Comparing Agent-based Control Architectures For Next Generation Telecommunication Network Infrastructures

Marco Pérez Hernández, Duncan McFarlane,
Manuel Herrera, Amit Kumar Jain, Ajith Kumar Parlikad

Institute for Manufacturing, Department of Engineering
University of Cambridge, United Kingdom

[mep53, dm114, amh226, akj29, aknp2] @cam.ac.uk

Abstract: Multi-agent systems have been an effective choice for designing control systems that are flexible and agile. However, few attention has been given to the evaluation of the architectures of such systems. This becomes critical with the emerging requirements in complex domains such as digital network infrastructures. In this paper, we propose an approach for the evaluation of agent-based control architectures and introduce three multi-agent based architectures for the supervisory control of network service operations of the next generation of digital infrastructures. With the proposed approach, we evaluated the architectures and the implemented control systems prototypes under a realistic network infrastructure environment. Our approach has been effective to evaluate the candidate architectures. The results of communication overhead and reaction time, have shown that agent-based *hierarchical* and *heterarchical-ring* architectures have outperformed the *heterarchical-complete network* architecture.

Keywords: Optimization and control of large-scale network systems; Multiagent systems; Decentralized and distributed control

1. INTRODUCTION

The digital telecommunication network infrastructures are fundamental for the realisation of the developments in fields such as the internet of things, autonomous cars, virtual and augmented reality, among others. These infrastructures should support different control domains and deal with vast amounts of operational data. In addition, they should tolerate various demand profiles and resource constraints while delivering high performance services. Given these characteristics, a distributed approach is sensible for the supervisory control of these infrastructures.

Agent-based control systems offer flexibility and inherent distribution of control. Although these systems have been proposed in different domains, it is difficult to evaluate how the architectures of these systems are suitable for the described infrastructures. As control architectures are tailored to the domain and problem at hand, a criteria for comparing such architectures and an approach for their evaluation are still open challenges. The aim of this paper is to assess the suitability of different agent-based supervisory control architectures for the service operations in the context of the next generation telecommunications infrastructures. We focus on: *Q1: Which of these architectures perform better in this context?* and *Q2: What are the key aspects to consider in the assessment these architectures?* We address these questions by identifying a set of key control requirements for service operations of digital infrastructures and by proposing the criteria for evaluation of architectures from the design and operational perspectives. These are our contributions that are presented in the remaining of paper as follows. First, in section 2, we review the relevant reference control architectures and key previous efforts for evaluation of agent-based control architectures. Then, in section 3, we summarise the identified control system requirements. Next, in section 4, we introduce the three agent-based control candidate architectures for evaluation. The design perspective of *Q2* is tackled in section 5 and the operational perspective in section 6. The results and discussion addressing *Q1* are in section 7. Finally, concluding remarks are in section 8.

2. DISTRIBUTED CONTROL SYSTEMS

We review the reference control architectures and key evaluation approaches for agent-based control architectures.

2.1 Control System Architectures

The reference architectures present the essence of functionalities, components, their relations and structure of the control systems. In addition to centralised architectures, there are three main distributed control architectural approaches: *hierarchical*, *heterarchical* and *holonic* (Bongaerts et al., 2000). The *hierarchical* and *heterarchical*

⋆ This work is funded by EPSRC and BT. Grant EP/R004935/1.

are the two ends of the control spectrum whereas *holonic* control is an intermediate solution. In this study we focus on the *hierarchical* and *heterarchical* approaches.

The *hierarchical* architectures are organised vertically across various levels. The supervision function is clear with low-level supervisors commanded by high-level ones (Cai and Wonham, 2016). Dilts et al. (1991) identified two forms of this architecture: the *proper hierarchical*, with decision-making at higher levels and a command/response communication down to lower levels; and the *modified hierarchical*, with coordination among subordinates that might interact with others at the same level, without constant instructions from higher levels (Bongaerts et al., 2000). These architectures usually have limited flexibility to adapt to changes and disturbances (Dilts et al., 1991).

The *heterarchical* architectures do not have a direct controlling component. The supervision is cooperatively carried out and spread across the system. This brings the benefits of reduced complexity and higher flexibility (Duffie and Piper, 1987; Cai and Wonham, 2016). However, this horizontal distribution implies a lack of global view and brings unpredictability to the system, which is why is still controversial whether this is a supervisory control.

2.2 Evaluation of Agent-based Control Architectures

Few works have addressed the problem of evaluating or selecting among a set of control architectures. Liang et al. (2008) focus on transaction-based communication cost and the conflict-resolution lead time to briefly analysed performance of the architectures for Virtual Manufacturing Enterprises. Brintrup et al. (2011) proposed the metrics of response time, stability, scalability, and overall optimality to evaluate agent platforms that enable the vision of intelligent self-service assets. Kruger and Basson (2013) compared the implementation time of multi-agent systems and the IEC 61499 function blocks control approach. Drakaki and Tzionas (2016) used Petri net models to evaluate performance of hierarchical, heterarchical and hybrid systems for warehouse resource allocation. Kruger and Basson (2019) evaluated control systems for a manufacturing cell with attention to the implementation, particularly, development time, code complexity and code reuse, among other metrics. Palau et al. (2019) introduced a cost-component methodology to assess optimality of four agent-based architectures for collaborative prognostics. They compared the total cost of each architecture and processing cost in a simulated environment.

3. SERVICE OPERATIONS FOR DIGITAL INFRASTRUCTURES

The next generation digital infrastructures (NGDIs) are multi-layer networks that enable the provision of customised and heterogeneous digital services (see Fig 1). Beyond the convergent mobile-fixed access networks (Leitão et al., 2016), the physical and the software-defined networks (SDN) (Sezer et al., 2013), other networked systems such as the cyber-physical ones (Serpanos, 2018) are also embedded in the NGDIs making them complex to manage.

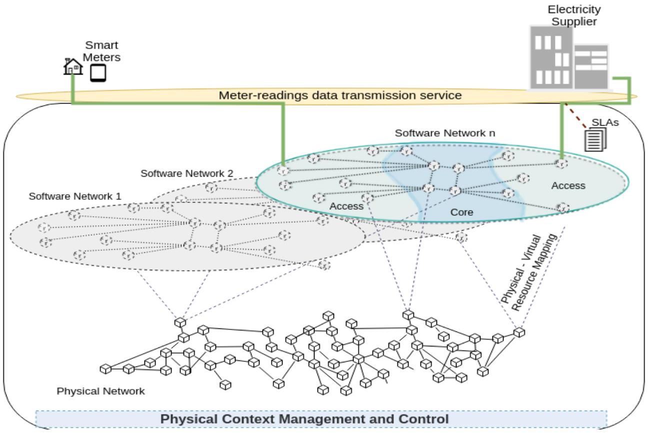

Fig. 1. Next Generation Digital Infrastructure

3.1 Service Operations

Service operations refers to the processes required to ensure that both the service experience and the service outcomes, such as "products", benefits or emotions, meet customer and business expectations (Johnston et al., 2012). In telecommunications, services such as telephony or on-demand video streaming, are a type of *user services* (Stavdas, 2010). These are supported by other *network services*, that refer to data transfer modes (Stavdas, 2010), e.g. a highly secured point-to-point circuit at 100MB/second. Likewise, these *network services* are supported by *IT services* enabling the management of resources, events and other related operational activities.

Network service operations include the *reactive* management of user requests, incidents and root-cause problem investigation as well as the *proactive* data analytics to trigger the adaptation of operations e.g. asset maintenance re-planning. Existing IT Service Management frameworks such as the IT Infrastructure Library (Orr et al., 2011), provide a thorough description of processes, roles and activities involved in the IT service operation. However, different levels of control and automation are required to timely execute expected activities, considering the complexity of the NGDIs, the service quality requirements, the necessary volumes of assets and data and particularly, the demanded fast response times.

3.2 Control Requirements for Service Operations Control

In service operations, the supervisory control system manages the collective strategy, of the telecommunication assets, to deliver services as requested by the user. Together with academic and industrial partners of the NG-CDI [1] programme, we identified the requirements for a supervisory control system in the context of service operations. Below, we describe the requirements we focus on, which are aligned to those identified in the literature (Wang et al., 2017; Qian et al., 2018; Saadon et al., 2019).

Req1 *Multi-service flexibility* - The control system should enable configuration of multiple types of services with different properties, workflows, resources, assets, performance indicators and target levels.

---

[1] http://www.ng-cdi.org/

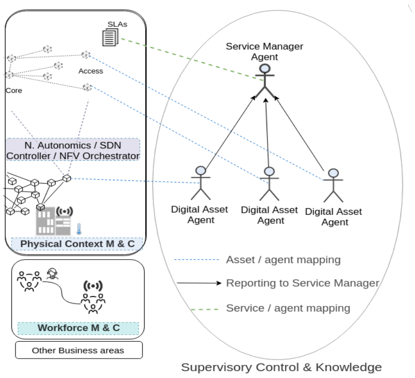

Fig. 2. *Hierarchical* Architecture

Req2 *Service visibility and aggregation* - The control system should enable continuous evaluation of service performance against agreed levels. As service delivery brings multiple resources, assets, processes and roles, creating and maintaining and end-to-end service view implies aggregation of multiple data sources.

Req3 *Online anomaly management* - The control system should detect anomalies during operation and manage them effectively. An effective anomaly management should lead to better and timely fault detection.

Req4 *Multi-demand profile tolerance* - The control system should be aware of variations in the service activity and therefore in the operational measurements due to natural demand profiles –e.g. seasonal fluctuations.

In addition to these requirements, next generation digital infrastructures bring particular challenges, that given the complexity of current architectures and demands of new services, are still open (Qian et al., 2018; Steiner et al., 2016; Saadon et al., 2019). These challenges lead to other desired features for the control system such as support to multiple timescales and control levels, independence of the network under control and a bounded resource usage.

4. AGENT-BASED CONTROL CANDIDATES

The inherent distribution of operational data, resources and decision points, make this service operations control, suitable for an agent-based approach. The agents provide flexibility to the control architecture while enable integration of different data sources and control domains. We designed the following candidate architectures for evaluation.

Hierarchical (HR): As shown in Fig. 2, it has two main agents: Service Manager (SM) that maps one-to-one with services and Digital Asset (DA) that maps one-to-one with network elements. SM performs service aggregation, service view generation and supervision control decision-making. Inbound and outbound interfaces to low-level network controllers enable real-time data collection and actuation, respectively. A Function Provisioner (FP) agent works as a resource mediator between DAs and SMs.

Heterarchical-Ring (HT-R): This architecture has a ring topology as shown in Fig. 3. There are only DA agents also mapped one-to-one to the network elements. They perform data collection, time-based aggregation and keep partial view of the services. Each DA is connected directly to other two neighbour agents with whom DAs exchange operational data to make control decisions. The coordination is via a predefined policy, so DAs trigger control actions when the asset affected is under their responsibility.

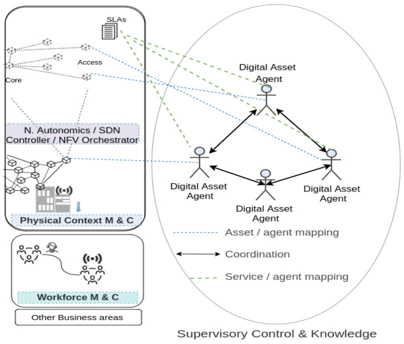

Fig. 3. *Heterarchical-Ring* Architecture

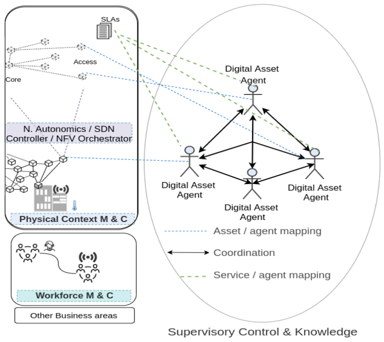

Fig. 4. *Heterarchical-Complete Network* Architecture

Heterarchical-Complete Network (HT-CN): The architecture depicted in Fig. 4 is also *heterarchical* but with a complete network topology. The DA agents store addresses of each other and they collect and report operational data creating a comprehensive service view that is used to make control decisions. Interfaces and communication with domain controllers are common to *HT-R* candidate.

5. CONTROL DESIGN EVALUATION

We introduce the degrees of freedom that support our design criteria and present the assessment of candidates based on these criteria.

5.1 Degrees of Freedom

a) Overall architecture: The *hierarchical* architecture suits digital infrastructure as the data transport networks are commonly organised in hierarchies (Stavdas, 2010). The *heterarchical* architectures favour flexibility, autonomy and concentration of decisions in agents with the same kind of behaviours.

b) Agent mapping: The operation approach with the controlled system can be coupled or embedded (Ribeiro, 2015) and determines how agents are related to entities of the controlled system. These can be services, network elements or functions, links, etc.

c) Agent responsibilities: How these are distributed across agents is a key activity in various methodologies (Wooldridge, 2000; Sterling and Taveter, 2009).

d) Interaction model: The *hierarchical* architecture enables a simple master-slave communication and coordination model (Bongaerts et al., 2000). However, this is more complex in the *heterarchical* ones and opens up various design choices (Bedrouni et al., 2009), e.g. based on available connectivity.

5.2 Design Criteria & Candidate Evaluation

The evaluation of candidates from a design perspective is summarised in Table 1 and discussed below.

Table 1. Design Evaluation

	Criteria / Candidate Architecture	*HR*	*HT-R*	*HT-CN*
Coverage	Req1 Multi-service flexibility	●●●	●●	●●
	Req2 Service visibility & aggregation	●●●	●	●●●
	Req3 Online anomaly management	●●●	●●	●●
	Req4 Multi-demand tolerance	●●●	●	●●●
	Clear traceability with controlled digital infrastructure	●●●	●●●	●●●
	Distribution of operational responsibilities	●●	●	●
	Distribution of control decision making	●●●	●	●●
	Communication/coordination overhead*	●	●●	●●●
	Runtime flexibility	●●	●●●	●●●

Low [●] Medium [●●] High [●●●] - (*) Here the lower is the better.

Requirements Coverage The main criteria are based on how well the design addresses the requirements identified. The multi service flexibility is better addressed by the *HR* candidate with one SM per service. It provides proper organisation, separation of concerns and is fully aligned with the service view. As *heterarchical* architectures do not have a *master*, each DA agent combines monitoring of individual asset and supervision of one or several services simultaneously. In the *HR* candidate, a new service can be added without overloading the DA agents. The opposite occurs in *heterarchical* candidates imposing a constraint in the number of services. The *HR* and *HT-CN* candidates bring best visibility of services enabling aggregation of data from each asset. Thanks to this visibility, the anomaly management is simple in the *HR* candidate and does not requires further coordination. However, the *HT-CN* requires all the agents connected and dealing with a high number of messages between them which leads to scalability constraints. The multi demand tolerance is harder to achieve in *HT-R* candidate, as additional monitoring data comes only from neighbours, then the observed behaviour might not reflect overall service conditions.

Asset Traceability The mapping between agents and the assets varies from one-to-one to one-to-many or many-to-many. The granularity of this relationship depends on how independent the asset is in its behaviour from others and its influence over the overall infrastructure. To favour the clarity of traceability, for the candidates proposed, we selected the one-to-one mapping.

Distribution of Responsibilities Distribution favours flexibility as changes in one function not necessarily affect others. For example, using multiple anomaly detection algorithms not necessary requires to change the way that assets are monitored. The *HR* candidate provides higher distribution of responsibilities with more agents involved.

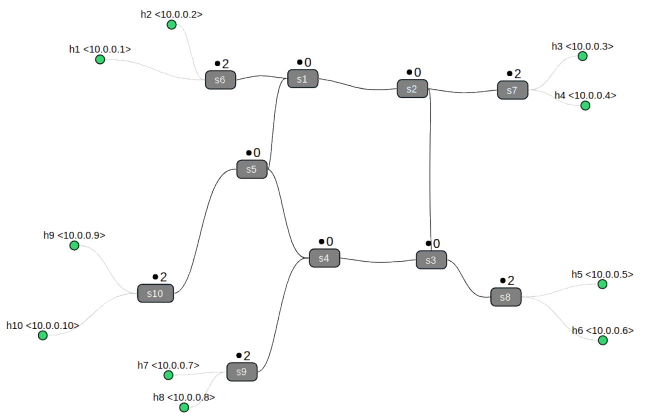

Fig. 5. Emulated Network Infrastructure

Distribution of Control Multiple agents with different control decision-making capabilities provide better separation for the control levels that the infrastructure requires. So, these control decisions can be taken at different time scales and contexts with no interference to each other.

Communication Overhead The master/slave structure of the *HR* candidate is enough to ensure coordination. In the *heterarchical* candidates, the initial decision is the communication topology and the model to maintain the links with each other agent. Furthermore, the coordination approach enables agents to reach consensus on actions to take, given the condition of the system. The latter approaches are more complex from a design perspective and then require the design of protocols and mechanisms to enable the communication and coordination, which brings additional workload to DA agents.

Runtime flexibility The agent-based control approach intrinsically provides runtime flexibility, however this is increased when agents do not depend fully on each other. In the case of the *HR* candidate, the SMs and DAs depend on the FP agent in order to locate each other. In the *heterarchical* candidates agents are more independent and therefore do not rely on a particular agent that enables communication and coordination among them.

6. CONTROL OPERATION EVALUATION

This evaluation is based on the control system prototypes we built with each architecture. The evaluation scenario and the experimental design are presented below.

6.1 Problem Description

We aim to measure the performance of different agent-based supervisory control systems while controlling a digital network infrastructure in operation. Suppose that a temporal network is provisioned for the video streaming of a sports event. It requires a high throughput data transmission between the content servers and clients. Such network is illustrated in Fig. 5, forming a ring topology where switches under supervision are rounded rectangles and hosts are circles. Servers *h1* and *h4* serve contents to clients *h9* and *h7*, respectively. We assume end-users use *h4* and *h7* as cache servers for video streaming. The control

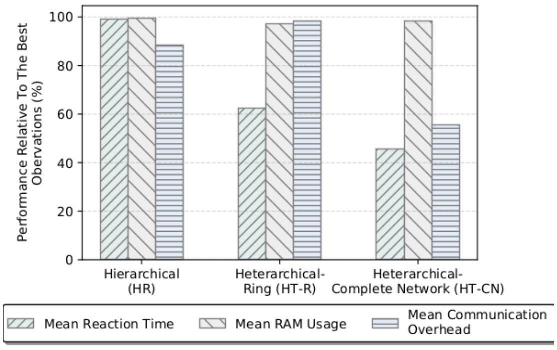

Fig. 6. Performance Comparison of Architectures

system monitors switches *s1* to *s10* to avoid any reduction of the throughput and trigger the corrective action timely. In this case this action is to serve contents from another available server. The low-level network control is enough to guarantee service provision but the supervisory control ensures that target throughput levels are met. This problem gives room to demonstrate operation considering the requirements identified in section 3.2.

6.2 Experimental Design

We simulated traffic from content providers to clients and evaluate the performance of the control systems built from the candidate architectures. The experimental set up is summarised in Table 2. Each system went through periods of standard operation and simulated failures. The digital network topology and the traffic were kept stable.

We focused evaluation on three characteristics: the *resource usage*, measured in MB of RAM used by the agents, the *communication overhead*, given by the bytes transmitted within the system, and the *reaction time*, measured between point t_0, when the traffic is sensed, and the point t_1, when an anomaly is detected. For each characteristic, we calculated a performance indicator relative to the best observation as follows. First, we scaled the observations of each characteristic using the absolute maximum ($MaxAbs$) method (Bonaccorso, 2017). Then, we calculated the vector D with the differences of each observation X_i from the best observation, in this case, X_{min}. Finally, as the lowest difference indicates better performance and we wanted a value in the $[0 - 100]$ interval, we applied the equation 1.

$$P(D) = (1 - D_i) \cdot 100 \tag{1}$$

6.3 Prototypes and Emulation Environment

For each architecture described in section 4 an agent-based control system prototype has been developed in Scala. We used the actor model abstractions as the fundamental building blocks. Actors have been widely used in engineering of distributed system and have shown benefits for handling concurrency and managing programs running in different network nodes (De Koster et al., 2016). We used a toolset[2] that included *akka* actors for building containerised agents, with *mininet* providing a realistic emulated network and *Ryu* as the low-level SDN controller.

Table 2. Experiment Details

Number of runs per architecture	5
Observation time (minutes)	6
RAM Limit Per Agent (MB)	350
Max CPUs per agent	2
CPU Speed (Mhz)	1200

7. RESULTS & DISCUSSION

Fig. 6 shows the performance of the prototypes of each architecture. Overall, the *HR* architecture outperforms the other candidates, especially in the reaction time. The *HR* candidate is slightly better in RAM usage and shows a high performance in communication overhead. The second best is the *HT-R* architecture, with performance above 97% in RAM usage and nearly 10% more in communication overhead than the *HR* option. We now discuss the results obtained for each evaluated characteristic.

Table 3. Reaction Time (Seconds)

Candidate	Mean	Std	Min	Max	% Perf.
Hierarchical (HR)	6.00	0.15	5.88	6.25	99.22
Heterarchical-Ring (HT-R)	11.88	0.52	11.25	12.38	62.5
Heterarchical-Complete Network (HT-CN)	14.57	1.36	12.33	16.00	45.68

Reaction Time In Fig. 6 a higher performance means a lower reaction time. The Table 3 shows that best mean reaction time along the runs is 6 seconds, for the prototype with the *HR* architecture. This is explained, as DAs have predefined SMs they need to report their readings to and also the SMs are specialised in deciding and triggering the actions for the anomalies. This is contrary to the *heterarchical* architectures where DAs have several responsibilities, including network elements monitoring, data aggregation, decision-making and triggering of actions.

Table 4. RAM Usage (MB)

Candidate	Mean	Std	Min	Max	% Perf.
Hierarchical (HR)	287.83	0.68	286.89	288.51	99.68
Heterarchical-Ring (HT-R)	295.56	1.20	293.65	296.64	97.08
Heterarchical-Complete Network (HT-CN)	291.89	2.69	289.37	295.91	98.31

RAM Usage Table 4 summarises RAM usage. The three candidates perform similarly, with the *HR* candidate using slightly less RAM than the others. Considering the available RAM (350MB), the differences among prototypes are low as shown by the performance index with values above 98%. There is no substantial difference in RAM usage among the architectures evaluated.

Table 5. Communication Overhead (MB)

Candidate	Mean	Std	Min	Max	% Perf.
Hierarchical (HR)	1.24	0.02	1.20	1.27	88.62
Heterarchical-Ring (HT-R)	1.04	0.02	1.01	1.07	98.50
Heterarchical-Complete Network (HT-CN)	1.91	0.11	1.75	2.04	55.69

---

[2] https://akka.io http://mininet.org https://osrg.github.io/ryu/

Communication Overhead Table 5 summarises the mean of bytes transferred per agents in each architecture. The mean is particularly high for the *HT-CN* architecture, which almost double the others. This is explained as DAs in *HT-CN* manage connections with each other therefore requiring extensive exchange of messages. The other candidates show similar figures under 1.25 MB transmitted. Note the standard deviation is also low among the runs per architecture. The *HT-CN* architecture performs around a 42% worse than the *HT-R*.

The results of the prototype evaluation are aligned with our initial design evaluation. The dispersion across the runs is low, which give us confidence especially in the communication overhead and reaction time metrics. Ordered from best to worst performance the architectures are *hierarchical*, *heterarchical-ring* and *heterarchical-complete network*. The *hierarchical* candidate easily provides a global service view that is more complex and expensive to achieve in the *heterarchical-complete network* candidate.

8. CONCLUSIONS

We evaluated three agent-based control candidate architectures for the supervision of digital infrastructures. In addition, we proposed an approach for evaluating these type of architectures from a design and operational perspective. In the design perspective, we considered: requirements-coverage, asset traceability, distribution of responsibilities and control. For the operational evaluation, we built three prototypes, one per each candidate architecture and set a realistic environment based on an emulated network. We measured resource-usage, with focus in RAM, communication overhead and the reaction time. We built a performance index based on the difference of observations of each evaluated characteristic to the best value obtained. Our results show that the *hierarchical (HR)* followed by the *heterarchical-ring (HT-R)* candidate architectures perform better in the anomaly detection scenario, over a digital infrastructure, in comparison to the *heterarchical-complete network (HT-CN)* architecture.

REFERENCES

Bedrouni, A., Mittu, R., Boukhtouta, A., and Berger, J. (2009). Coordination Strategies and Techniques. In *Distributed Intelligent Systems*, 1–181.

Bonaccorso, G. (2017). *Machine learning algorithms.* Packt Publishing Ltd.

Bongaerts, L., Monostori, L., McFarlane, D., and Kádár, B. (2000). Hierarchy in distributed shop floor control. *Computers in Industry*, 43(2), 123–137.

Brintrup, A., McFarlane, D., Ranasinghe, D., Sánchez López, T., and Owens, K. (2011). Will intelligent assets take off? Toward self-serving aircraft. *IEEE Intelligent Systems*, 26(3), 66–75.

Cai, K. and Wonham, W.M. (2016). Localization for large-scale systems. In *Supervisor Localization*, 73. Springer.

De Koster, J., Van Cutsem, T., and De Meuter, W. (2016). 43 years of actors: a taxonomy of actor models and their key properties. In *Proceedings of the 6th Intl. Workshop on Programming Based on Actors, Agents, and Decentralized Control*, 31–40.

Dilts, D., Boyd, N., and Whorms, H. (1991). The evolution of control architectures for automated manufacturing systems. *Journal of Manufacturing Systems*, 10(1), 79.

Drakaki, M. and Tzionas, P. (2016). Modeling and performance evaluation of an agent-based warehouse dynamic resource allocation using Colored Petri Nets. *Intl. Journal of Computer Integrated Manufacturing*, 29(7), 736.

Duffie, N. and Piper, R. (1987). Non-Hierarchical Control of A Flexible Manufacturing Cell. *Robotics & Computer Integrated Manufacturing*, 3(2), 175–179.

Johnston, R., Clark, G., and Shulver, M. (2012). *Service Operations Management.* Pearson Education Limited.

Kruger, K. and Basson, A.H. (2013). Multi-agent systems vs IEC 61499 for holonic resource control in reconfigurable systems. *Procedia CIRP*, 7(August 2014), 503.

Kruger, K. and Basson, A.H. (2019). Evaluation of JADE multi-agent system and Erlang holonic control implementations for a manufacturing cell. *Intl. Journal of Computer Integrated Manufacturing*, 32(3), 225–240.

Leitão, F., David Carnero Ros, R., and RiusRiu, J. (2016). Fixed-mobile convergence towards the 5G era: Convergence 2.0: The past, present and future of FMC standardization. *2016 IEEE Conference on Standards for Communications and Networking, CSCN 2016*, 1–6.

Liang, F., Fung, R.Y., Jiang, Z., and Wong, T.N. (2008). A hybrid control architecture and coordination mechanism in virtual manufacturing enterprise. *Intl. Journal of Production Research*, 46(13), 3641–3663.

Orr, A.T., Britain, G., et al. (2011). *Introduction to the ITIL service lifecycle.* The Stationery Office.

Palau, A., Dhada, M., and Parlikad, A. (2019). Multi-Agent System architectures for collaborative prognostics. *Journal of Intelligent Manufacturing.*

Qian, F., Ye, Y., Shan, N., and Su, B. (2018). A novel architecture of telecommunication networks for next generation internet. *MATEC Web of Conferences*, 173.

Ribeiro, L. (2015). *The Design, Deployment, and Assessment of Industrial Agent Systems.* Elsevier Inc.

Saadon, G., Haddad, Y., and Simoni, N. (2019). A survey of application orchestration and OSS in next-generation network management. *Computer Standards and Interfaces*, 62(June 2018), 17–31.

Serpanos, D. (2018). The Cyber-Physical Systems Revolution. *Computer*, 51(3), 70–73.

Sezer, S., Scott-Hayward, S., Kaur Chouhan, P., Fraser, B., Lake, D., J., F., Viljoen, N., M., M., and T., N.R. (2013). Are We Ready for SDN? Implementation Challenges for Software-Defined N. *Future Carrier Networks*, 51(7), 36.

Stavdas, A. (2010). *Core and Metro Networks.* Wiley.

Steiner, W., Peon, P.G., Gutierrez, M., Mehmed, A., Rodriguez-Navas, G., Lisova, E., and Pozo, F. (2016). Next generation real-time networks based on IT technologies. *IEEE Intl. Conference on Emerging Technologies and Factory Automation, ETFA*, 2016-Novem.

Sterling, L. and Taveter, K. (2009). *The Art of Agent-Oriented Modeling.* The MIT Press.

Wang, K., Wang, Y., Zeng, D., and Guo, S. (2017). An SDN-Based Architecture For Next-Gen. Wireless Networks. *IEEE Wireless Communications*, 24(1), 120.

Wooldridge, M. (2000). The Gaia Methodology for Agent-Oriented Analysis and Design. *Autonomous Agents and multi-agent systems*, 285–312.