Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 107 (2022) 1150–1155

www.elsevier.com/locate/procedia

55th CIRP Conference on Manufacturing Systems

Comparison of Three Agent-Based Architectures for Distributed Additive Manufacturing

Lorenzo Giunta[a], Martins Obi[a], Mark Goudswaard[a], Ben Hicks[a] and James Gopsill[a,b]

[a]*Design Manufacturing Futures Lab, University of Bristol, UK.*
[b]*Centre for Modelling and Simulation, Bristol, UK.*

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000. *E-mail address:* l.giunta@bristol.ac.uk

Abstract

Models of distributed agent-based Additive Manufacturing (AM) systems have suggested that considerable productivity, reactivity, and resiliency gains can be realised. However, there remains scepticism and real-world validation is required to demonstrate the benefits, practicalities, and challenges of implementing agent-based manufacturing systems. It is only then that potential gains can be evidenced and the Return on Investment fully appraised.
This paper discusses the underlying communication architecture required for the creation of a hardware technology demonstrator that enables the evaluation of distributed, agent-based, AM. Three architectures to create the desired network (Host-Client, Peer-to-Peer, and Digital Shadow) are discussed, with a focus on implementation and operational factors. The paper continues by describing the future development needed for the creation of a demonstrator aimed at exploring the ease of adoption of agent-based approaches and the potential for integration into existing AM workflows. The planned demonstrator is to be implemented at the University of Bristol to co-ordinate production across multiple sites in order to support their education and research activities as well as provide a platform for public and industry engagement.

Keywords: Additive Manufacturing; Agent Based; Network; Distributed Manufacturing; Responsive;

1. Introduction

Ever since its inception, Additive Manufacturing (AM) has promised considerable benefits. Parts made via AM can be easily customised, require less material, and do not require part specific tooling [1]. However, the technology's potential remains underutilised - particularly where responsiveness and agility are concerned. This has been highlighted by the COVID19 pandemic. AM facilities attempted to respond to the increased demand for parts, such as face shields, respirators, and other medical equipment yet the lack of coordination and organization impacted their efficacy and responsiveness [2].

By combining AM with an agent-based architecture it may be possible to improve the responsiveness and resilience of AM. Unlike traditional approaches to production, which can lead to bottlenecks due to their sequential and top down architectures [3], agent-based approaches permit increased flexibility and responsiveness. This is due to the agents' independence, which permits parallelization of decisions as well as real time adjustments to bid for jobs based on the agents' status and objectives [4], [5].

This paper discusses three architectures for implementing a distributed AM system: Host-Client (HC), Peer-to-Peer (P2P), and Digital Shadow (DS). The paper continues by examining the implications of each architecture, and their respective benefits. Additionally, the paper discusses some of the barriers that must be tackled prior to the creation of a demonstrator system, as well as discussing future research plans to further the development of agent-based AM.

10.1016/j.procir.2022.05.123

2. Related work

This section presents relevant literature pertaining to Multi-Agent Systems (MAS), anarchic manufacturing, and digital twins/shadows. The literature discussed in this section provides the basis for the development of the three potential architectures for an agent-based AM system by defining the concepts that will be relied on to develop the architectures.

The architectures will need to cater for two types of agent: the machines producing the AM parts and the jobs being submitted by clients.

2.1. Multi-agent system

Despite the potential logistical advantages presented by AM [6], there remain considerable hurdles to the realisation of AM's full potential. Presently, the logistics process for the creation and distribution of products relies on the flow of components and products from manufacturers to clients. Simultaneously, manufacturers receive information and feedback from the customers [1].

However, as the supply chain becomes more elongated and complex, inertia grows within the system. Should any part of the chain fail, the detection of the failure may be delayed, as well as any response thereto. Furthermore, should there be a sudden desire to make changes, these can become complex to implement: requiring a whole new supply chain to be setup or for clients to work within the constraints of their existing supply chains.

Rather than viewing the supply chain as bidirectional and largely linear, MAS consider it as a network. This network is composed of multiple nodes, each representing an individual agent. These agents each represent a client, supplier, manufacturer, or more generally, any decision maker within the supply chain [7]. Within a MAS, each agent acts individually but is continuously sharing information and data to the other members of the network [8], [9]. This allows each agent to read the status of the overall network and react appropriately thereto. As each agent is independent, they can each attempt to obtain specific outcomes, such as profit or lead time. However, as a collective whole, the MAS shares a common goal, such as the provision of one or more goods or services [7].

2.2. Anarchic manufacturing

It is important to note that a MAS need not be decentralised: the organization of the network can be hierarchical or fully centralised [7]. While each agent may act independently, there may be constraints placed on who they can communicate with, or what intermediaries any decisions must be directed through. An anarchic approach to MAS eschews this in favour of a fully heterarchical approach to the MAS network [10]. In such an anarchic system, full decision-making power is kept with each agent. Each agent attempts to optimize for one or more variables. The emergent behaviour that occurs when multiple agents, who may not share the same optimization parameters, can match or outperform more traditional hierarchical MAS setups [11]. Furthermore, an anarchic manufacturing approach is best suited to non-static situations where this flexibility can best exploit the improved responsiveness of the system [10].

In this context, the two types of agent, Machine and Job, can communicate over a spectrum of possible interactions. This is depicted in Fig 1. At one end of the spectrum is *co-ordinated* manufacture where Job agents are submissive, with the Machine agents negotiating/deciding which jobs they wish to take on. At the other end is the *marketplace* where the Machine agents are submissive and jobs determine which machine they will be created by. In the middle is *brokering* where both Machine and Job agents negotiate and determine which manufacturing resources will process which jobs.

Fig 1. Components and information flow between manufacturers and clients

2.3. Digital twin/shadow

The term Digital Twin (DT) does not have a single widely accepted definition [12], [13]. However, a DT can generally be understood as a digital clone of a physical entity (PE), containing all the relevant information that can be realistically measured and linked to the PE in such a way as to allow the free flow of data between the DT and the PE and vice versa [13]–[15]. How the information flows between the DT and the PE are what distinguishes a DT from a Digital Shadow (DS).

A DS is unidirectional, being updated from information gathered from the PE but does not directly influence the PE itself [12], [16], [17].

For example, a DT of a cutting tool would contain all the relevant information to create a perfect digital clone. Information regarding the material of the tool, number of flutes, coatings, helix angle, and any other necessary values would be stored within the DT. As the cutting tool is used, information regarding its usage would be fed to the DT, updating it to reflect, for example, wear, chipping, and thermal stress. The information stored within the DT would then be used to change the behaviour of the cutting tool: limiting speed, increasing lubrication, or changing other parameters as necessary.

This example highlights the bidirectional nature of information flow in DT as the DT influences the PE and vice versa in real time. Conversely, a DS would still update with the information collected from the cutting tool in real time but would not directly influence how the cutting tool is used [16]. Rather, it could be used to alert the operators regarding the need for maintenance or replacement of the cutting tool.

3. Agent-based architectures concepts

This section discusses three potential architectures to achieve an agent-based network for AM based on MAS, anarchic manufacturing, and DS approaches respectively. Each architecture discussed is presented as a concept and a potential pathway to realisation is described.

3.1. Host-Client (HC)

The HC architecture concept is based on an MAS network. This concept was ideated to easily support the deployment of an agent-based AM control network requiring the least complexity in terms of infrastructure, hardware, and code. Thus, this concept relies on a centralised host for the storage of the Job agents. This hierarchical approach allows for full oversight over job allocation as well as simpler development of the network architecture. This simpler development is supported by the flow of Jobs to Machines, as there is no need to program the interaction between Jobs and Machines, merely that between the Machines and the host, as the jobs are already present in the host. This compartmentalisation of operations typically lends itself to a co-ordinated manufacture approach with the machines-leading the querying and requesting of jobs from the host.

3.1.1. Concept

Fig 2 depicts the HC architecture. Agents do not interact with one another directly, rather all communication occurs through the central host. Clients submit jobs to the host via a submission portal thereby creating a Job agent and therein specifying a number of parameters for its control including: strike price, latest delivery date, minimum resolution, and AM material(s) to be used. These jobs are stored within a pool in the host. AM machines connect via a web browser with the browser providing the Machine agent logic for communicating with the host to bid for jobs based on similar parameters set for each by their respective owners.

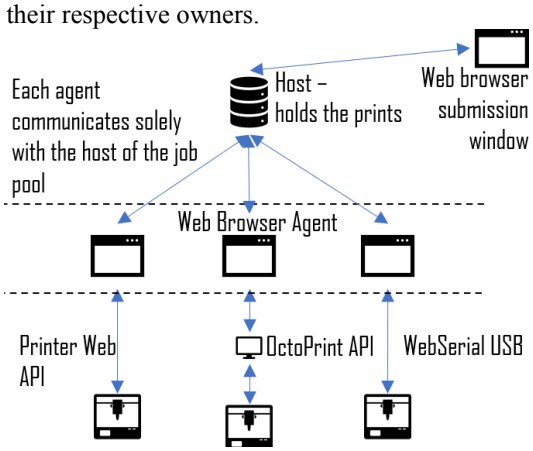

Fig 2. Host-Client Architecture Diagram

3.1.2. Realisation

A realization pathway for the creation of a HC architecture is through the use of a virtual machine (VM) to run the central host server. This server could be hosted locally or in the cloud, such as through AWS. By deploying software to the server using Docker [18] it is possible to containerize the code, allowing for easier deployment across multiple operating systems, simplifying the process of expansion. The use of a VM would also permit the seamless expansion of the service as more resources could be allocated to match demand.

Additionally, the host can be used to store and deploy the user submission window. Clients wishing to submit a job could connect to the host via a webapp.

Communication between the printers and the host could similarly be addressed via a web browser agent. This too could be stored on the host server, allowing the printer owner to simply create a new agent instance by connecting to the host as a manufacturer and setting their parameters for job acceptance. The user interface could similarly be created using web technologies, such as ReactJS, and the communication between the browser and the printer handled by either the printers' native API, Octoprint (an open-source remote printer control software) or through a web serial USB connection.

As all the parameters for both jobs and printers would be stored on the host, job allocation could be performed locally, potentially improving overall processing times. Once a job is allocated to a specific printer, the G-code would be 'pushed' to the printer via the web browser running the local agent instance. This approach simplifies the hardware needed, as any internet capable device can be used to connect a printer to the service, so long as it can also transmit the G-code to the printer once it is received.

3.2. Peer-to-Peer (P2P)

The P2P architecture provides a direct communication between the Machine and Job agents and would typically lend itself to anarchic manufacturing. P2P removes the reliance on a centralised host. This decentralisation allows for improved failover response and protection from malicious actors attempting to prevent access to the host server (e.g., a DDoS attack). Furthermore, the ability to broker directly between printer agents and Job agents improves the privacy aspect of the system. It becomes harder to track which printer agents are being awarded jobs as the brokering occurs directly between Job and Machine agents rather than through a centralised host.

3.2.1. Concept

Fig 3 presents one of two example P2P architecture setups using the platform. In this first setup, the web service that acted as a host in the HC architecture is now a peer-server that is used to broker connections between the agents. Thus, the Job agents are no longer stored on the server. Rather, the web browser being used to submit the job becomes the Job agent and can contact the peer-server to receive information on all the available machines on the network. It can then request a communication channel for a Machine agent directly, asking them to bid on the job and eventually allocate the job to any of the printers it is in contact with (depending on the type of communication implemented c.f. Fig 2). In addition, the Machine agents can be pooled together in the address book and messages broadcast from Job Agents to the Machine Agents (i.e., asking all the machines if they are available and could accept a job).

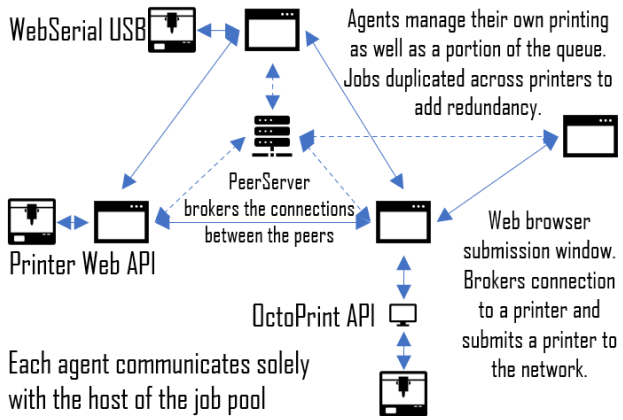

Fig 3. Peer-to-Peer Architecture Diagram with Peer-Server

A challenge for the first P2P network is that the submitter of the job has to keep their web browser open until the job has been accepted by the network. In some instances, and communication structures, this may be instant or take a long time as negotiations get protracted. To alleviate this concern, the platform allows for the Job agents to distributed amongst the Machine agents, as illustrated in Fig 4, thereby the machines collectively act as the Host in the Host-Client architecture. Machine agents are then not only tasked with bidding for jobs, but also acting on behalf of the jobs that they have been given. As the Machine agents are more likely to be permanently online, this allows the Job agents to have protracted negotiations without burdening the submitter of the job.

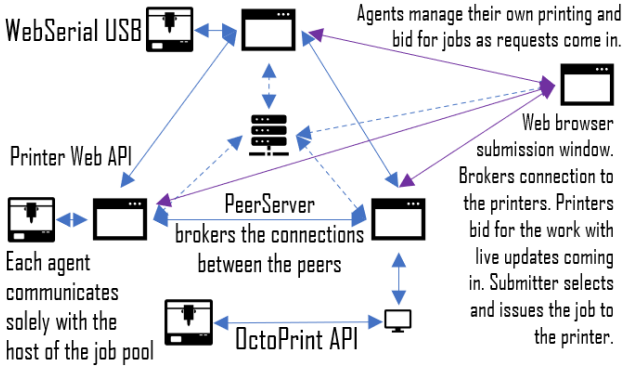

Fig 4. Peer-to-Peer Architecture Diagram with Distributed Job Agents

3.2.2. Realisation

The requirements for the P2P architecture closely match those of the HC architecture. However, the peer-server is expected to require considerably less resources, as it will not have to handle brokering between printer and Job agents. This decrease in resources is nevertheless matched by an increased complexity in the design of the agents themselves.

In the case of a distributed Job agent architecture, the reliance on a gossip protocol allows all the agents on the network to disseminate information between them. This again reduces the reliance on centralized infrastructure but also increases the complexity of the Machine and Job agents who now need to handle additional traffic and process the gossip.

Furthermore, there are difficulties in the authentication process of the Machine and Job agents. With such a decentralized system blocking malicious actors from joining the network becomes more complex, in particular where malicious Job agents are concerned. If a job is submitted with a G-code that could cause damage to a printer, e.g., slamming the printhead into the bed or overheating the nozzle, there are limited ways to detect this and trace it back to a specific malicious actor. As such, it may be necessary to transmit STL files rather than G-code directly, allowing the printer agent to proceed with the slicing based on the parameters set by the job agent.

3.3. Digital shadow

The DS architecture hosts the entire network and agent set within the cloud (via the central web service that was once the host and peer-server) it is possible to create a digital shadow of the agent-based network. This allows for the network structure to be modified on the fly, allowing for the exploration of different conditions as well as the optimization of specific parameters to respond to demand. For example, the structure of the network could be changed from hierarchical to flat in order to reduce the brokering process. The job allocation could, for example, be switched from first-come first-served towards an allocation that minimises the delivery distance for jobs.

3.3.1. Concept

Fig 5 illustrates the architecture of a DS setup. Here the Job and Machine agents are not stored locally but in the cloud. The hardware needed to interact with the cloud thus only needs minimal computational resources in order to create a Job agent and accept a print from the Machine agent as well as communicate and update its status.

As the whole network is in the cloud, it is possible to have omniscient and omnipotent oversight over both network and the job allocation process thereby enabling intervention if necessary. Where a HC would have, by nature, a hierarchical setup, it is possible to restructure the Job agents and the Machine agents into an anarchic setup or any other desired network configuration. Additionally, as the brokering occurs fully within the cloud it is possible to improve the speed of job allocation. The only requirement being that the agents can be linked to their physical manifestations, in order to submit jobs for a client, or accept jobs as a manufacturer.

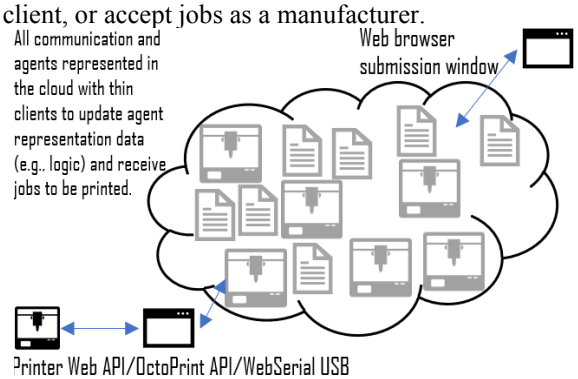

Fig 5. Digital Shadow Architecture Diagram

3.3.2. Realisation

The realization of a DS architecture presents some inherent difficulties. Firstly, the required server space will be larger than that needed for the other architectures discussed here. Nonetheless, this can still be handled by a VM hosted on a cloud computing platform. Furthermore, the complexity of generating and maintaining the agents within the cloud presents some inherent difficulties. The physical entities are required to be linked to their virtual shadows, to receive or submit jobs as the physical entities are no longer acting as agents themselves but have their actions dictated by the cloud. Thus, should a disconnect occur, an appropriate protocol needs to be devised. This could be through the creation of a local backup containing the relevant details for the job.

4. Architecture comparison

Three potential architectures (HC, P2P, and DS) for an agent-based approach to AM were proposed. The architectures each present strengths and weaknesses for the deployment, and eventual adoption, of an agent-based network for AM machine control and job distribution.

Among the factors to keep in mind when selecting architectures are: Security (SEC), Resilience (RES), Maintenance (MNT), Ease of Deployment (EoD), Operating Expenses (OpEx), and Capital Expenses (CapEx).

Table 1 ranks the three architectures against the six evaluation criteria. The evaluation of these concepts was performed by members of the Design Manufacturing Lab through a workshop activity which resulted in the participants scoring the concepts in order of preference, from best (1) to worst (3) in order to arrive at a ranking for each criteria. Notes were taken to identify significant points regarding each architecture for the deployment and management by a service provider intending to provide the service. As the architectures exist only as concepts, it was not possible to provide a quantitative analysis of their performance against one another.

Table 1. Comparison of Architectures for each Criteria (1 Best, 3 Worst)

	SEC	RES	MNT	EoD	OpEx	CapEx
HC	1	3	2	1	2	2
P2P	3	1	1	3	1	1
DS	2	2	3	2	3	3

HC's simplistic nature allows it to perform well specifically if the goal is to obtain an initial functional prototype. This is due to a low CapEx, in particular where small localised setups are concerned, in addition to the ease of deployment and maintenance of the system. Furthermore, the centralised nature of the HC architecture allows for accurate access control, as well as the reduction of anonymity for users accessing the network. This provides an advantage in the form of increased security, while sacrificing privacy and anonymity of users. On premise HCs could be air-gapped from other networks for additional protection. However, the centralised nature of the HC architecture does not provide resilience as there is a single point of failure: the host server. While possible to mitigate this by, for example, adding redundant servers, this increases the complexity and the costs associated with running the HC architecture, increasing OpEx. This is in addition to the base OpEx expected from hosting and maintaining a server. Furthermore, the possibility of a DDoS attack remains, and additional measures need to be taken to mitigate the risk, such as a traffic filter (e.g., Cloudflare). Moreover, as the system expands the maintenance required to support the system increases to handle the growing traffic.

The P2P architecture scored third place for deployment, owing to the more complex network setup. Security also poses a considerable challenge as the anonymity awarded by the system shields malicious actors from consequences. As such, guaranteeing the integrity of each job submitted becomes pivotal. This challenge can be tackled, either through improved authentication protocols, thereby sacrificing privacy, or by changing the job submission to provide STL files rather than G-code directly. This last option can completely bypass the risk of malicious G-code but greatly increases the complexity of the agent as well as job processing time. Nonetheless, P2P architectures provide considerable benefits. In particular, the use of a serverless architecture, such as that illustrated in Fig 4, allows for a system that has no single points of failure and is thus fully redundant. By allowing the users to create and host their own local agent, the majority of the expenditures would be focused on the development of the software required. In the spirit of a fully decentralised platform the code required to join the architecture could be provided to users free of charge, provided they supply their own hardware to host the agents, thereby distributing both CapEx and OpEx costs.

Lastly, the DS architecture appears to provide no immediate benefits over either HC or P2P architectures. The major advantage afforded, providing full control and oversight, may prove a benefit for research purposes but is unlikely to be a valuable addition for industry. This is due to the inherent complexity of such a system which hinders its deployment and maintenance. It should be noted, however, that past the initial deployment the expansion of the DS system should be relatively simple as a cloud computing platform can easily allocate additional resources. However, while it is unlikely that the cloud itself could be brought down by an attack such as a DDoS, the risk remains for a malicious actor to attempt a man-in-the-middle attack. This can, however, be fairly easily mitigated through trusted certificates and end to end encryption.

The three architectures presented within this paper each provide potential benefits for the development of an agent-based AM control network. While the comparison of the architectures highlighted the HC and P2P architectures as the most promising, it did not take into account any specific use cases. The six criteria used for evaluation reflect some parameters that must be optimised to guarantee success but their respective importance may be affected by different scenarios. For example, should the intention behind the network's creation be for improved response to natural disasters, the importance of resilience and deployment would likely be much higher than security or maintenance.

Furthermore, there may be criteria that have not been considered, whose relevance may be highlighted when exploring specific scenarios. For example, the exact parameters of the Machine and Job agents have not yet been explored. Additionally, the supported file formats to be used for the Job agent creation presents some interesting challenges: G-code is machine and AM material specific. When considering a network of multiple AM machines with different types of materials the G-code must be generalized so it may be accepted by different machines. This could be achieved by sharing STL rather than G-code, allowing the Machine agent to generate the G-code, or by completing the brokering process prior to the generation of the G-code, allowing the client to generate the G-code only once the Job agent has successfully allocated itself.

Moreover, while some consideration has been given to security, the exact framework for this has not been explored within this paper. This is vital for the wider scale adoption of agent-based AM manufacturing. Either a central authority must be able to verify the agents registering on the network or some delocalised system must be adopted to avoid malicious actors.

5. Conclusion

This paper set out to develop and appraise potential architectures for the creation of an agent-based AM control network. Three prospective architectures were established: HC, P2P, and DS. Of these three, HC and P2P were determined to be most suitable for the development and deployment of an agent-based network. This was due to HC's high Ease of Deployment and Security rating compared to the other two identified architectures. These high ratings indicate that a HC architecture could be quickly and securely deployed in a pilot study to analyse the impact of agent-based systems. Such a small-scale deployment would likely not incur large Operational Expenses or Capital Expenses costs thus mitigating some of the disadvantages of a HC architecture. P2P was identified as a prospective architecture due to its comparatively higher scores in Resilience, Maintenance, Operational Expenses, and Capital Expenses. Such an architecture would likely be more suited to a larger scale deployment, potentially with multiple manufacturing locations.

Future work should focus on the implementation and exploration of the architectures, both to better understand their potential benefits and limitations, as well as establishing the production scenarios that the architectures are most suited to.

References

[1] S. H. Huang, P. Liu, A. Mokasdar, and L. Hou, "Additive manufacturing and its societal impact: A literature review," *Int. J. Adv. Manuf. Technol.*, vol. 67, no. 5–8, pp. 1191–1203, 2013.

[2] M. Goudswaard, J. Gopsill, A. Ma, A. Nassehi, and B. Hicks, "Responding to rapidly changing product demand through a coordinated additive manufacturing production system: a COVID-19 case study," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1193, no. 1, p. 012119, Oct. 2021.

[3] M. K. Lim and D. Z. Zhang, "An integrated agent-based approach for responsive control of manufacturing resources," *Comput. Ind. Eng.*, vol. 46, no. 2, pp. 221–232, 2004.

[4] M. K. Lim and Z. Zhang, "A multi-agent based manufacturing control strategy for responsive manufacturing," *J. Mater. Process. Technol.*, vol. 139, no. 1-3 SPEC, pp. 379–384, 2003.

[5] J. Gopsill, M. Goudswaard, C. Snider, J. Johns, and B. Hicks, "Achieving Responsive and Sustainable Manufacturing Through a Brokered Agent-Based Production Paradigm," in *Smart Innovation, Systems and Technologies*, vol. 52, 2021, pp. 24–33.

[6] A. R. Mashhadi, B. Esmaeilian, and S. Behdad, "Impact of additive manufacturing adoption on future of supply chains," *ASME 2015 Int. Manuf. Sci. Eng. Conf. MSEC 2015*, vol. 1, pp. 1–10, 2015.

[7] M. Caridi and S. Cavalieri, "Multi-agent systems in production planning and control: an overview," *Prod. Plan. Control*, vol. 15, no. 2, pp. 106–118, Mar. 2004.

[8] Q. Guo and M. Zhang, "A novel approach for multi-agent-based Intelligent Manufacturing System," *Inf. Sci. (Ny).*, vol. 179, no. 18, pp. 3079–3090, 2009.

[9] H. Tang, D. Li, S. Wang, and Z. Dong, "CASOA: An Architecture for Agent-Based Manufacturing System in the Context of Industry 4.0," *IEEE Access*, vol. 6, pp. 12746–12754, 2017.

[10] A. Nassehi and A. Ma, "A Prelude to Anarchic Manufacturing Systems," in *IESM 2017, 7th International Conference on Industrial Engineering and Systems Management*, 2017, pp. 256–261.

[11] A. Ma, A. Nassehi, and C. Snider, "Anarchic manufacturing," *Int. J. Prod. Res.*, vol. 57, no. 8, pp. 2514–2530, Apr. 2019.

[12] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital Twin in manufacturing: A categorical literature review and classification," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018.

[13] E. Negri, L. Fumagalli, and M. Macchi, "A Review of the Roles of Digital Twin in CPS-based Production Systems," *Procedia Manuf.*, vol. 11, no. June, pp. 939–948, 2017.

[14] M. Garetti, P. Rosa, and S. Terzi, "Life Cycle Simulation for the design of Product–Service Systems," *Comput. Ind.*, vol. 63, no. 4, pp. 361–369, May 2012.

[15] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, "Digital twin-driven product design, manufacturing and service with big data," *Int. J. Adv. Manuf. Technol.*, vol. 94, no. 9–12, pp. 3563–3576, Feb. 2018.

[16] S. M. E. Sepasgozar, "Differentiating digital twin from digital shadow: Elucidating a paradigm shift to expedite a smart, sustainable built environment," *Buildings*, vol. 11, no. 4, 2021.

[17] R. Vrabič, J. A. Erkoyuncu, M. Farsi, and D. Ariansyah, "An intelligent agent-based architecture for resilient digital twins in manufacturing," *CIRP Ann.*, vol. 70, no. 1, pp. 349–352, 2021.

[18] E. Casalicchio, "Container Orchestration: A Survey," in *Systems Modeling: Methodologies and Tools*, A. Puliafito and K. S. Trivedi, Eds. Cham: Springer International Publishing, 2019, pp. 221–235.

Acknowledgements

The work reported in this paper was undertaken as part of the Brokering Additive Manufacturing project, conducted at the University of Bristol's DMF Lab (www.dmf- lab.co.uk) and funded by the Engineering and Physical Sciences Research Council (EPSRC), Grant reference EP/V05113X/1.