} $$

The profit for company A in connection with order o is calculated as follows:

$$P^{A,o} = p^A \cdot \left(\alpha^o \cdot i^o + \sum_{j=1}^{J}({c_{manuf}}_j^{\ A} + c_{WIP,j}^A)\right) \qquad \text{Eq. (22)}$$

9.6.2. Penalty income

Penalty income is paid by resource offerors to the lead company to compensate costs due to early or late delivery. This is equal to the cost that was described in subsection 9.3 (penalty cost): the penalty income for lead company A in connection with the job part that was outsourced to company B (${i_{pen}}^{A,B}_{\ \ j}$) is equal to the penalty cost that company B has to pay in connection with the same job (${c_{pen}}_j^B$).

$${i_{pen}}^{A,B}_{\ \ j} = {c_{pen}}_j^B \qquad \text{Eq. (23)}$$

9.7. Incomes and costs of the platform

In this subsection, costs associated with marketing, operating and maintenance, and incomes of the platform (paid by federation members) will be discussed. Figure 37 summarizes these costs and incomes:

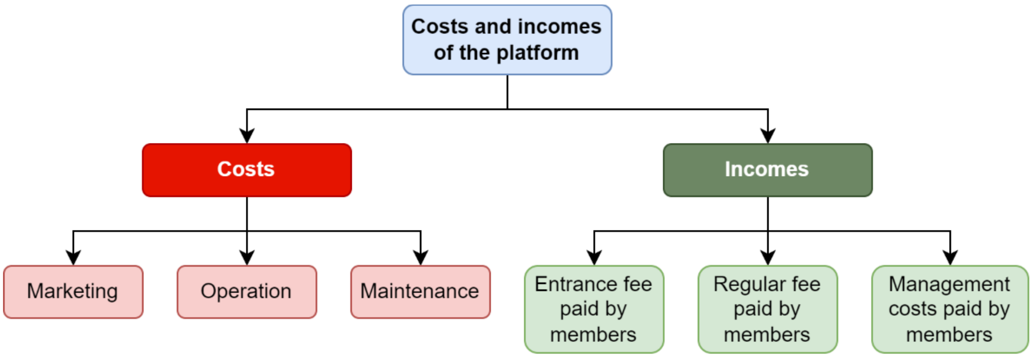

Figure 37. Costs and incomes of the platform

In the presented financial model, the incomes of the platform are equal to the management costs paid by the federation members to the platform (sum of management costs incurring for the companies, without resource planning costs):

$$i^{platform} = \sum_k^K (c_{entr}^k + c_{part}^k \cdot T + c_{offer,a}^k \cdot n_{offer}^k + c_{request,a}^k \cdot n_{request}^k + c_{contract} \cdot n_{contract}^k) \qquad \text{Eq. (24)}$$

The costs of the platform consist of the marketing costs (c_{mar}) to attract a higher number of participants, and the operational and maintenance costs of the IT system of the platform (c_{op} and c_{maint}, e.g., maintenance of server and web page). The platform could be profitable if the number of participants is high enough to compensate for these costs.

$$c^{platform} = c_{mar} + c_{op} + c_{maint} \qquad \text{Eq. (25)}$$

9.8. Is it worthwhile to join a platform?

As mentioned, profit is accumulated in three places in the presented model: in the lead companies, in the offering companies, and in the platform. The role of the platform is to help companies with their resource issues, not to gather as much profit as possible (and consequently, weaken the companies). Collecting too high entrance and participation fees and administration costs would lead to lower number of participants and interactions and would result in lower profit.

For a company, to be worthwhile joining the platform, the additional income generated by the higher resource utilization level (i.e., being able to complete more incoming orders and completing jobs outsourced by others) has to cover the additional costs that incur due to platform-based resource sharing.

According to the presented model, additional income can be generated by:

- Completing additional orders due to the more complex matching logic of the platform. As mentioned in the *Incomes of the participants* section, income is generated by (1) insourced jobs and (2) order completion.
- Regularly sending offers to the platform and completing additional jobs outsourced by other companies. Since the platform can combine offers from separate companies to fulfil one request, smaller amount of offered capacities could be matched with requests, also.
- Penalty income in case of a partner delivers inaccurately (note: this is spent on covering the additional inventory costs or lost sales).

Additional costs are:

- Management costs including entrance fee, regular participation fee, and administrative costs of sending messages (requests, offers, contracting).
- Manufacturing, inventory and distribution costs caused by completing additional incoming orders and jobs outsourced by other federation members.
- Additional penalty costs might be paid due to inaccurate job completions.

If a company wants to decide whether it is worthwhile joining the platform, it must consider the following aspects:

Completing additional orders

Does the company receive orders (that can be completed only by outsourcing a part of them) often enough? Do the other federation members have the appropriate resource types to complete these orders? Do they have enough free capacity to offer their resources?

Completing outsourced jobs

Does the company have additional resources it can regularly offer? Do the other federation members receive orders that could require these resources? Can these resources be offered at the appropriate price level? Is the company reliable enough in terms of delivery accuracy?

Balance between incomes and costs

As described above, higher resource utilization level and additional incomes come with additional platform-related costs, also. A company, to decide whether it is beneficial to join or leave the federation, must analyze the incoming orders and interactions with others from the past and make forecasts for the future to be able to determine the possible benefits.

9.9. Simulation experiments – financial model

To investigate the financial model for the resource sharing federation, use cases were created using the extended version (including cost functions, too) of the agent-based simulation model described in Chapter 6.

Table 14 summarizes the main input parameters for simulation experiments. For the parameters determined by using a truncated normal distribution, the mean and sigma values are included in the table (for the constant ones, sigma is 0). In these cases, the difference between the lower and upper bounds of the distribution from its mean is sigma/2. Regarding the partner selection, the hypothesis is that the best results are expected if companies of the same region and with a similar product portfolio take part in a resource sharing cooperation. This way, both transportation and machine setup efforts are minimized and lead to fewer additional costs [8]. Thus, in the model, 10 companies – located in county seats in Hungary on the map – were implemented, and all of them had 16 different resource types out of the required 20 that were used to compose the orders. One order included 3 job types that may require different resource types to complete. The resource type of a job was chosen randomly from the 20 possibilities.

As indicated in Table 14, each company received an order every time unit that included 3 jobs, consisting of in an average of 400 products requiring 4 pcs. of resources for in average 12 time units. The platform could combine a maximum of 3 offers to fulfil the requirements of a request in order to reduce administrative and logistics costs. The planning horizon, i.e., the length of the time interval for which the companies could offer their resources in advance, was set to be 40 time units. The on-time delivery (2 time units) and the delivery acceptance (4 time units) intervals are applied as follows: the middle of the interval is the accurate delivery time in the contract, as it can be seen in Figure 16.

Regarding the cost parameters, the entrance fee to be paid when entering the federation was 10 monetary units, the regular fee was 5; the latter was paid after every 20 time units. The experiments were run for a period of 1000 time units. In the case of outsourced jobs, it was assumed that the manufacturing cost per product for the outsourced jobs was 90% of the insourced ones – meaning the partner companies work less expensively. Outsourcing in reality often happens towards a company whose core business is the specific job type and thus can operate its resources in a more efficient way.

Table 14. Input parameters for simulation experiments

Parameter	Mean	Sigma	Unit
General simulation parameters			
Order interarrival time	1	0	tu*
Incoming order length	12	6	tu
Incoming order resource quantity	4	2	pcs.
Number of products in one order	400	200	pcs.
Number of jobs in one order	3	0	pcs.
Max. number of offers to be combined by the Platform	3	0	pcs.
Planning horizon	40	0	tu
Probability of cancelling an order (for all companies)	2	0	%
Simulation time	800	0	tu
Length of on-time delivery interval	2	0	tu
Length of delivery acceptance interval	4	0	tu
Cost parameters			
Entrance fee to join the federation	10	0	mu**
Regular fee for federation members	5	0	mu
Regular fee payment time interval	20	0	tu
Initial capital for companies	100	0	mu
Manufacturing cost per product for the outsourced jobs compared to insourced ones	90	0	%
Profit margin for all companies	10	0	%

*time units, **monetary units

After running the model for 1000 time units, using the parameters introduced in Table 14, cumulated costs and incomes of Company1 are shown in Figure 38.

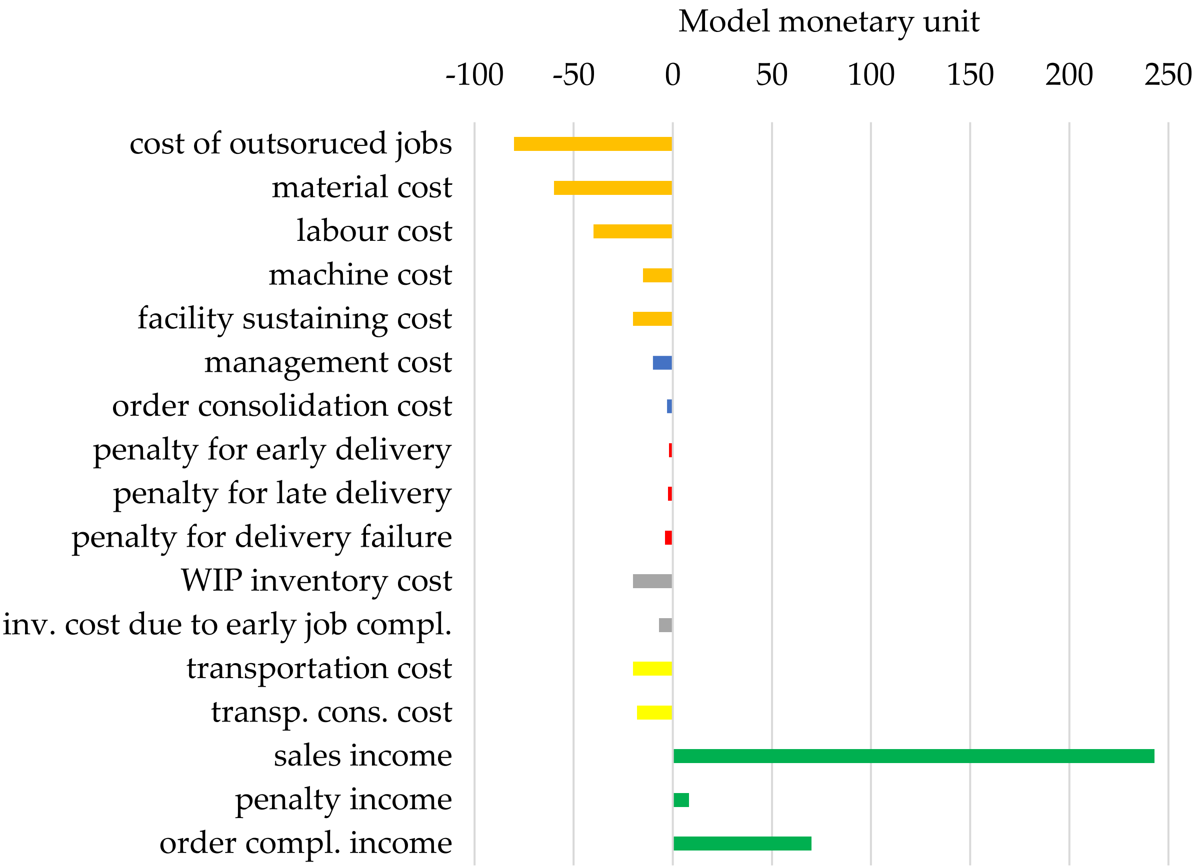

Figure 38. Costs and incomes of Company 1 in a test case

9.9.1. Effect of changing order interarrival time

In this experiment, a realistic scenario was investigated when a company is continuously a member of the federation, but it receives a lower number of orders for a given time period due to a temporary demand decrease. The aim was to examine the effect of this on the company from the monetary perspective.

For one of the 10 companies (Company1) the order interarrival time was increased temporarily at time unit 200 from the original value 1, meaning the company received orders less frequently for 200 time units; and from 400 time units, it was set back to 1. One can see in Figure 39, that in the first two cases (order interarrival time is 1 or 2 time units), the revenue of the company continuously increased. In contrast, when it was receiving orders less frequently (order interarrival time is 3 or more time units) it can be noticed that in the short term, it is not worthwhile for the company to be the part of the federation because of the decreased incomes (due to lower resource utilization) are not covering the management fees. However, in the long term, it is worthwhile joining the federation because the overall balance is positive on the horizon of 1000 time units.

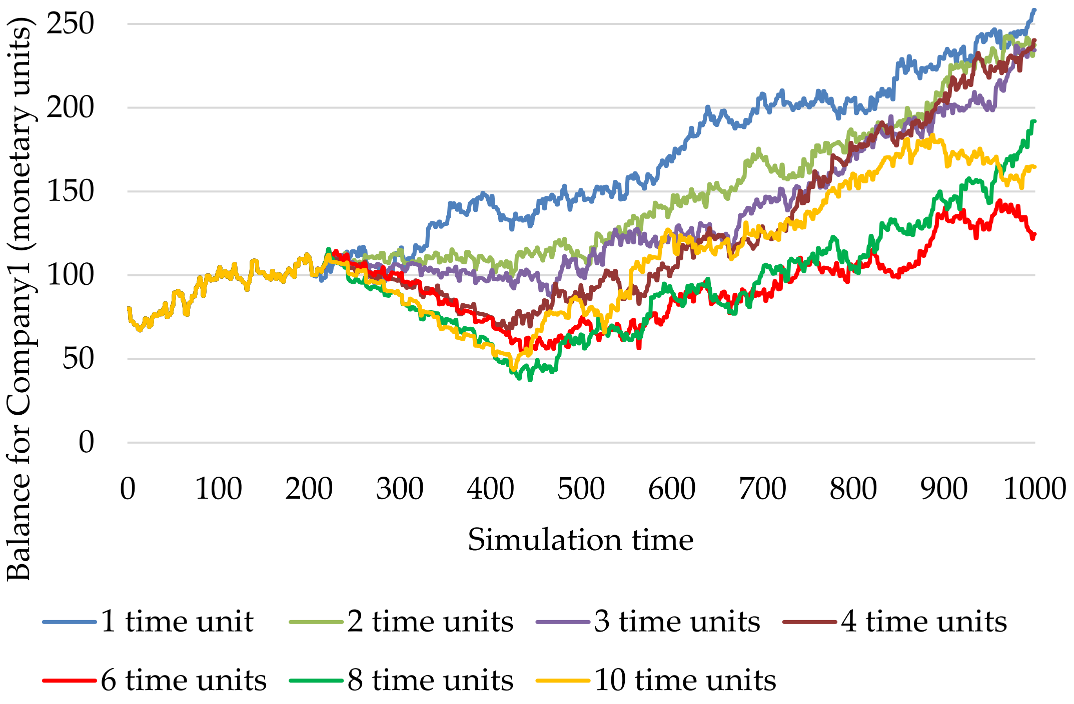

Figure 39. Effect of temporarily increasing order interarrival time (temporal decrease in the load)

The experimental results confirm that accurate demand forecasts are highly important to the companies. Up to a certain order frequency limit, it is worthwhile to be a member of the federation, but of course, the platform cannot solve the problem of large-scale demand decreases: in such cases, if a company may exit the federation temporarily not pay the participation fee for this period. Nevertheless, one goal for the platform is to motivate the companies not to quit, for example, by raising the entrance fee that must be paid again when re-joining.

9.9.2. Effect of the price of outsourced jobs

In real industrial environments, companies outsource a job only if the manufacturing cost of the outsourced job is lower than the cost of manufacturing it internally, and the difference at least covers the additional costs (transportation, administration) the outsourcing occurs. Of course, in some cases, it is reasonable to undertake an order even if the profit related to it is negative, not to lose trust towards the customer, and to have a long-term successful relationship.

In the next experiment, another federation member, Company2 was tested, and the extent to which the manufacturing price of the outsourced orders are less expensive compared to the manufacturing price of insourced orders was investigated. Here it was assumed that companies outsource a job in any case when they cannot complete an order with their own resources, without regarding the financial balance of doing so. As one can see in Figure 40, in case of this ratio is 90%, it is worthwhile to be a member of the federation and outsource orders, but as the ratio increases, the company becomes lossmaking. If the partners are using the same unit prices as the resource requester company (100%), depending on the contents of the orders, there are time intervals where the income is positive (higher number of insourced jobs), but in other cases, the income is negative (higher number of outsourced jobs, a loss is created).

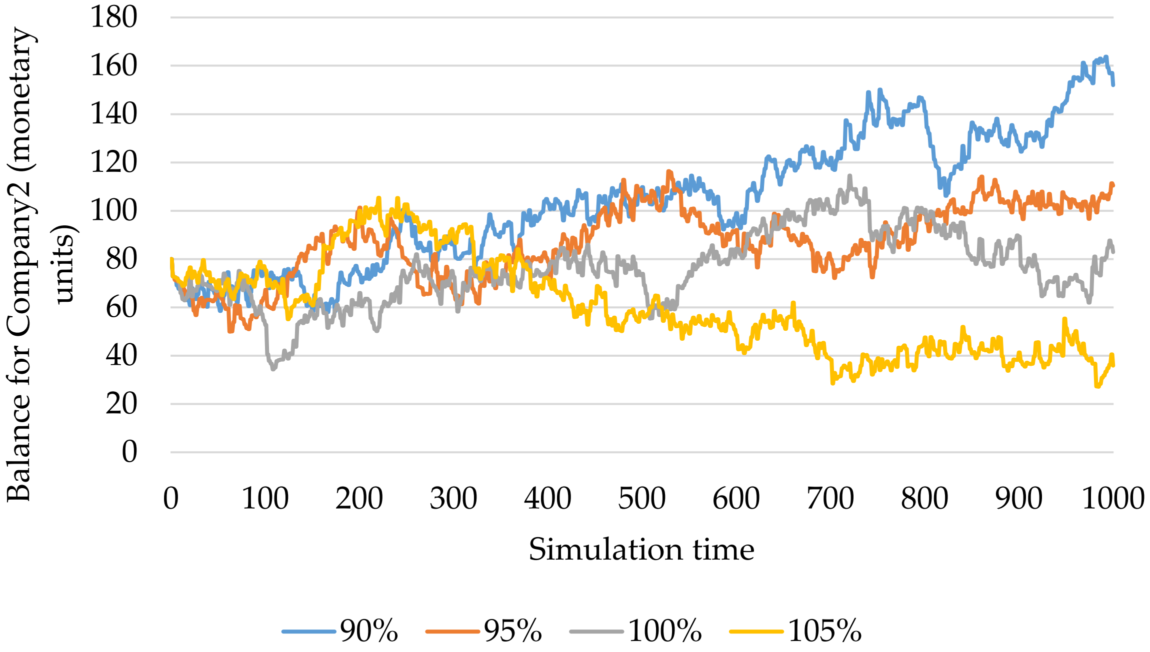

Figure 40. Impact of the price of outsourced jobs

In such a resource sharing federation, depending on the extent of the additional costs, a company could find the specific cost level where it is reasonable to outsource orders, taking the loss of trustfulness towards the customers into consideration. This cost level – based on the cost functions introduced in the financial model – depends on the internal manufacturing costs, the additional costs incurring due to outsourcing (administration and management costs), and the income that can be realized by completing the order.

10. Conclusions and outlook

10.1. New scientific results

Thesis 1:

In the case of customer orders that are volatile in terms of the number of required resources and the time to use them, occasionally resulting in underutilized resources or resource shortages, the platform-based resource sharing between manufacturing companies is achievable with the mechanism shown in Figure 14. The essence of the mechanism is that companies with a shortage or surplus of resources can send requests and offers to a central platform, whose role is to provide offers, which meet the requirements of the requests. The requesting company selects the best offer based on its own preferences and, once the job is completed (or cancelled), evaluates the partner's performance – which will be used for decision-making in the future.

Thesis 2:

When sharing production resources, the average service level of the participants increases if the trustfulness of the partner in terms of meeting delivery deadlines is considered in the decision between resource offers.

Trustfulness is the weighted average of the following two values on a scale of 0-100 (weights are determined by the company's own preferences):

1. Trust: an internal, subjective, *cumulated* value determined by the resource requesting company, based on scores of direct bilateral interactions.
2. Reputation: an external, public, *cumulated* value, including interactions with other partners, also. The platform calculates it based on the scores sent by resource requesters and makes them available to all participants.

Score for a *given job* can be determined by the following calculation:

$$\text{If } \delta_j < L_j \qquad r_{t_r,j}^{A,B} = 100\left(1 - \frac{\delta_j}{L_j}\right) \cdot \gamma \cdot \mu_j \tag{1}$$

$$\text{If } \delta_j \geq L_j \qquad r_{t_r,j}^{A,B} = 0 \tag{2}$$

where $r_{t_r,j}^{A,B}$ is the score given to the offeror company B at time t_r for the work j requested by company A. δ_j is the absolute deviation from the bounds of the acceptance interval defined by A for j if the time of completion is outside of it; otherwise, $\delta_j = 0$.

L_j is the contractually agreed period of time needed to complete j. γ is the penalty factor to give higher penalties for inaccuracies (value between 0 and 1), μ_j is the quality multiplier (value between 0 and 1), which the company determines for j on the basis of its own set of requirements.

To calculate the cumulated *trust* or *reputation* of a company at time t, the weight of the score given at time t_r is determined as follows:

$$w(t_r, t) = \frac{\theta}{\theta + (t_r - t)} \tag{3}$$

where θ is the smoothing factor. Based on this, at time t, the trust calculated by company A about company B can be calculated by determining the weighted average of the scores given in t_r time points, before time t:

$$\varphi^{A,B}(t) = \frac{\sum_{t_r \le t} w(t_r, t) \cdot r_{t_r,j}^{A,B}}{\sum_{t_r \le t} w(t_r, t)} \tag{4}$$

Reputation is calculated in the same way, but by the platform, and using scores about all interactions of company B.

Withdrawal of resource offers or requests or cancellation of a contract is also penalized. In this case, the cumulative trustfulness of the offeror will be reduced by an amount to be agreed by the participants.

Based on the results of a series of experiments conducted by an agent-based simulation model including the trust and reputation model described above, the resource sharing mechanism has the following properties:

1. Taking trustfulness into account has a beneficial effect on the overall performance of resource sharing companies.
2. The effect of taking trustfulness into account is smaller when cooperating firms are overloaded, as members are forced to work with less reliable partners to fulfil orders.
3. The positive effect of taking trustfulness into account is proportional to the size of the federation.

Thesis 3:

According to the comparison of the direct communication- and platform-based resource sharing mechanism, performed by agent-based simulation experiments, the following conclusions can be drawn, in the case of applying the same set of orders for the same companies:

1. In the case of platform-based resource sharing, the average resource utilization of companies is about 30% higher, due to the complex request-offer matching logic of the platform (combining and dividing offers according to the request).
2. The average service level in terms of meeting deadlines is 4-5% higher in the case of direct communication-based resource sharing, as reliable companies are less loaded due to the lower number of matches and can therefore offer their resources more often. However, the rate of orders fulfilled is lower in this case.
3. In the case of platform-based resource sharing, the communication load on the group of companies is significantly lower, it is approximately directly proportional to the number of members in the federation. In contrast, the relationship is approximately quadratic for the direct communication mechanism.

Thesis 4:

The financial model developed for fluctuating orders and platform-based resource sharing, which details the incomes and costs of the participants and the platform, as well as the parameters of external orders, is suitable for the following (as shown by a series of experiments with an agent-based simulation model):

1. To determine whether a company should join the platform in the short or long term.
2. To determine under what conditions it makes sense to outsource work through the platform.

10.2. Application of the results

The direct antecedent of the research was the research project titled *Enterprise Collaboration Space Design for Crowdsourced Manufacturing*, which was completed in 2016-2017 in collaboration with SZTAKI and Hitachi, Japan. As a continuation of this topic, the results presented here and were closely related to several domestic and international research projects:

1. INEXT A5 trust mechanisms: Research on prime exploitation of the potential provided by the industrial Digitalization (2018-2024)
2. OTKA Optimization for Sustainable Supply Chains (2019-2022)
3. EPIC Excellence in Production Informatics and Control (2017-2024)
4. Research on cooperative production and logistics systems to support a competitive and sustainable economy (2022-2025)

Using the platform-based resource sharing mechanism, and the associated trustfulness rating system model, a web-based platform can be created in reality. It can act as a virtual capacity extension of companies to assist them in operating more efficiently and utilizing their resources as much as possible, particularly in the cases of fluctuating or unforeseeable customer orders.

Due to the evolution of manufacturing and production informatics in recent years, the information technology background needed to operate these platforms has become available, as evidenced by the presence of companies offering Manufacturing-as-a-Service solutions (e.g., *Spanflug, Up2parts, Daedalus, Xometry, Proto Labs, Haizol*) and projects aimed at developing and operating platforms (e.g., *Catena-X, Smart Factory Web)*. Most of the platforms, after receiving an order, automatically assign manufacturer(s) to it, based on an internal assessment system and with the use of artificial intelligence. This way, they immediately associate a price and a deadline with the offer.

The results of this research support the viability of platforms based on resource sharing (Thesis 3 and 4), provide a basis for the development of a related communication structure (Thesis 1) and a corresponding rating system (Thesis 2).

10.3. Summary of the thesis

In the thesis, a novel, platform-based resource sharing mechanism was introduced that helps manufacturing companies in effectively utilizing their resources even in an environment with non-predictable, volatile customer demands.

In the first part of the thesis, the relevant literature was reviewed, focusing on distributed production concepts trying to cope with nowadays challenges. Benefits of resource sharing for Build-To-Order manufacturing companies were discussed, examples from other researchers were mentioned. Agents, multi-agent systems and agent-based simulation modelling were also investigated, as agent-based simulation was applied in this thesis to test and validate hypothesizes about the resource sharing mechanisms. One of the novelties in the proposed model is considering trustfulness, thus, trust and reputation systems (including classification, attack types and defense mechanisms, security issues and case studies) were also detailed. Cost models from the literature were also summarized, and the novelty of the research presented in this thesis – considering capacity constraints, including trust and reputation in decision making, and the financial model – were highlighted.

Then, a new direct exchange-based resource sharing model was introduced: basic concepts were defined, the model and the communication structure were detailed, and the calculation of available resources, as well as the decision-making logic were presented. A novel platform-based resource sharing mechanism was also introduced, the central role of the platform and the communication logic were detailed. The agent-based simulation model was also described: the structure of the model, agent types and the difference between modelling reliable and non-reliable participants were discussed.

The suggested trust model was also presented in detail. Experiments were performed to investigate the effect of considering trustfulness and the impact of lead time prediction accuracy in resource sharing. Based on the results it was shown that, taking trustfulness into account has a beneficial effect on the overall performance of resource sharing companies. However, the effect of taking trustfulness into account is smaller when cooperating firms are overloaded, as members are forced to work with less reliable partners to fulfil orders. It was also shown that the positive effect of taking trustfulness into account is proportional to the size of the federation.

The platform-based and direct exchange-based mechanisms were compared using different aspects of resource sharing, and average resource utilization, service level of the participants and communication load of the two solutions were investigated with simulation experiments. Agent-based simulation experiments have shown that the difference between average resource utilization is approx. 30% in favor of platform-based resource sharing, due to the more complex matching logic. However, the average service level is approx. 4~5% higher in the direct exchange-based mechanism, since lower resource utilization causes that reliable companies have more free capacities that can be requested by others. Communication load is one order of magnitude lower in the platform-based case, because the companies are sending messages only to the parameter instead of each other.

Finally, a new financial model was also presented and tested, which includes the incomes and costs of the companies and the platform. Manufacturing, inventory, penalty, distribution, administration costs, sales and penalty incomes were formulated in the model, and incomes of the platform were also detailed. Financial advantages and disadvantages of joining a platform were discussed, as well. Simulation experiments were conducted to investigate the effect of changing order interarrival times and the price of outsourced jobs. Based on the results, although it may be unprofitable for a company to join the platform in the short term, it can be profitable in the long term, depending on the frequency of incoming orders (the load of the federation members). The manufacturing cost of outsourced jobs also has a strong impact on the participants' profit and, above a certain cost limit, may make it unprofitable to outsource jobs in the long run.

10.4. Future work and outlook

The platform-based resource sharing model could be further developed in different directions to be more realistic. One of these is to apply more complex mechanisms, such as multi-criteria decision-making for the federation members, where not only trustfulness and price but for example, environmental awareness (preferring shorter logistical routes) could be a new aspect. Another direction could be to investigate approaches in financial and commodity markets.

As it was mentioned in the thesis, real systems could be vulnerable to malicious companies that are trying to affect the operation of the mechanism to get some advantage. For example, by providing false ratings, sending fake orders or requests – but these activities could be handled and penalized if the platform is monitoring each company's activities and is able to notice patterns in the ratings and messages.

Another promising research direction could be the investigation of the bullwhip effect in the case of interdependent jobs: how can the platform help if one job is failed or cancelled, causing re-planning resources and reorganizing production for other companies planning to work on jobs from the same order. This aspect could be examined in a more detailed way if the simulation model contains not only the parameters of the companies but the simulation model of their manufacturing system, too.

An interesting experiment would be to investigate how companies can increase their trustworthiness by overplanning capacities and this way delivering more accurately. Overplanning can occur additional costs for them, but in return, due to the higher trustfulness level, they would win more jobs causing higher resource utilization and higher profit.

Bibliography

[1] AACE International (2022). Cost Engineering Terminology: AACE International Recommended Practice No. 10S-90 - TCM Framework: General Reference (https://web.aacei.org/docs/default-source/rps/10s-90.pdf). Accessed on 28 February 2023.

[2] Aelker, J., Meister, V., Forster, C., Zapp, M., & Bauernhansl, T. (2013). Opposites Attract: An Approach to Collaborative Supply Chain Management between Semiconductor and Automotive Companies. *Industrial and Systems Engineering Review*, *1*(1), 13-30.

[3] Akkerman, R., Farahani, P., & Grunow, M. (2010). Quality, safety and sustainability in food distribution: a review of quantitative operations management approaches and challenges. *OR spectrum*, *32*, 863-904.

[4] Aminzadegan, S., Tamannaei, M., & Rasti-Barzoki, M. (2019). Multi-agent supply chain scheduling problem by considering resource allocation and transportation. *Computers & Industrial Engineering*, *137*, 106003.

[5] Angrish, A., Craver, B., Hasan, M., & Starly, B. (2018). A case study for Blockchain in manufacturing: “FabRec”: A prototype for peer-to-peer network of manufacturing nodes. *Procedia Manufacturing*, *26*, 1180-1192.

[6] Bachmann, S., & von Hattburg, A. T. (2021). Effiziente Krisenkommunikation–transparent und authentisch. *Springer Books*.

[7] Balke, T., König, S., & Eymann, T. (2009). A survey on reputation systems for artificial societies. *Technical Report 46. Bayreuth University*.

[8] Becker, T., & Stern, H. (2016). Impact of resource sharing in manufacturing on logistical key figures. *Procedia Cirp*, *41*, 579-584.

[9] Ben-Arieh, D. (2000). Cost estimation system for machined parts. *International Journal of Production Research*, *38*(17), 4481-4494.

[10] Bendul, J. C., & Blunck, H. (2019). The design space of production planning and control for industry 4.0. *Computers in Industry*, *105*, 260-272.

[11] Berman, S., & Marshall, A. (2014). The next digital transformation: from an individual-centered to an everyone-to-everyone economy. *Strategy & Leadership*, *42*(5), 9-17.

[12] Bode, C., & Wagner, S. M. (2015). Structural drivers of upstream supply chain complexity and the frequency of supply chain disruptions. *Journal of Operations Management*, *36*, 215-228.

[13] Bolton, G., Greiner, B., & Ockenfels, A. (2013). Engineering trust: reciprocity in the production of reputation information. *Management science*, *59*(2), 265-285.

[14] Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the national academy of sciences*, *99*(suppl_3), 7280-7287.

[15] Brinkhoff, A., Özer, Ö., & Sargut, G. (2015). All you need is trust? An examination of inter-organizational supply chain projects. *Production and operations management*, *24*(2), 181-200.

[16] Busch, A., & Dangelmaier, W. (Eds.). (2013). Integriertes Supply Chain Management: Theorie und Praxis effektiver unternehmensübergreifender Geschäftsprozesse. *Springer-Verlag.*

[17] Cai, S., Goh, M., De Souza, R., & Li, G. (2013). Knowledge sharing in collaborative supply chains: twin effects of trust and power. *International journal of production Research*, *51*(7), 2060-2076.

[18] Caniato, F. F. A., & Rice, J. (2003). Building a secure and resilient supply chain. *Supply Chain Management Review*, *7*(5), 22-30.

[19] Capaldo, A., & Giannoccaro, I. (2015). Interdependence and network-level trust in supply chain networks: A computational study. *Industrial Marketing Management*, *44*, 180-195.

[20] Chan, W. K. V., Son, Y. J., & Macal, C. M. (2010). Agent-based simulation tutorial-simulation of emergent behavior and differences between agent-based simulation and discrete-event simulation. *Proceedings of the 2010 winter simulation conference*, 135-150.

[21] Chang, E., Hussain, F., & Dillon, T. (2006). Trust and reputation for service-oriented environments: technologies for building business intelligence and consumer confidence. *John Wiley & Sons.*

[22] Chen, Q., Xu, Q., & Wu, C. (2019). Dynamic matching in cloud manufacturing considering matching costs. *Complexity*, *2019*. 1-16.

[23] Chang, L., Ouzrout, Y., Nongaillard, A., Bouras, A., & Jiliu, Z. (2014). Multi-criteria decision making based on trust and reputation in supply chain. *International Journal of Production Economics*, *147*, 362-372.

[24] Cheikhrouhou, N., Pouly, M., & Madinabeitia, G. (2013). Trust categories and their impacts on information exchange processes in vertical collaborative networked organisations. *International Journal of Computer Integrated Manufacturing*, *26*(1-2), 87-100.

[25] Chiadamrong, N., & Wajcharapornjinda, P. (2012). Developing an economic cost model for quantifying supply chain costs. *International Journal of Logistics Systems and Management*, *13*(4), 540-571.

[26] Chida, T., Kaihara, T., Fujii, N., & Kokuryo, D. (2019). Stability analysis on resource matching in crowdsourced manufacturing. *Procedia CIRP*, *81*, 405-410.

[27] Chida, T., Kaihara, T., Fujii, N., Kokuryo, D., & Shiho, Y. (2020). Computational Study on Strategyproofness of Resource Matching in Crowdsourced Manufacturing. *International Journal of Automation Technology*, *14*(5), 734-743.

[28] Chopra, S., Meindl, P., & Kalra, D. V. (2007). *Supply Chain Management by Pearson*. Pearson Education India.

[29] Sodhi, M. S., & Chopra, S. (2004). Managing risk to avoid supply-chain breakdown. *MIT Sloan management review*, *46*(1), 53-61.

[30] Coleman, J. S. (1994). Foundations of social theory. *Harvard university press*.

[31] Craighead, C. W., Blackhurst, J., Rungtusanatham, M. J., & Handfield, R. B. (2007). The severity of supply chain disruptions: design characteristics and mitigation capabilities. *Decision sciences*, *38*(1), 131-156.

[32] Cummings, L. L., & Bromiley, P. (1996). The organizational trust inventory (OTI). *Trust in organizations: Frontiers of theory and research*, *302*(330), 39-52.

[33] Crotts, J. C., & Turner, G. B. (1999). Determinants of intra-firm trust in buyer-seller relationships in the international travel trade. *International journal of contemporary hospitality management*.

[34] Cui, J., Ren, L., Mai, J., Zheng, P., & Zhang, L. (2022). 3D printing in the context of cloud manufacturing. *Robotics and Computer-Integrated Manufacturing*, *74*, 102256.

[35] D. Roy, M. & Sarker, B. R. (2021). Optimizing a supply chain problem with nonlinear penalty costs for early and late delivery under generalized lead time distribution. *Computers & Industrial Engineering*, *160*, 107536.

[36] Das, T. K., & Teng, B. S. (1998). Between trust and control: Developing confidence in partner cooperation in alliances. *Academy of management review*, *23*(3), 491-512.

[37] Datta, P. (2017). Supply network resilience: a systematic literature review and future research. *The International Journal of Logistics Management*, *28(4),* 1387–1424.

[38] Engelhardt-Nowitzki, C., Nowitzki, O., & Zsifkovits, H. (Eds.). (2010). *Supply Chain Network Management: Gestaltungskonzepte und Stand der praktischen Anwendung*. Springer-Verlag.

[39] Faria, J. A., Nunes, E., & Matos, M. A. (2010). Cost and quality of service analysis of production systems based on the cumulative downtime. *International journal of production research*, *48*(6), 1653-1684.

[40] Fictiv Global Manufacturing Network (https://www.fictiv.com). Accessed on 28 February 2023.

[41] Fisher, M. L., Hammond, J. H., Obermeyer, W. R., & Raman, A. (1994). Making supply meet demand in an uncertain world. *Harvard business review*, *72*, 83-83.

[42] Fisher, M. L. (1997). What is the right supply chain for your product? *Harvard business review*, *75*, 105-117.

[43] Frame, P. (1992). Saturn to fine suppliers $500/minute for delays. *Automotive News*, *21*(12), 36.

[44] Franke, J., Stockheim, T., & König, W. (2005). The impact of reputation on supply chains. An analysis of permanent and discounted reputation. *Information Systems and E-Business Management*, *3*, 323-341.

[45] Furtado, P., & Frayret, J. M. (2015). Proposal sustainability assessment of resource sharing in intermodal freight transport with agent-based simulation. *Ifac-papersonline*, *48*(3), 436-441.

[46] Galaskiewicz, J. (2011). Studying supply chains from a social network perspective. *Journal of Supply Chain Management*, *47*(1), 4-8.

[47] Ganesan, S., & Hess, R. (1997). Dimensions and levels of trust: implications for commitment to a relationship. *Marketing letters*, *8*, 439-448.

[48] Gödri, I. (2022). Improving Delivery Performance in High-Mix Low-Volume Manufacturing by Model-Based and Data-Driven Methods. *Applied Sciences*, *12*(11), 5618.

[49] Grout, J. R. (1997). A model of incentive contracts for just-in-time delivery. *European Journal of Operational Research*, *96*(1), 139-147.

[50] Guiffrida, A. L., & Nagi, R. (2006). Cost characterizations of supply chain delivery performance. *International Journal of Production Economics*, *102*(1), 22-36.

[51] Guiffrida, A. L., & Jaber, M. Y. (2008). Managerial and economic impacts of reducing delivery variance in the supply chain. *Applied mathematical modelling*, *32*(10), 2149-2161.

[52] Gulati, R. (1995). Does familiarity breed trust? The implications of repeated ties for contractual choice in alliances. *Academy of management journal*, *38*(1), 85-112.

[53] Gyulai, D., Pfeiffer, A., Nick, G., Gallina, V., Sihn, W., & Monostori, L. (2018). Lead time prediction in a flow-shop environment with analytical and machine learning approaches. *IFAC-PapersOnLine*, *51*(11), 1029-1034.

[54] Hayes, R. H. (1979). Link manufacturing process and product life cycle. *Harvard business review*, 133-140.

[55] Harlamova, M., & Kirikova, M. (2018). Towards the trust model for Industry 4.0. *Databases and Information Systems: 13th International Baltic Conference, DB&IS 2018, Trakai, Lithuania, July 1-4, 2018, Proceedings 13*, 43-57.

[56] Heinen, E., & Heinen, E. (1976). Die Produktionsfunktion als produktions-und kostentheoretisches Grundmodell. *Produktions-und Kostentheorie*, 221-256.

[57] Hippmann, S., Klingner, R., & Leis, M. (2019). Digitization–Areas of application and research objectives. *Digital transformation*, 9-17.

[58] Hofmann, E., & Rüsch, M. (2017). Computers in Industry Industry 4. 0 and the current status as well as future prospects on logistics. *Comput. Ind*, *89*, 23-34.

[59] Hohenstein, N. O., Feisel, E., Hartmann, E., & Giunipero, L. (2015). Research on the phenomenon of supply chain resilience: a systematic review and paths for further investigation. *International Journal of Physical Distribution & Logistics Management*, *45*(1/2), 90-117.

[60] Görg, H., Mösle, S., Petersen, T., Egger, H., Kilic, K., Marin, D., and others & Bogaschewsky, R. (2020). Neustart der Industrie unter dem Einfluss von Covid-19: Wie bereit ist die globale Lieferkette?. *ifo Schnelldienst*, *73*(05), 3-34.

[61] Hoppe, M., & Christ, A. (2014). The Transformation of Transportation: Which Borders Will We Have to Cross in the Future?. *Global Studies Journal*, *7*(3).

[62] Houser, D., & Wooders, J. (2006). Reputation in auctions: Theory, and evidence from eBay. *Journal of Economics & Management Strategy*, *15*(2), 353-369.

[63] Hou, Y., Wang, X., Wu, Y. J., & He, P. (2018). How does the trust affect the topology of supply chain network and its resilience? An agent-based approach. *Transportation Research Part E: Logistics and Transportation Review*, *116*, 229-241.

[64] Huhns, M. N., & Singh, M. P. (1998). Agents and multiagent systems: Themes, approaches, and challenges. *Readings in agents*, 1-23.

[65] Huurne, T. M., Ronteltap, A., Corten, R., & Buskens, V. (2017). Antecedents of trust in the sharing economy: A systematic review. *Journal of Consumer Behaviour*, *16*(6), 485-498.

[66] International Electrotechnical Commission (IEC) (2015). Factory of the Future. *IEC Whitepaper*

[67] ISO (2018). BSI Standards Publication Automation systems and integration — Interoperability of capability units for manufacturing application solutions.

[68] Ivanov, D., & Dolgui, A. (2021). OR-methods for coping with the ripple effect in supply chains during COVID-19 pandemic: Managerial insights and research implications. *International Journal of Production Economics*, *232*, 107921.

[69] Jabbarzadeh, A., Fahimnia, B., & Sabouhi, F. (2018). Resilient and sustainable supply chain design: sustainability analysis under disruption risks. *International Journal of Production Research*, *56*(17), 5945-5968.

[70] Jennings, N. R., Sycara, K., & Wooldridge, M. (1998). A roadmap of agent research and development. *Autonomous agents and multi-agent systems*, *1*, 7-38.

[71] Ji, M., Fang, J., Zhang, W., Liao, L., Cheng, T. C. E., & Tan, Y. (2018). Logistics scheduling to minimize the sum of total weighted inventory cost and transport cost. *Computers & industrial engineering*, *120*, 206-215.

[72] Josang, A., (2007). Trust and reputation systems. *Foundations of security analysis and design IV*, 209–245.

[73] Jøsang, A., & Golbeck, J. (2009). Challenges for robust trust and reputation systems. *Proceedings of the 5th International Workshop on Security and Trust Management (SMT 2009), Saint Malo, France, 5(9)*

[74] Jung, J. Y. (2002). Manufacturing cost estimation for machined parts based on manufacturing features. *Journal of intelligent manufacturing*, *13*, 227-238.

[75] Kádár, B., Egri, P., Pedone, G., & Chida, T. (2018). Smart, simulation-based resource sharing in federated production networks. *CIRP Annals*, *67*(1), 503-506.

[76] Kaihara, T., Nishino, N., Ueda, K., Tseng, M., Váncza, J., Schönsleben, P., ... & Takenaka, T. (2018). Value creation in production: Reconsideration from interdisciplinary approaches. *CIRP Annals*, *67*(2), 791-813.

[77] Kaihara, T., Katsumura, Y., Suginishi, Y., & Kadar, B. (2017). Simulation model study for manufacturing effectiveness evaluation in crowdsourced manufacturing. *CIRP Annals*, *66*(1), 445-448.

[78] Kamalahmadi, M., Shekarian, M., & Mellat Parast, M. (2022). The impact of flexibility and redundancy on improving supply chain resilience to disruptions. *International Journal of Production Research*, *60*(6), 1992-2020.

[79] Kleemann, F. C., & Frühbeis, R. (2021). Resiliente Lieferketten in der VUCA-Welt. *Supply Chain Management für Corona, Brexit & Co. Wiesbaden: Springer Gabler.*

[80] Koutrouli, E., & Tsalgatidou, A. (2012). Taxonomy of attacks and defense mechanisms in P2P reputation systems—Lessons for reputation system designers. *Computer Science Review*, *6*(2-3), 47-70.

[81] Kouvelis, P. (2022). Paradoxes and mysteries in virus-infected supply chains: Hidden bottlenecks, changing consumer behaviors, and other non-usual suspects. *Business Horizons*, *65*(4), 469-479.

[82] Krishnan, R., Martin, X., & Noorderhaven, N. G. (2006). When does trust matter to alliance performance?. *Academy of Management journal*, *49*(5), 894-917.

[83] Kumar, G., Meena, P., & Difrancesco, R. M. (2021). How do collaborative culture and capability improve sustainability?. *Journal of cleaner production*, *291*, 125824.

[84] Kühn, M. (2019). Krisenerfahrungen und Erfahrungskrisen von Führungskräften: Perspektiven und Potenziale einer strukturhermeneutischen Betrachtung. *Springer-Verlag.*

[85] Laeequddin, M., Sahay, B. S., Sahay, V., & Waheed, K. A. (2010). Measuring trust in supply chain partners' relationships. *Measuring Business Excellence*, *14*(3), 53-69.

[86] Lanza, G., Ferdows, K., Kara, S., Mourtzis, D., Schuh, G., Váncza, J., ... & Wiendahl, H. P. (2019). Global production networks: Design and operation. *CIRP annals*, *68*(2), 823-841.

[87] Laval, V. (2015). Impact of recent megatrends on the controlling role model. *Procedia Economics and Finance*, *27*, 54-63.

[88] Lee, Y., Lee, K. M., & Lee, S. H. (2020). Blockchain-based reputation management for custom manufacturing service in the peer-to-peer networking environment. *Peer-to-Peer Networking and Applications*, *13*, 671-683.

[89] Leitao, P., Barbosa, J., Papadopoulou, M. E. C., & Venieris, I. S. (2015, March). Standardization in cyber-physical systems: The ARUM case. *Proceedings of the 2015 IEEE International Conference on Industrial Technology (ICIT)*, 2988-2993.

[90] Liu, Y., Wang, L., Wang, X. V., Xu, X., & Zhang, L. (2019). Scheduling in cloud manufacturing: state-of-the-art and research challenges. *International Journal of Production Research*, *57*(15-16), 4854-4879.

[91] Lu, Y., Xu, X., & Xu, J. (2014). Development of a hybrid manufacturing cloud. *Journal of manufacturing systems*, *33*(4), 551-566.

[92] Lu, G., Lu, J., Yao, S., & Yip, Y. J. (2009). A review on computational trust models for multi-agent systems. *The open information science journal*, *2*, 18-25.

[93] Mayer, R. C., Davis, J. H., & Schoorman, F. D. (1995). An integrative model of organizational trust. *Academy of management review*, *20*(3), 709-734.

[94] Melo, S., Macedo, J., & Baptista, P. (2019). Capacity-sharing in logistics solutions: A new pathway towards sustainability. *Transport Policy*, *73*, 143-151.

[95] MFG Custom Manufacturing Marketplace (https://www.mfg.com) Accessed on 28 February 2023.

[96] Monostori, L., Váncza, J., & Kumara, S. R. (2006). Agent-based systems for manufacturing. *CIRP annals*, *55*(2), 697-720.

[97] Moorman, C., Zaltman, G., & Deshpande, R. (1992). Relationships between providers and users of market research: The dynamics of trust within and between organizations. *Journal of marketing research*, *29*(3), 314-328.

[98] Murphy, P. E., & Gundlach, G. T. (1997). A typology of trust in business. *Proceedings of the 1997 Special Conference on New and Evolving Paradigms: The Emerging Future of Marketing*, 596-598.

[99] Newman, S. T., Nassehi, A., Xu, X. W., Rosso Jr, R. S. U., Wang, L., Yusof, Y., ... & Dhokia, V. (2008). Strategic advantages of interoperability for global manufacturing using CNC technology. *Robotics and Computer-Integrated Manufacturing*, *24*(6), 699-708.

[100] Niazi, A., Dai, J. S., Balabani, S., & Seneviratne, L. (2006). Product cost estimation: Technique classification and methodology review. *Journal of Manufacturing Science and Engineering, 128(2),* 563–575.

[101] Nick, G., Szaller, Á., Viharos, Zs. J., Várgedő, T., Bányai, D., Fries, C., Fechter, M. (2020) Intelligent production of the future - first results of a survey. *Proceedings of the IMEKO TC 10 and EUROLAB Virtual Conference "Global Trends in Testing, Diagnostics and Inspection for 2030*, 402–407.

[102] Nwana, H. S. (1996). Software agents: An overview. *The knowledge engineering review*, *11*(3), 205-244.

[103] Ong, N. S. (1995). Manufacturing cost estimation for PCB assembly: An activity-based approach. *International Journal of Production Economics*, *38*(2-3), 159-172.

[104] Palmié, M., Boehm, J., Lekkas, C. K., Parida, V., Wincent, J., & Gassmann, O. (2021). Circular business model implementation: Design choices, orchestration strategies, and transition pathways for resource-sharing solutions. *Journal of cleaner production*, *280*, 124399.

[105] Pedone, G., & Mezgár, I. (2018). Model similarity evidence and interoperability affinity in cloud-ready Industry 4.0 technologies. *Computers in industry*, *100*, 278-286.

[106] Pettersson, A. I., & Segerstedt, A. (2013). Measuring supply chain cost. *International Journal of Production Economics*, *143*(2), 357-363.

[107] Pfeiffer, A., Gyulai, D., Szaller, Á., & Monostori, L. (2018, December). Production log data analysis for reject rate prediction and workload estimation. Proceedings of the *2018 Winter Simulation Conference (WSC)*, 3364-3374

[108] Pinyol, I., & Sabater-Mir, J. (2013). Computational trust and reputation models for open multi-agent systems: a review. *Artificial Intelligence Review*, *40*(1), 1-25.

[109] Ramakurthi, V., Manupati, V. K., Machado, J., Varela, L., & Babu, S. (2022). An innovative approach for resource sharing and scheduling in a sustainable distributed manufacturing system. *Advanced Engineering Informatics*, *52*, 101620.

[110] Riegelsberger, J., Sasse, M. A., & McCarthy, J. D. (2003). The researcher's dilemma: evaluating trust in computer-mediated communication. *International Journal of Human-Computer Studies*, *58*(6), 759-781.

[111] Rivkin, J. W., & Siggelkow, N. (2007). Patterned interactions in complex systems: Implications for exploration. *Management science*, *53*(7), 1068-1085.

[112] Rousseau, D. M., Sitkin, S. B., Burt, R. S., & Camerer, C. (1998). Not so different after all: A cross-discipline view of trust. *Academy of management review*, *23*(3), 393-404.

[113] Russel, R. S., & Taylor III, B. W. (1998). *Operations management: focusing on quality and competitiveness*.

[114] Sabater, J., & Sierra, C. (2005). Review on computational trust and reputation models. *Artificial intelligence review*, *24*, 33-60.

[115] Schallmo, D. R., & Rusnjak, A. (2021). Roadmap zur digitalen Transformation von Geschäftsmodellen. *Digitale Transformation von Geschäftsmodellen: Grundlagen, Instrumente und Best Practices*, 1-36.

[116] Scholz-Reiter, B., Wirth, F., Makuschewitz, T., & Schönlein, M. (2011). Robust capacity allocation in dynamic production networks. *CIRP annals*, *60*(1), 445-448.

[117] Schweitzer, M., Küpper, H. U., Friedl, G., Hofmann, C., & Pedell, B. (2015). *Systeme der Kosten-und Erlösrechnung*. Vahlen.

[118] Sinha, A. K., & Anand, A. (2021). Development of a supply chain configuration model for new product development: a multi-objective solution approach. *Soft Computing*, *25*, 8371-8389.

[119] Spieske, A., & Birkel, H. (2021). Improving supply chain resilience through industry 4.0: A systematic literature review under the impressions of the COVID-19 pandemic. *Computers & Industrial Engineering*, *158*, 107452.

[120] *Die folgen der Corona-Pandemie in 10 zahlen: Pressemitteilung Nr. N 023 vom 31* (https://www.destatis.de/DE/Presse/Pressemitteilungen/2021/03/PD21_N023_p001.html). Accessed on 28 February 2023.

[121] Supply Chain Council. (2008). Supply chain operations reference model. *Overview of SCOR version*, *5*(0).

[122] Suri, R. (2020). *Quick response manufacturing: a companywide approach to reducing lead times*. Productivity Press.

[123] Swiss Virtuellefabrik. (https://www.virtuellefabrik.ch). Accessed on 28 February 2023.

[124] Szaller, Á., & Kádár, B. (2021). Platform and direct exchange-based mechanisms for resource sharing in distributed manufacturing: a comparison. *CIRP Annals*, *70*(1), 407-410.

[125] Szaller, Á., Bozóki, S., Csató, L., Váncza, J., Egri, P., Huszárik, Á., Szádoczki, Zs. (2020) A survey of trust and reputation systems. *SZTAKI internal technical report*

[126] Szaller, Á., Egri, P., & Kádár, B. (2020). Trust-based resource sharing mechanism in distributed manufacturing. *International Journal of Computer Integrated Manufacturing*, *33*(1), 1-21.

[127] Szaller, Á., Pedone, G., Egri, P., Szalóki, Á., & Nick, G. (2020). A mutualistic framework for sustainable capacity sharing in manufacturing. *Procedia CIRP*, *93*, 938-943.

[128] Tao, J., Shao, L., Guan, Z., Ho, W., & Talluri, S. (2020). Incorporating risk aversion and fairness considerations into procurement and distribution decisions in a supply chain. *International Journal of Production Research*, *58*(7), 1950-1967.

[129] Trkman, P., & McCormack, K. (2009). Supply chain risk in turbulent environments—A conceptual model for managing supply chain network risk. *International Journal of Production Economics*, *119*(2), 247-258.

[130] Ustundag, A., & Tanyas, M. (2009). The impacts of radio frequency identification (RFID) technology on supply chain costs. *Transportation Research Part E: Logistics and Transportation Review*, *45*(1), 29-38.

[131] Váncza, J., & Márkus, A. (2000). An agent model for incentive-based production scheduling. *Computers in Industry*, *43*(2), 173-187.

[132] VDI 3633 Blatt 1: 1993-12 Simulation of systems in materials handling; logistics and production (https://www.vdi.de/richtlinien/details/vdi-3633-blatt-1-simulation-of-systems-in-materials-handling-logistics-and-production-fundamentals). Accessed on 28 February 2023.

[133] Vogel-Heuser, B., Bauernhansl, T., & Ten Hompel, M. (2017). Handbuch Industrie 4.0 Bd. 4. *Allgemeine Grundlagen*, *2*.

[134] Wang, E. K., Liang, Z., Chen, C. M., Kumari, S., & Khan, M. K. (2020). PoRX: A reputation incentive scheme for blockchain consensus of IIoT. *Future Generation Computer Systems*, *102*, 140-151.

[135] Whicker, L., Bernon, M., Templar, S., & Mena, C. (2009). Understanding the relationships between time and cost to improve supply chain performance. *International Journal of Production Economics*, *121*(2), 641-650.

[136] Windt, K., Böse, F., & Philipp, T. (2008). Autonomy in production logistics: Identification, characterisation and application. *Robotics and Computer-Integrated Manufacturing*, *24*(4), 572-578.

[137] Xometry Manufacturing Partner Network (https://www.xometry.com). Accessed on 28 February 2023.

[138] Wang, Y., Zhang, J., Guan, X., Xu, M., Wang, Z., & Wang, H. (2021). Collaborative multiple centers fresh logistics distribution network optimization with resource sharing and temperature control constraints. *Expert Systems with Applications*, *165*, 113838.

[139] Wang, Y., Assogba, K., Fan, J., Xu, M., Liu, Y., & Wang, H. (2019). Multi-depot green vehicle routing problem with shared transportation resource: Integration of time-dependent speed and piecewise penalty cost. *Journal of Cleaner Production*, *232*, 12-29.

[140] Wei, Y., & Egbelu, P. J. (2000). A framework for estimating manufacturing cost from geometric design data. *International Journal of Computer Integrated Manufacturing*, *13*(1), 50-63.

[141] Yan, K., Cheng, Y., & Tao, F. (2016). A trust evaluation model towards cloud manufacturing. *The International Journal of Advanced Manufacturing Technology*, *84*, 133-146.

[142] Yang, X., Wang, S., Yang, B., Ma, C., & Kang, L. (2019). A service satisfaction-based trust evaluation model for cloud manufacturing. *International Journal of Computer Integrated Manufacturing*, *32*(6), 533-545.

[143] Ye, Q., Xu, M., Kiang, M., Wu, W., & Sun, F. (2013). In-depth analysis of the seller reputation and price premium relationship: a comparison between eBay US and Taobao China. *Journal of Electronic Commerce Research*, *14*(1), 1.

[144] Zaheer, A., McEvily, B., & Perrone, V. (1998). Does trust matter? Exploring the effects of interorganizational and interpersonal trust on performance. *Organization science*, *9*(2), 141-159.

[145] Zhang, J., Zhu, M., Yu, M., Wang, K., & Zhang, R. (2022). Evaluation of multiple air pollutant emission reductions in northern China district via resource sharing. *Journal of Cleaner Production*, *379*, 134743.

[146] Zhang, Y., Tang, D., Zhu, H., Li, S., & Nie, Q. (2021). A flexible configuration method of distributed manufacturing resources in the context of social manufacturing. *Computers in Industry*, *132*, 103511.

[147] Zhou, Z., Pei, J., Liu, X., Fu, H., & Pardalos, P. M. (2021). Effects of resource occupation and decision authority decentralisation on performance of the IoT-based virtual enterprise in central China. *International Journal of Production Research*, *59*(24), 7357-7373.

List of Figures

List of Tables

Appendix

Classes of the agent-based simulation model

Equipment class

An instance of the **Equipment** class represents one resource type in a specific company. It contains the following fields:

- *name* of the resource type
- *amount* of the resource type that the specific company has (e.g. 3 CNC machines)
- *available* amount in the given time point
- list of *planned requests* for the specific resource type (list of Request instances already undertaken)
- *price* of one resource unit (used when making offers)
- *quality* of the resource type

Offer class

An **Offer** contains the following information:

- offer sender *company*
- *amount* of offered resources
- *resource type*
- *start* and *end time* of the offer
- offer *ID,* used to help finding the specific offer when deleting it after its end time reached
- *restoreOffer* event that is called when the end time is reached; it deletes the offer from the Platform offer database

OfferCombination class

An **OfferCombination** is a combination of offers. When receiving a new request, the Platform checks whether it can be fulfilled with different combinations of offers in its database (according to Figure 15), and first generates all the possible combinations of offers. In the next step, it checks if these combinations fulfil the requirements of the request or not. If yes, the appropriate **OfferCombinations** are placed to a **MatchingOffers** instance, detailed in the next paragraph.

An instance of an **OfferCombination** contains

- the list of *offers*
- the *fitness* of offers that is calculated when it is sent to a company; this is compared to find the winner

- *weights* of the offers that are also used in decision making: these weights are calculated based on the ratio of offered resource loads (amount multiplied with time interval).

MatchingOffers class

Contains all the **OfferCombinations** that match a specific request in an ArrayList data form, and the request also – this “package” is sent to the requester company for decision making.

Request class

A **Request** is generated in the following cases:

- when a company receives an order, one **Request** is generated for each job (or two of them are generated, one for the insourced and another for the outsourced job part)
- when an **Offer** is sent to the Platform, and a virtual **Request** is inserted into the company’s production plan, in which the amount of required resources is equal to the offered amount.

A **Request** contains the following information:

- name request sender *company*
- requested *resource type*
- requested resource *amount*
- request *generation time*
- *start* and *end* of the requested interval
- **CapacitySeize** and **CapacityRelease** events (in order to be accessible through the request instance)
- *ID* in case of virtual requests generated due to offer sending, in order to delete it from the company’s production plan in an easier way if the offer is not matched

Reputation class

Reputation class is used to store reputation values for each company. The two class fields included are the *value* and the *number of interactions* based on which the aggregated value is calculated in the **rep_calc** function of the Platform agent.

The following classes are defined to help to model the messages sent between the Platform and the companies.

ReactToOffers class

Sent by the requester company to the Platform about a list of **OfferCombinations** the Platform sent to the requester company earlier. The following information is included:

- *name* of requester,
- winner **OfferCombination,**
- rejected **OfferCombination(s),**
- the specific **Request**, to delete it from the Platform request database after matching.

NotifyRequester class

The aim of this message is to notify the requester company about the validity of the offers it has accepted in connection with a specific request. Contains a boolean value (true if all the offers are valid in the winner **OfferCombination**, false otherwise) and the specific **Request**.

NotifyOfferor class

The aim of this message is to notify the offeror company about the acceptance of a specific **Offer** it sent to the Platform earlier.

Notations of the trust model

Table 15. List of notations in the financial model

Notation	Meaning
$r_{t_r,j}^{A,B}$	rating given in time point t_r about a specific interaction between lead company A and resource offeror company B, about job *j*
δ_j	time difference between the delivery deadline and the real delivery time in case of job j
L_j	length of job j in time
γ	the penalty factor applied on federation level to penalize inaccurate deliveries to a higher extent
μ_j	quality factor to rate the quality of the resource offeror's work about job *j*
$w(t_r, t)$	weight that is assigned to a rating given in time point t_r in order to calculate the cumulative rating in time point *t*
θ	decay factor used to affect the shape of the $w(t_r, t)$function
$\varphi^{A,B}(t)$	cumulative trustfulness calculated by lead company A about resource offeror company B in time point *t*

Notations of the financial model

Table 16. List of notations in the financial model

Notation	Meaning
All cost types	
A	resource requesting (lead) agent
B	resource offering agent
subscript j	job index

superscript o	order index
superscript C	index of customer outside the federation
superscript x	the index of the number of times a company entered the federation
α^o	number of products in order o
α_j^B	number of products in the part of job j that was outsourced to company B
Manufacturing costs	
$C^B_{manuf_x}$	total manufacturing cost for company B for the time interval it was a member of the federation for the xth time
c^B_{sust}	facility sustaining cost for company B, per time unit
T_x	length of time interval when company B is a member of the federation for the xth time, in time units
$c^B_{manuf_j}$	total manufacturing cost of the part job j that was outsourced to company B
$c^B_{material_j}$	material cost for one product in job j for company B
$c^B_{labour_j}$	labour cost per time unit in job j, for company B
$c^B_{machine_j}$	machine cost per time unit in job j, for company B
t_j^B	time interval required to complete the part of job j that was outsourced to company B
Inventory costs	
$c^B_{WIP,j}$	WIP inventory cost for the part of job j that was outsourced to company B
$c^B_{inv_j}$	inventory cost of agent B for one product in job j, for one time unit
t_j^B	manufacturing time of the part of job j that was outsourced to company B
$c^A_{inv_{j,B}}$	total inventory cost that incurs due to early completion for lead company A for the specific part of job j that was outsourced to company B
Penalty cost	
$c^B_{pen_j}$	penalty cost for resource offering company B in case of job j
$c^A_{inv_j}$	inventory cost of agent A for one product in job j, for one time unit
$c^A_{qual_j}$	quality reduction cost of agent A for one product in job j, for one time unit
$c^A_{loss_j}$	sales opportunity loss for agent A for one product in job j, for one time unit
c^A_{fail}	fixed failed delivery factor for agent A
P^o	profit generated by selling order o
${t_{sa_j}}^B$	start of *delivery acceptance* interval of the part of job j that was outsourced to agent B
${t_{ea_j}}^B$	end of *delivery acceptance* interval of the part of job j that was outsourced to agent B
${t_{so_j}}^B$	start of *on-time* interval of the part of job j that was outsourced to agent B
${t_{eo_j}}^B$	end of *on-time* interval of the part of job j that was outsourced to agent B
${t_{d_j}}^B$	delivery time of the part of job j that was outsourced to B
Transportation cost	
$c^{AC,o}_{transport}$	transportation cost of order o between company A and customer C
C^o	customer (outside the federation) of order o
d^{AC}	distance between agent A and customer C on the road
${c_{transport}}^o$	cost factor for specific transport type in case of order o, per product
${c_{shipment}}^A$	fixed shipment sending cost for lead agent A
Transportation consolidation cost	
$c^{B,j}_{tr.cons}$	transportation consolidation cost for resource offeror company B, in case of job j
$c^{A,j}_{tr.cons}$	transportation consolidation cost for lead company A, in case of job j
d^{AB}	distance between agent A and B on the road
$c_{transport_j}$	cost factor for specific transport type in case of a specific job

$c_{arrival}{}^{A}$	arrival (administration) cost for company A
$c_{shipment}{}^{B}$	shipment sending (administration) cost for company B
Management cost	
$c^{A}_{man_x}$	total management cost for company A for the time interval it was a member of the federation for the xth time
T_x	length of time interval when company A spent as a member of the federation federation for the xth time, in time units
c_{entr}	one-time entrance fee of the platform
c_{part}	regular participation fee of the federation, per time unit
$c_{offer,a}$	administrative cost of sending an offer (paid to the Platform)
$c_{offer,p}$	resource planning cost of sending an offer to the Platform
$n_{offer}{}^{A}{}_{x}$	number of offers sent to the platform by company A during the time it was a member of the federation for the xth time
$c_{request,a}$	administrative cost of sending a request (paid to the Platform)
$c_{request,p}$	resource planning cost of sending a request to the Platform
$n_{request}{}^{A}{}_{x}$	number of requests sent to the platform by company A during the time it was a member of the federation for the xth time
$c_{contract}$	administrative cost of contracting in case of a match
$n_{contract}{}^{A}{}_{x}$	number of contracts signed by company A during the time it was a member of the federation for the xth time
Order consolidation cost	
$c^{A}_{ord.cons}$	total order consolidation cost for company A
$c^{A}_{ord.cons_j}$	order consolidation cost of job j
Sales income, profit	
$i_{sales_j}{}^{B}$	sales income for company B for completing (a part of) job j
$c_{manuf_j}{}^{B}$, $c_{manuf_j}{}^{A}$	manufacturing cost of (a part of) job j for resource offeror company B manufacturing cost of (a part of) job j for lead company A
p^{B}, p^{A}	profit margin of resource offeror company B and lead company A
$i_{sales}{}^{A,o}$	additional income for completing order o
i^{o}	additional income per product for order completion
P^{B}_{j}	profit generated by company B from job j
$P^{A,o}$	profit generated by company A from order o
Penalty income	
$i_{pen}{}^{A,B}{}_{j}$	penalty income for lead company A in connection with the job part that was outsourced to company B
$c_{pen_j}{}^{B}$	penalty cost for resource offering agent B in case of job j
Incomes and costs of the Platform	
$i^{Platform}$	total income of the Platform
k	federation member index
K	number of members in the federation
$c^{Platform}$	total costs of the Platform
c_{mar}	marketing costs of the Platform
c_{op}	operating costs of the Platform
c_{maint}	maintenance costs of the Platform