7-2-2021

Reshaping the Landscape of the Future: Software-Defined Manufacturing

Lei Xu
The University of Texas Rio Grande Valley, lei.xu@utrgv.edu

Lin Chen

Zhimin Gao

Hiram Moya
The University of Texas Rio Grande Valley

Weidong Shi

Recommended Citation

Xu, Lei, Lin Chen, Zhimin Gao, Hiram Moya, and Weidong Shi. 2021. "Reshaping the Landscape of the Future: Software-Defined Manufacturing." Computer 54 (7): 27–36. https://doi.org/10.1109/MC.2021.3074851.

Reshaping the Landscape of Future Manufacturing: Software Defined Manufacturing

Lei Xu
University of Texas Rio Grande Valley

Lin Chen
Texas Tech University

Zhimin Gao
Auburn University at Montgomery

Hiram Moya
University of Texas Rio Grande Valley

Weidong Shi
University of Houston

***Abstract*—The prosperity of the manufacturing sector is critical for the overall competitiveness of a nation/region. Most of the existing efforts focus on the modernization and digitization of each individual factory, which can only bring isolated performance improvements and cannot optimize the ecosystem as a whole. A novel idea to overcome these limitations and keep improving the performance of the manufacturing sector is to enable horizontal scaling by breaking the silos and allowing more flexible resource sharing. Following this insight, we describe the concept of software defined manufacturing, which divides the manufacturing ecosystem into software definition layer (SDL) and physical manufacturing layer (PML). PML consists of a large number of unified manufacturing equipment and SDL takes care of all other jobs. Software defined manufacturing allows better resource sharing and collaboration, and has the potential to transform the existing manufacturing sector.**

Index Terms: Industrial control, Manufacturing, Security and Protection

INTRODUCTION

It is the consensus that a robust manufacturing sector is essential to maintain the overall competitiveness of a nation. Major countries are investing heavily to modernize their manufacturing sector and gain an edge over the competition. Stock and Seliger proposed the concept of Industry 4.0 back in 2011 [1], which received extensive studies and was the first national strategy on next-generation manufacturing development. China's *Made in China 2025* plan is another significant effort in promoting future manufacturing. *Innovation 25* is Japan's package program to empower the country's economy through innovation and an open attitude, where improving Japan's competitiveness in manufacturing is an important part of the program.

Although these efforts have different names, they share the same philosophy of increasing the investment of technology to improve the overall competitiveness of their manufacturing sector. One important investment direction of all these strategies is to develop new information and communications technologies (ICT) and to accelerate their integration with traditional manufacturing infrastructures, such as industrial Internet of Things (IoT), artificial intelligence (AI), and virtual reality/augmented reality (VR/AR) for the industry. Most of these efforts aim at connecting the physical manufacturing infrastructure more closely with ICT and build closed and proprietary manufacturing infrastructures. The expectation is that the integration greatly improves manufacturing efficiency and reduces labor intensity or utilization at the same time. People are then freed from repetitive tasks and more time can be spent on creative activity. Based on the advancement in factory digitization, the concept of agile manufacturing was proposed [2], which focuses on enabling an organization to quickly respond to customer needs and market changes.

Originally, the integration of ICT and manufacturing infrastructure focused on each individual factory, where resource sharing is not a factor for consideration. This brings at least two limitations: (i) The information system invested in each individual factory is not fully utilized. The information system capability (e.g., computation and communication) is designed to handle the peak demand of a factory, and at other times the resources are wasted. (ii) Each factory is a silo. While a factory is equipped with an advanced information system and highly automated with precision operation capability, it does not have complete support for collaboration with other factories, especially for those who belong to other entities. As collaboration plays a more important role in modern manufacturing, this in turn limits the potential of each individual factory.

The isolation also brings other issues. For instance, modern manufacturing usually involves a global supply chain. That is, a single link on the chain without the most advanced configuration will limit the whole process. With factories becoming more sophisticated, it becomes harder for newcomers to jump on the competitive stage and may lead to monopolies, which in turn limits innovation.

The development of cloud computing sheds light to mitigate the above limitation, and the concept of *cloud manufacturing* was proposed [3]. In the paradigm of cloud manufacturing, instead of maintaining dedicated computation infrastructure for each factory, most of the computation and storage jobs are moved to the cloud computing infrastructure that is shared by multiple factories owned by different entities. Cloud manufacturing enjoys all the benefits of cloud computing from the perspective of information processing, for example, high flexibility, availability, scalability, and the pay-as-you-go service manner; and partially addresses the issue of waste of information processing resources.

As the deployment of the 5G mobile network increases, the resource-sharing between different factories is further improved. The 5G infrastructure enables various virtual technologies to support *network slicing*, which allows the network operator to build a virtual dedicated network with different features for a certain application scenario. It also offers a good option to replace dedicated and proprietary communication systems integrated with factories.

Besides leveraging cloud computing and 5G to improve the level of resource sharing, crowdsourcing [4] also becomes more popular in manufacturing. To accelerate new product development and to introduce innovative enhancements to existing ranges, there is an urgent need to utilize and share organizational and employee knowledge. Crowdsourcing offers a collaborative idea generation and problem solving mechanism. It allows one to obtain explicit knowledge from a broader community and extract previously unknown tacit knowledge in a less formal manner [5].

From the adoption of these technologies and methodologies, it can be seen that the trend for future manufacturing at least consists of two essential components, resources sharing and collaboration. Resource sharing reduces the idle time of the information system that is integrated with the manufacturing infrastructure, and provides a higher level of flexibility and resilience. At the same time, close collaboration accelerates manu-

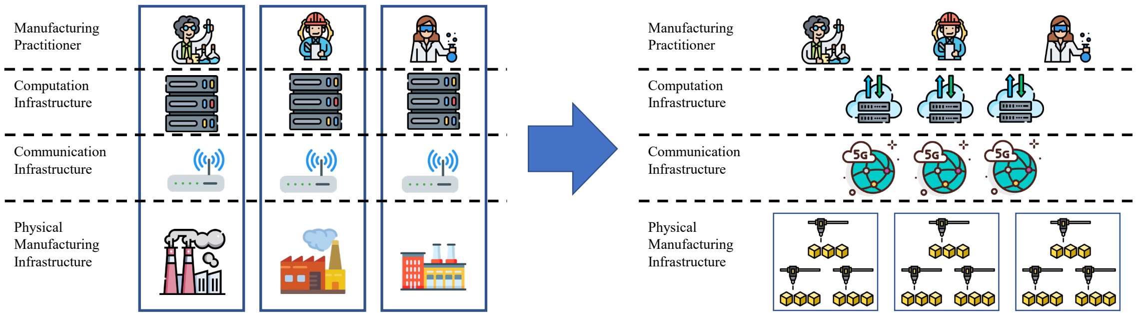

Figure 1. Moving from current manufacturing infrastructure to the future software defined manufacturing infrastructure. Under the framework of software defined manufacturing, both information processing resources and physical manufacturing resources are shared by all practitioners, which not only improve the level of resource utilization, but also enable smoother and easier collaboration.

facturing innovation and R&D.

Along this direction, the landscape of manufacturing will continue to transform to further enhance resource sharing and collaboration. One promising approach is *software defined manufacturing*. Software defined manufacturing learns lessons from the key technologies from ICT, such as software defined network (SDN [6]) and software defined infrastructure (SDI [7]). The overall idea is to convert the infrastructures of certain manufacturing sectors to a unified, shared platform. Practitioners of that sector can schedule their work freely on the platform. The platform then allows practitioners to collaborate easily on all types of tasks to unleash the platform's potential fully.

OVERVIEW OF THE SOFTWARE DEFINED MANUFACTURING

Software defined manufacturing breaks the boundaries between different factories by extending existing ideas such as cloud based manufacturing and 5G, and brings resource sharing and collaboration to a new level, which are critical for future manufacturing.

Architecture of Software Defined Manufacturing

Figure 1 presents the major differences between the current practice of utilizing information technology to modernize manufacturing infrastructure and the future software defined manufacturing infrastructure. As depicted on the left side of Figure 1, current R&D efforts such as Industry 4.0 [1], [8] focus on modernization and digitization of each single manufacturing factory by introducing automated tools like robots and AI components to reduce the working intensity, improve production efficiency, and eliminate potential failures with better prediction. Under this framework, the potential of each factory can be maximized and the productivity of the whole sector scales vertically.

On the right side of Figure 1, instead of splitting the manufacturing sector vertically into silos, software defined manufacturing framework breaks the boundaries and makes it an open, sharable ecosystem. Specifically, software defined manufacturing divides the industry ecosystem horizontally into two layers, the physical manufacturing layer and the software definition layer, which are accessible to and shared by all practitioners.

- *Physical manufacturing layer (`PML`).* The framework requires the physical manufacturing layer to support universal production and be as flexible as possible. The `PML` is composed of a group of sites, and each site has a set of equipment that can be controlled by software for manufacturing. Here the equipment can be a 3D printer or other instruments that have the capability to make a wide range of different items. The `PML` is shared by multiple practitioners. Although it is a challenge that each site can support all manufacturing activities, it is feasible to build such sites for a specific sector to cover the majority of the production activities. The auto industry has adopted a similar strategy by introducing car platforms to increase the percentage of sharable components [9]. The industry has been pursuing *flexible manufacturing* for a long time, which aims at the ability to deal with mixed parts, allow variation in parts assembly, and support production volume and design changes. Most of these features are aligned with the goals of PML. Leading equipment manufacturers like ABB and Leidos have developed a variety of technologies to support flexible manufacturing that can be leveraged for the construction of PML. At the early stage of software defined manufacturing, PML can also include more than one type of site to support multiple sectors. But the sharing nature of these sites does not change.

- *Software definition layer (SDL).* The SDL is responsible for everything except the actual manufacturing, including design, development, simulation, and control of the whole information infrastructure of the manufacturing process (e.g., sending instructions and receiving feedback from PML sites and equipment). SDL can be further decomposed into two sub-layers, the computation layer and the communication layer. The computation layer manages a variety of valuable digital assets for a practitioner such as designs and technological processes, which determine a practitioner's capability and level of efficiency to make products. The communication layer is a bridge that connects the cyber world and the physical world, which supports reliable connections between the computation infrastructure and the physical manufacturing infrastructure. The communication layer is also configurable and controllable through the computation layer, i.e., a practitioner can decide the features of connections needed for the manufacturing tasks, such as bandwidth and latency. Similar to PML, the SDL infrastructure is shared by multiple practitioners. The difference is that SDL is shared across different sectors. At the same time, a practitioner can store and manage private assets using SDL for their own usage, trading with others, or collaboration with other practitioners.

Components and Usage of Software Defined Manufacturing

There are three types of components in the software defined manufacturing infrastructure.

- Computing node. A large number of computing nodes compromise the computation layer of SDL, which is responsible for computation and storage tasks. There are different ways to organize computing nodes, and a simple way is to have a centralized party to provide and manage all computing nodes, like the current practice in cloud computing.
- Communication channel. Communication channels are also part of SDL and are responsible for providing reliable connections between other nodes in the system. As 5G infrastructure is deployed, the computing nodes and communication channel can converge and be managed by a single operator.
- Manufacturing node. These nodes compromise the PML. A manufacturing node is highly autonomous and can be owned by anyone. After it connects to the framework, it is fully controlled by the SDL for daily operations.

A practitioner p interacts with the software defined manufacturing as follows:

- p logs into its management portal hosted in the cloud, which is part of the software defined manufacturing framework. Through the management portal, p can see the demand and supply information, all its digital assets, the current status of manufacturing nodes, and other services (e.g., scheduling tool) it subscribed to.
- p finalizes the production plan and submits instructions to manufacturing nodes through the software defined manufacturing framework.
- A manufacturing node carries out the production according to received instructions (e.g., technological process and quantity).

A financial settlement protocol is integrated with the above process, and the payment option can be part of the manufacturing node status.

Figure 2 shows a concrete example of software defined manufacturing architecture from a practitioner's perspective. SDL is built and implemented using the cloud-edge computation infrastructure, which includes a limited number of cloud computing data centers with plenty of computation and storage resources and a large number of edge computing data centers that are physically distributed. The cloud (the centralized computing infrastructure in Figure 2) is responsible for interacting with the practitioner and most of the computation and storage intensive manufacturing activities related R&D work. As the physical manufacturing facilities (manufacturing node) are geographically distributed, multiple edge data centers (the distributed computing infrastructure in Figure 2) are utilized to control the operations of manufacturing nodes that are close to them. The communication infrastructure is integrated as part of the cloud-edge computation system.

A practitioner utilizes a cloud data center to finish the R&D work of a new product. The cloud also maintains the current statuses of all manufacturing nodes that are essential for scheduling, including locations, production capacities, planned workloads, and transportation information. When there are multiple types of manufacturing nodes in the system, the type information is also provided. The practitioner utilizes the information to schedule the physical production works. If the product is a sub-component of another product, the practitioner also takes the schedules of collaborators into consideration when determining their own plan.

After the schedule is finalized, instructions are distributed to edge data centers, and the edge data centers will coordinate with manufacturing nodes to start the production process. Even for a single practitioner, the manufacturing process can involve multiple steps. In this case, it is possible several manufacturing nodes are utilized. The end product of one manufacturing node is the raw product of another manufacturing node, and transportation is involved as part of the schedule.

Under the software defined manufacturing framework, `SDL` and `PML` do not belong to any specific entity. Both `SDL` and `PML` are provided virtually to all practitioners, and the resource allocation process is transparent to them. In other words, each practitioner believes is the only one using the system and does not need to be aware of other practitioners in the system. In Figure 2, resources covered by the shaded area are allocated to the practitioner, and from the view of the practitioner, all these resources are used exclusively. But in reality, these resources can also be utilized by another practitioner.

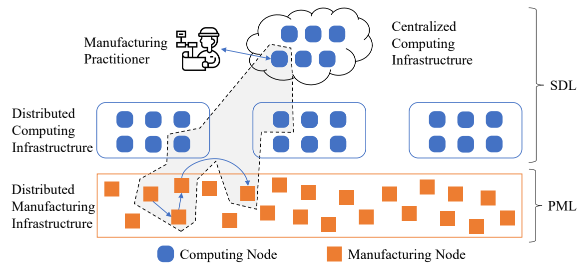

Figure 2. A use case of the software defined manufacturing from a practitioner's perspective.

IMPACTS OF SOFTWARE DEFINED MANUFACTURING

By dividing the manufacturing ecosystem horizontally instead of vertically, the software defined manufacturing framework has the potential to transforms the business model of manufacturing as it breaks the silos and enables a free and fully competitive market. In this section, we summarize the potential impacts of software defined manufacturing when it becomes more mature and is widely adopted in practice.

Accelerating Innovation

Innovation is playing a more important role in modern manufacturing. Within the current manufacturing ecosystem, the threshold to join the manufacturing business is becoming higher and higher, especially for advanced manufacturing that requires sophisticated and expensive equipment. Without access to these facilities, there is little a practitioner can do, e.g., conducting experiments to verify a new design or building a small number of products for the purpose of market feedbacks collection. Established big players will try to further enhance the barriers to maintain their monopoly position. Therefore, it prevents broad participation and indirectly hinders innovation.

The adoption of software defined manufacturing can fully change the game, and help to realize the concept given in social manufacturing [10], [11]. Under the new framework, the whole infrastructure (including both `SDL` and `PML`) does not belong to any specific practitioner and is accessible to the public. Similar to the scenario of cloud computing, one can utilize the resources by renting other than owning, and the cost is charged in the manner of pay-as-you-go. Therefore, one can easily test new ideas at a relatively low cost and increase the production scale easily when the concept is verified.

In summary, the new framework lowers the market access threshold and puts a new practitioner at the same starting point as his/her established competitors. Besides removing one of the most critical barriers for innovation in the manufacturing sector, it also encourages existing practitioners to invest more in R&D as traditional advantages of established players are greatly eliminated.

Improving Supply Chain Resilience and Flexibility

Modern manufacturing usually involves a complex and long supply chain, where most practitioners focus on their own business and compromise a tiny part of the whole system. Under the current framework, this fine-grained division of work has been proved to be successful in maximizing production efficiency. But at the same time, the sophisticated supply chain also brings more uncertainties to the ecosystem. The effect of a failure of a single link in the system can cascade to other practitioners on the same supply chain, and disrupt normal operations. The recent COVID-19 pandemic has shown the fragility of such a complex supply chain. As some of the major economies are locked down, the original supply chains are disrupted and it is hard, if not impossible, to re-build a similar one in another place in a short time period.

With the software defined manufacturing framework, `SDL` can easily re-schedule information processing related works from failed/disconnected computing nodes to others without service disruption. `PML` consists of a large number of unified manufacturing nodes that are flexible to produce different things. Although these manufacturing nodes may have different outputs, they can largely replace each other. Even if a node with higher output fails, multiple production nodes with relatively lower outputs can be used to replace its role with limited impacts on the supply chain. The substitutability of production nodes in `PML` gives a practitioner more flexibility to schedule physical production tasks. At the early stage of software defined manufacturing, there can be several groups of production nodes with different functions. This adds another constrain but will not affect the structure of the new framework.

In practice, this process can even be fully automated by a re-schedule service hosted in `SDL`. The practitioner can provide certain constraints beforehand, and `SDL` monitors the current status of the whole system. When some of the resources involved in current production activity become unavailable, the service runs an algorithm to determine the possible re-scheduling options based on the constraints and availability information. The practitioner can either select one from the options or allow the re-schedule service to do it automatically.

Facilitating and Enhancing Collaboration

The manufacturing sector can benefit significantly from collaboration due to the fine-grained division of works. However, collaboration under the existing paradigm is quite complex as each factory/entity is a closed system, and usually has limited and fixed partners. To initialize a new collaboration relationship, interested practitioners need to create a pair of dedicated interfaces between each other, which usually involves a lengthy administration process and cannot be reused with others.

Software defined manufacturing simplifies the process of collaboration from the following two perspectives: (i) Focusing on collaboration in `SDL`. As manufacturing nodes in `PML` are universal and completely controlled by computing nodes in `SDL`, two practitioners who want to collaborate only need to work in the cyber world to achieve consensus on the protocol for exchanging digital assets (e.g., designs, patents, and drawings) and provisioning adequate privileges to the other party to access their own resources in `SDL`. (ii) Open standards. The interfaces between major components of software defined manufacturing will be standardized, so the work that is done for one collaboration can be easily re-used for other collaborations.

By utilizing the above two features, a market for collaboration can be built as part of `SDL`, where a practitioner can publish all types of demands to seek potential collaborators. Other practitioners who have a solution for the problem can then bid or negotiate on the platform to become a collaborator. This collaboration market is similar to a crowdsourcing platform that fosters collaborations, but is dedicated to the manufacturing sector and closely integrated with the software defined manufacturing framework.

Improving Overall Efficiency

Ideally, a factory should maximize the utilization of its resources and meet all demands simultaneously. Under the current framework, this is very hard to achieve as a factory is built with fixed output capacity, but the demand keeps changing. When the output capacity and demand do not match, there is no way for a single factory to achieve the two goals at the same time. Even if the output capacity and demand match for a single factory, the actual output is also affected by upstream and downstream factories, which rely on the market mechanism to determine their outputs indirectly. Therefore, there are very likely mismatches of demand and supply, which causes underutilization and wastes resources.

There are two reasons for the above problem: lack of resource sharing and lack of information sharing. The software defined manufacturing solves this problem in a systematic way. Manufacturing nodes in `PML` are universal and shared by all practitioners, as long as practitioners have different peak demand times, `SDL` can easily do a load balancing to maximize the utilization of production nodes, which is similar to the scenario of cloud computing where physical servers are shared by multiple tenants with different peak demand times.

`SDL` also allows different practitioners to exchange demand/supply information easily and they can schedule production tasks only when there is a determined demand from others. Therefore, the demand and supply will be better matched and stocking cost will be reduced.

Cost Reduction

The cost of manufacturing can vary significantly for different industries, and we only analyze the major factors in a qualitative manner to show the potential of cost reduction with software defined manufacturing.

The major factors of the cost of conducting business in manufacturing in a traditional way include: (i) Cost of infrastructure building (B); (ii) Cost of producing (P), including material and labor cost; and (iii) Cost of the supply chain (S). While P and S are proportional to the output O, B is relatively independent of O. Therefore, the cost of each unit is $P/O + S/O + B/O$. Under the new framework, a practitioner does not need to invest in his/her own infrastructure but pays to rent necessary resources to create the manufacturing system when needed, which is denoted as R. The practitioner still needs to cover the costs of producing (P') and supply chain (S'). But in this case, R is also proportional to O. In other words, R/O is roughly a constant value under software defined manufacturing framework. For the same output O, the cost of each unit is $P'/O + S'/O + R/O$.

The cost of producing is similar for both cases (i.e., $P = P'$), and the supply chain cost for software defined manufacturing can be cheaper as the practitioner has more flexibility in the optimization of the supply chain (i.e., $S > S'$). The relationship between B and T is more complicated. When the infrastructure is fully utilized (O is maximized), $B/O < R/O$. However, when O is smaller, we have $B/O > R/O$. In summary, when O does not reach the full capacity of the manufacturing infrastructure, $P/O + S/O + B/O > P'/O + S'/O + R/O$.

From the infrastructure owner's perspective, although he/she can only collect a rent fee that is proportional to the output, the facility can be rented to multiple practitioners.

TECHNOLOGY CHALLENGES

While having the potential to revolutionize manufacturing and bring many advantages, there are several technical challenges that need to be addressed.

Construction of Flexible and Universal Manufacturing Node

One of the most important key enablers for software defined manufacturing is the need for a flexible and universal manufacturing node, i.e., a node can produce a wide range of products and easily switch from one product to another.

Without a large number of such nodes, SDL cannot have enough freedom to schedule production tasks to realize benefits such as innovation acceleration and better system efficiency.

One promising direction to implement such a flexible and universal manufacturing node is to utilize 3D printing and additive manufacturing technologies [12], [13]. Recent 3D printing and additive manufacturing technologies can use different materials to make many different products, from mechanical parts to medical tools and devices. Such production technologies can be easily integrated with SDL as they are fully controlled by software. Most existing 3D printing technologies are not scalable enough and too expensive for massive production. To overcome this challenge, it requires both advancement in material and print technology itself. Some works have been done along with these directions [14], [15].

Even if the universal manufacturing node is not mature at this time, the software defined manufacturing is still valuable: (i) The new framework can be applied to more specific sectors so the hardness of building a unified manufacturing node; (ii) Instead of using a single type of manufacturing nodes, the framework can support multiple types of manufacturing nodes. These strategies can help the early adoption of software defined manufacturing.

Incentive Mechanism Design

To maximize the resource utilization, both the SDL and the PML infrastructures are shared by all practitioners. A natural question is: who will build and maintain these infrastructures? Building and running the SDL is relatively simple as it is very similar to the cloud/edge computing infrastructure we have today, and the major vendors (e.g., Amazon, Microsoft, and Google) have established a mature business model. The incentive mechanism that encourages the construction of manufacturing nodes in PML is less clear. To achieve some of the benefits of software defined manufacturing such as high supply chain resilience, it is better to deploy the manufacturing nodes globally, so practitioners have more choices when part of PML fails and disconnects from the system.

However, without a carefully designed incentive mechanism, investors may tend to build the manufacturing nodes into several clusters to reduce the cost of movement of components and attract more practitioners to schedule their jobs to manufacturing nodes that belong to a small number of clusters. This will form positive feedbacks and leads to a further concentration of manufacturing nodes, which is undesirable from the perspective of whole system resilience.

To avoid this from happening, the software defined manufacturing framework needs a compensation system to encourage deployment of manufacturing nodes in less favored locations, e.g., the system can collect a certain percentage of income from manufacturing nodes deployed in popular locations to balance the profits rate.

System Security

The software defined manufacturing provides a disruptive way to organize manufacturing, maximizing the level of resource sharing to improve both flexibility and efficiency. To fully unleash the potential of this new way of organizing the manufacturing process, several security challenges need to be addressed:

Isolation in the multi-owners/users infrastructure. All components of SDL and PML are shared by multiple practitioners for efficiency and flexibility improvement. However, resource sharing also brings new challenges for the information managed and processed within the framework. Without a properly implemented isolation mechanism, a practitioner may leak valuable digital assets to an attacker. The isolation within PML is relatively easy to achieve as a manufacturing node will only carry out one task at a time. It is more complex to implement an isolation mechanism in the SDL. Existing technologies on virtual machines isolation [16] can be utilized to mitigate the challenge. But these methods usually rely on the assumption that the SDL infrastructure is trusted. When the SDL infrastructure is owned and managed by multiple parties, the situation is more complex and it is hard to make such an assumption.

Trustworthiness of manufacturing nodes. A manufacturing node accepts instructions from the SDL to finish the production process. The instructions usually include valuable intellectual property details (IPs) and leakage will lead to serious consequences. Another challenge is that the PML needs to guarantee received instructions are used in an expected way. For instance, a manufacturing node cannot "replay" received instructions to make extra products. One way to address these two challenges is to make sure that each manufacturing node in PML is trusted, i.e., it will always follow the protocol with computing nodes in SDL and keep information safe. To achieve this goal, trusted computing hardware can be integrated with each manufacturing node to offer a trusted execution environment (TEE) [17] for the control of the node. While this may not be able to guarantee that such nodes can be fully trusted, it can greatly reduce the attack surface.

Coordination in a decentralized environment. As SDL is owned and maintained by multiple parties, it is possible that two computing nodes that serve different practitioners want to send conflicting instructions to the same manufacturing node. In this case, the software defined manufacturing framework needs to provide a resolution mechanism. The emerging decentralized ledger (DL) technology provides a promising direction to address this issue. A DL is a data structure maintained by multiple peers, who run a consensus protocol to determine whether an instruction should be accepted, for example, blockchain technologies.

Integration of DL and TEE for software defined manufacturing protection. Integration of DL and TEE has the potential to address most of the security concerns in software defined manufacturing. Figure 3 provides an overview of the integrated security protection mechanism:

- Digital assets protection. The digital assets ownership information is managed by the DL in the same way as cryptocurrency, and the contents can be stored in the cloud in an encrypted format, which will only be decrypted when inside the TEE of a manufacturing node.
- Instruction enforcement. The TEE integrated with the manufacturing node only accepts instructions through the DL, and the execution results are sent back to the DL for recording.

Scheduling Optimization

The modern supply chain has been studied a lot and many techniques have been developed

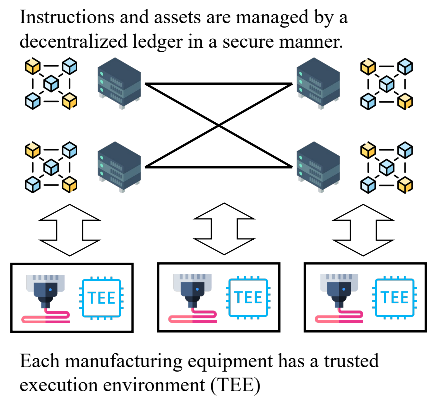

Figure 3. Security protection of the software defined manufacturing infrastructure.

to optimize its operation. The software defined manufacturing framework strives to achieve a balance between users' flexibility and overall system performance. Optimization under this new framework is substantially different from existing problems. Specifically, the software defined manufacturing creates a dynamic environment, where the practitioners are able to interact with the whole manufacturing process through SDL and can thus bring a significant amount of uncertainties. For instance, a practitioner may cancel its production request through SDL to a production node (or a group of production nodes) in PML in the middle of the manufacturing process. This may either due to unexpected situations faced by the practitioner or the practitioner being malicious at the beginning.

Such "reoptimization-compatible" problem is a challenging topic in operations research as most of the common techniques in classical algorithmic design will fail. However, researchers have made some progress in the iterative augmentation method for integer linear programming [18], which is a fundamental mathematical tool for various optimization problems and can be utilized to partially address this specific scheduling optimization challenge.

Compatibility

Most advanced manufacturing equipments use proprietary protocols and do not talk with each other. Under the software defined manufacturing framework, it is possible that the general manufacturing nodes are provided by different vendors that is not compatible with each other. To mitigate this challenge, a gateway can be deployed in the edge computing center which works as a proxy between practitioners and different manufacturing nodes. The practitioner can interact with the framework through a unified interface and does not need to worry about the specific equipment behind.

CONCLUSION

It is impossible to overestimate the importance of the manufacturing sector for a country/region to flourish. Software defined manufacturing framework is based on, but further extends, existing R&D trends such as Industrial 4.0 and cloud manufacturing. The novel framework will transform the current manufacturing business model and pave the way for future manufacturing. It will bring many benefits that are not imaginable with today's manufacturing infrastructures, such as innovation acceleration, better efficiency, and high resilience. Although there are still some temporary challenges to fully realize the framework, solid progresses have been made in most related directions and supporting technologies are becoming more mature.

REFERENCES

1. Tim Stock and Günther Seliger. Opportunities of sustainable manufacturing in industry 4.0. *Procedia Cirp*, 40:536–541, 2016.
2. Amine M Houyou, Hans-Peter Huth, Christos Kloukinas, Henning Trsek, and Domenico Rotondi. Agile manufacturing: General challenges and an iot@ work perspective. In *Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on*, pages 1–7. IEEE, 2012.
3. Xun Xu. From cloud computing to cloud manufacturing. *Robotics and computer-integrated manufacturing*, 28(1):75–86, 2012.
4. Sujan Sarker, Md Abdur Razzaque, Mohammad Mehedi Hassan, Ahmad Almogren, Giancarlo Fortino, and Mengchu Zhou. Optimal selection of crowdsourcing workers balancing their utilities and platform profit. *IEEE Internet of Things Journal*, 6(5):8602–8614, 2019.
5. Richard D Evans, James X Gao, Sara Mahdikhah, Mourad Messaadia, and David Baudry. A review of crowdsourcing literature related to the manufacturing industry. *Journal of Advanced Management Science*, 4(3):224–321, 2015.
6. Fei Hu, Qi Hao, and Ke Bao. A survey on software-defined network and openflow: From concept to implementation. *IEEE Communications Surveys & Tutorials*, 16(4):2181–2206, 2014.
7. Jieyu Lin, Rajsimman Ravichandiran, Hadi Bannazadeh, and Alberto Leon-Garcia. Monitoring and measurement in software-defined infrastructure. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 742–745. IEEE, 2015.
8. Alejandro White, Ali Karimoddini, and Mohammad Karimadini. Resilient fault diagnosis under imperfect observations–a need for industry 4.0 era. *IEEE/CAA Journal of Automatica Sinica*, 7(5):1279–1288, 2020.
9. Arnaldo Camuffo. Rolling out a world car: globalization, outsourcing and modularity in the auto industry. *Korean Journal of Political Economy*, 2(1):183–224, 2004.
10. Fei-Yue Wang, Xiuqin Shang, Rui Qin, Gang Xiong, and Timo R Nyberg. Social manufacturing: A paradigm shift for smart prosumers in the era of societies 5.0. *IEEE Transactions on Computational Social Systems*, 6(5):822–829, 2019.
11. Gang Xiong, Fei-Yue Wang, Timo R Nyberg, Xiuqin Shang, Mengchu Zhou, Zhen Shen, Shuangshuang Li, and Chao Guo. From mind to products: Towards social manufacturing and service. *IEEE/CAA Journal of Automatica Sinica*, 5(1):47–57, 2017.
12. Tuan D Ngo, Alireza Kashani, Gabriele Imbalzano, Kate TQ Nguyen, and David Hui. Additive manufacturing (3d printing): A review of materials, methods, applications and challenges. *Composites Part B: Engineering*, 143:172–196, 2018.
13. Ian Gibson, David W Rosen, Brent Stucker, et al. *Additive manufacturing technologies*, volume 17. Springer, 2014.
14. Naresh D Sanandiya, Yadunund Vijay, Marina Dimopoulou, Stylianos Dritsas, and Javier G Fernandez. Large-scale additive manufacturing with bioinspired cellulosic materials. *Scientific reports*, 8(1):1–8, 2018.
15. J Shah, B Snider, T Clarke, S Kozutsky, M Lacki, and A Hosseini. Large-scale 3d printers for additive manufacturing: Design considerations and challenges. *The International Journal of Advanced Manufacturing Technology*, 104(9-12):3679–3693, 2019.
16. Aurélien Wailly, Marc Lacoste, and Hervé Debar. Towards multi-layer autonomic isolation of cloud computing and networking resources. In *2011 Conference on Network and Information Systems Security*, pages 1–9. IEEE, 2011.

17. Taras A Drozdovskyi and Oleksandr S Moliavko. mtower: Trusted execution environment for mcu-based devices. *Journal of Open Source Software*, 4(40):1494, 2019.
18. Raymond Hemmecke, Shmuel Onn, and Lyubov Romanchuk. N-fold integer programming in cubic time. *Mathematical Programming*, 137(1-2):325–341, 2013.

Lei Xu, is an assistant professor in the Department of Computer Science, University of Texas Rio Grande Valley. He received his Ph.D. degrees in 2011 from the Chinese Academy of Sciences. His research interests include applied cryptography, cloud/mobile security, and decentralized systems. Contact him at xuleimath@gmail.com.

Lin Chen, is an assistant professor in the Department of Computer Science, Texas Tech University. He received his Ph.D. degrees in 2013 from Zhejiang University. His research interests include algorithms and complexity, distributed systems and cybersecurity. Contact him at chenlin198662@gmail.com.

Zhimin Gao, is an assistant professor in the Department of Computer Science, Auburn University at Montgomery. He received his Ph.D. degrees in 2018 from University of Houston. His research interests include blockchain, cloud computing, cybersecurity, IoT, and 5G. Contact him at mtion@msn.com.

Hiram Moya, is an associate professor in the Department of Manufacturing and Industrial Engineering, University of Texas Rio Grande Valley. He received his Ph.D. degrees in 2012 from Texas A&M University. His research interests include Queueing Theory, Optimization, and SCM. Contact him at hiram.moya@utrgv.edu.

Weidong Shi, is an associate professor in the Department of Computer Science, University of Houston. He received his Ph.D. in computer science from Georgia Institute of Technology. His research interests include computer architecture, cybersecurity, and computer systems. Contact him at wshi3@central.uh.edu.