MOSAIK: A Formal Model for Self-Organizing Manufacturing Systems

Victor Charpenay, Daniel Schraudner, Thomas Seidelmann, Torsten Spieldenner, Jens Weise, René Schubotz, Sanaz Mostaghim, Andreas Harth

***Abstract*—In this paper, we review past and current system architectures displaying self-organization in the domain of manufacturing. Based on a corpus of 84 reference papers, we find that multi-agent systems (MAS) play a significant role in self-organization, especially MAS featuring bio-inspired algorithms for agent coordination. The emergence of new classes of Cyber-Physical Systems (CPS) further strengthens the prevalence of MAS on the subject. As outcome of our review, we devise the MOSAIK model, a generic model synthesizing all system architectures found in our corpus. The MOSAIK model can be used as a reference for formally comparing distinct architectures. We also use it to identify gaps for future research on self-organizing manufacturing systems. The model includes the central concepts of Agent and Artifact, which suggest that the Web is an adequate communication infrastructure for modern manufacturing systems: Agents become (autonomous) Web Agents and Artifacts become resources exposed by Web servers.**

***Index Terms*—Self-Organization, Manufacturing, Industry 4.0, Multi-Agent Systems**

I. INTRODUCTION

Manufacturing systems always had to deal with contradictory optimization criteria, such as maximizing production throughput while minimizing energy consumption. Although optimization over time (producing many small batches of few product variants) or space (producing few large batches of many product variants) can be individually solved in an efficient way, favoring one dimension generally comes at the expense of the other. Yet, the objective of recent research is precisely to get the best of both worlds by building manufacturing systems that are capable of optimally producing many small batches of many product variants. This property is referred to as *mass customization* [1]. In addition to being capable of mass customization, manufacturing systems should be predictable, in particular with respect to potential failures. A high level of predictability in that respect would allow for *predictive maintenance* [1]. Yet again, prediction becomes increasingly difficult when production lacks regularity, as is induced by mass customization.

The hope that all these contradictory criteria may still be combined into a single system has its roots in recent advances in communication technologies. These allow for Cyber-Physical Systems (CPS)—at the basis of most manufacturing processes—of much greater complexity than previously considered [2]. Considering the high inherent complexity of CPS, it is far from trivial to obtain a configuration which shows the desired properties of mass customization and predictability. Cyber-physical manufacturing systems must thus show yet another property (or quality) to fulfill their objective: *self-organization*. There already exist several approaches for self-organizing manufacturing systems [1], [3]–[5], which largely adopt techniques from the field of Multi-Agent Systems (MAS). Yet, the most recent survey on the topic [6] discusses gaps in terms of missing technologies or industry standards rather than in terms of research opportunities. In contrast, in this paper, we introduce a formal model for the domain of manufacturing, the MOSAIK model, to be used as a reference for future research on self-organization in manufacturing.

The remainder of the paper is structured as follows: Sec. II summarizes a corpus of publications that are representative of research on self-organizing manufacturing systems. In Sec. III, we introduce MOSAIK, our domain model for manufacturing, and then show in Sec. IV how this model covers our corpus and how it suggests possible gaps for future research. Sec. V concludes the paper.

II. CORPUS SELECTION & CLASSIFICATION

A. Source Corpus

The model we aim to build is at the intersection of two topics: self-organization and manufacturing. To identify relevant publications, we performed a systematic review of a corpus consisting of 196 journal papers exported from Web of Science (an open-access publication database). The corpus was obtained as a result of the plain query “self-organization AND manufacturing”. This source corpus mixes research from very different topics, such as the self-assembly of polymers or swarm robotics. To refine the scope of our review, we assigned one of the following high-level disciplines to all publications: materials science, biology, ecology, applied physics, applied mathematics, philosophy, industrial engineering, and computer science. We then kept papers from the latter two disciplines only, as they are the only ones in which self-organization is controlled by software. The corpus we obtained after filtering includes 67 source papers and 1,320 cited papers. There are 84 papers that are cited multiple times and form a dense citation graph. We therefore selected these 84 papers as the reference corpus for our review. The source corpus, along with the filtering process, is documented online [7]. Next, we provide a conceptual overview of the filtered reference corpus.

B. Filtered Corpus

The concept map depicted on Fig. 1 summarizes all concepts we could identify in our corpus. This concept map

Manuscript first received June 1st, 2020; revised October 10th 2020.

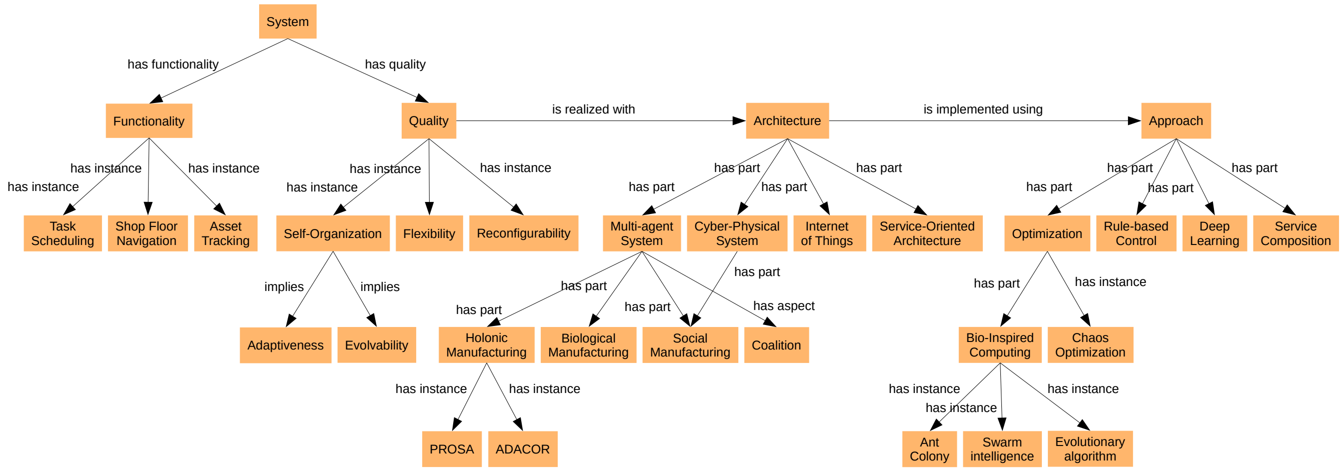

Fig. 1. Concept map for a corpus of publications on self-organizing manufacturing systems

was built by hierarchically grouping keywords automatically extracted from the papers' titles. Figure 1 only depicts the resulting categories but not the extracted keywords, for space reasons. Similarly, all citations in this section are to be interpreted as examples instead of exhaustive lists. We refer to the online documentation of the corpus for the full classification [7].

The concept map can be read by following relationships, starting from the topmost concept of system. That concept has relationships to two further concepts: functionalities and qualities. We start with the typical functionalities of a manufacturing system in which self-organization has a role to play (in order of importance): task scheduling [3], shop floor navigation [8], and asset tracking [9], a recent functionality that was made possible by the Internet of Things (IoT). Then, we move on to the qualities that a manufacturing system displays in relation to self-organization. Although flexible and reconfigurable manufacturing systems appeared as early as in the 1980s [10], not all such systems are self-organized. Evolvable assembly systems, however, combine flexibility or reconfigurability with self-organization [4], as do adaptive manufacturing systems [11].

Reading the concept map further, we have several architectures that help realize self-organization. As mentioned in Sec. I, the MAS architecture is the most important one in that respect. A MAS in the context of manufacturing is also often a holonic system, which can be understood as a "recursive" MAS (agents being themselves multi-agent subsystems). Most papers in our corpus refer to PROSA [12] and ADACOR [11] as baseline holonic manufacturing architectures. We will provide more details on these approaches in Sec. IV. PROSA/ADACOR extensions include, in particular, methods to create coalitions of agents and contract-based negotiation [13]. ADACOR has been later refined to ADACOR^2, accentuating the self-organizing property of the system [5]. However, it is worth noting that the proposed approach, consisting of applying bio-inspired interaction mechanisms between agents, had originally been introduced much earlier. Under the name of biological manufacturing it was already described in 1997, 18 years before ADACOR^2 [14]. With the emergence of the Industry 4.0 paradigm in 2014, several architectural concepts have gained importance in our corpus: CPS [2], in the first place, but also the IoT [9] and Cloud computing, in the form of the Service-Oriented Architecture (SOA) [15]. CPS design directly depends on IoT and SOA. As these new topics emerge, MAS principles remain of importance, e.g. under the name of Social Manufacturing [16]. As our corpus suggests, CPS, IoT and SOA do not replace MAS research but rather amplify it.

To finish our overview of the concept map, we list concrete approaches to implement a MAS in manufacturing: optimization techniques, such as bio-inspired algorithms [1] or chaos optimization [17], rule-based control (including fuzzy rules, for agents to be capable of evolving) [18], service composition on SOA-based Cloud systems [17], and deep learning [19]. Bio-inspired algorithms are by far the most frequent approach in our corpus, especially those based on stigmergy. Stigmergy is a type of indirect communication between agents that is realized by leaving traces in the environment [20].

C. Latest Publications

Our corpus of 84 papers spans about two decades. Yet, besides the emergence of CPS-related concepts induced by Industry 4.0, we could not identify any conceptual break over the whole period. PROSA and ADACOR remain the baseline architectures for most approaches, with only subtle differences between the two (see Sec. IV). Because our source corpus extracted from Web of Science is partial regarding most recent publications, we further reviewed publications from 2018 or later that relate to self-organization and manufacturing. We also included publications relating to CPS and MAS from that period, given the conclusions of the previous section.

The many publications that are still being published indicate that self-organizing manufacturing systems is still an active research topic. However, the approaches seem to remain within the frame of MAS with bio-inspired algorithms (to control agents' behavior or their interactions). Interestingly, Web ontologies and similar knowledge representations, shared across agents, play a greater role in these publications than in our corpus [21]–[24], although the topic has always been part of MAS design. This evolution may be explained by the increasing availability of sensor data in a CPS, that agents must exchange without losing contextual metadata. Contextual reasoning is also the motivation for extending the PROSA/ADACOR baseline with agents that are also capable of interacting on the basis of assumptions, rather than directly observable facts (model-based agents) [25]. A similar attempt had been made in 2017 with the CASOA architecture [26], but even older studies with similar objectives can be found in our corpus, where stigmergic interactions allow simpler agents to exchange partial plans [20]. The main noticeable difference over time between the various publications is scale (with respect e.g. to the amount of exchanged data, the complexity of industrial processes or the number of agents). As a result, special care is given to scalability and energy efficiency issues in recent work [27]–[29].

From reviewing our corpus, it appears that a common baseline (MAS) architecture underlies most studies. This general observation strongly suggests to formalize this baseline in an axiomatic way. A formal model could then be used to compare existing approaches, as well as to identify gaps that our corpus may have left. This is the subject of the remainder of this paper.

III. THE MOSAIK DOMAIN MODEL FOR MANUFACTURING

We now introduce a generic domain model for self-organizing systems in manufacturing: the MOSAIK domain model (referring to a project entitled *Methodik zur selbstorganisierten Aggregation interaktiver Komponenten*—Method for the Self-Organized Aggregation of Interactive Components). This model, summarized on Fig. 2, shall include all domain elements required to describe the systems, architectures, and approaches that were exposed in Sec. II. In the following, we first present the core domain elements of MOSAIK in more detail and then introduce certain patterns that one can apply to instantiate the model.

A. Core Domain Elements

Our domain model relies on a basic meta-model that includes four elements: *classes*, *association* relationships, *inheritance* relationships and *composition* relationships. The semantics of these meta-elements follows the Unified Modeling Language (UML) specification. Note that we use the word 'entity' as a synonym for 'UML object' in the following, so as to avoid confusion with 'the object of an action'.

We describe the MOSAIK model next, by means of definitions.

Definition 1. *An* Agent *is an entity that can perform Actions. An* Action *is a temporal entity that induces changes on other entities (not themselves Actions). An Action may change the attributes of some entity, then called the object of the Action, or it may result in the creation of a new entity. An Action may also require some instrument, an entity that is not changed by the Action, or other Agents, then called participants.*

Our definition of Agent does not specify whether Agents are humans or pure software agents, leaving room for human-machine interaction.

Definition 2. Artifacts *are physical entities that are located somewhere on a* Shop Floor*. Artifacts are either* Storage Spaces*,* Transportation Devices*,* Workstations *or* Products*. Each of these entity types implies a specific involvement in actions. A Product is an entity that can be carried by Transportation Devices via* Move Actions*. A Product can also be processed at some Workstation as the object of a* Control Action*, which may result in a different Product (e.g. as the effect of assembling two other Products). A Storage Space is any other entity of interest on a Shop Floor.*

In the literature, we find various terms for the artifacts we list in our model: a Workstation may also be a 'machine' or a 'workshop'; an intermediary Product is also called a 'workpiece'. What is important however, is that any of these terms denotes a specific instance of the corresponding class and not a more general entity. The terminology we have adopted in MOSAIK is derived from `schema.org`. The notion of Artifact comes from a well known MAS architecture: *Agents & Artifacts* (A&A), which aims at separating agents that have goals from purely reactive entities that belong to the agents' environment [30]. These reactive entities are collectively referred to as 'artifacts' in the A&A architecture. It is worth noting that Workstations or Transportation Devices, when used as instruments by software Agents, must be cyber-physical, that is, they must provide a digital interface to the operating Agent.

On another level, there is no notion of 'product order' in our model. However, an order may be seen as an interaction between agents that leads to the creation of some Product (via a Control Action) followed by its shipment (a Move Action).

Definition 3. *A* Manufacturing Process *is a temporal entity that encapsulates the steps to manufacture a given Product. It is therefore composed of Move Actions and Control Actions.*

Definition 4. Interact Actions*,* Update Actions *and* Check Actions *involve no Artifact. In an Interact Action, one agent sends a message to one or several other Agents participating in the interaction. Update Actions and Check Actions refer to updating and actively retrieving digital information stored in a collection of entities called a* Dataset*.*

Agent interactions and operations on digital Datasets (which can range from a large relational databases to a few bytes of memory on a chip with radio-frequency identification—RFID) are the domain elements that have been studied most in academic research so far.

Next, we present a number of design patterns that can be defined using the MOSAIK model.

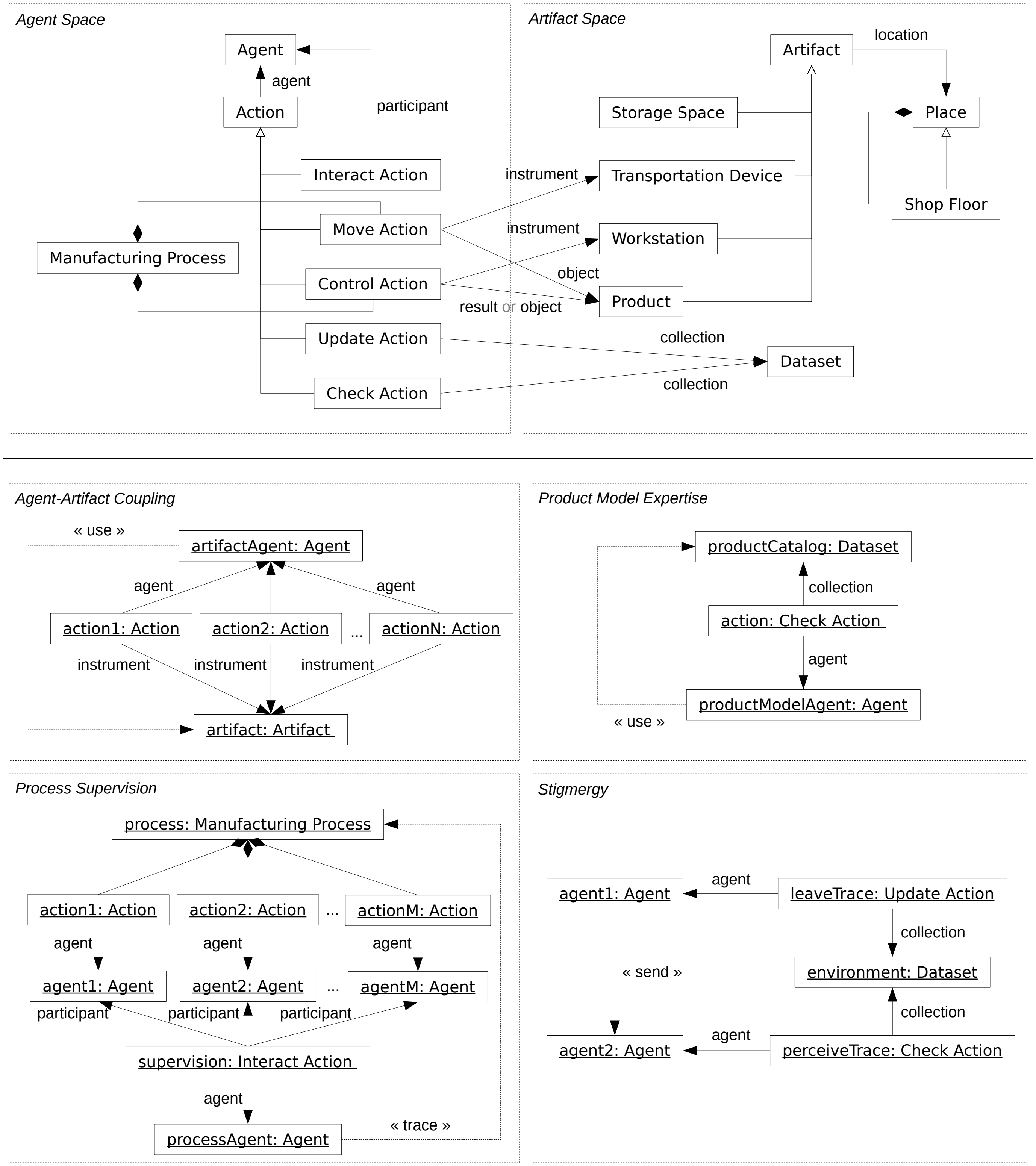

Fig. 2. The MOSAIK model for the manufacturing domain (top) and four specific instantiations of the MOSAIK model, defined as design patterns (bottom)

B. Design Patterns

The MOSAIK domain model itself was designed to have only few model elements. It includes less than 50 model elements in total. However, when instantiating this model for the various system configurations found in our corpus, patterns could be identified. In the following, we introduce some of these *design patterns*, i.e. instantiations of (parts of) the MOSAIK model in specific arrangements that introduce dependencies between entities. We introduce the following design patterns, also depicted on Fig. 2:

- Agent-Artifact Coupling
- Product Model Expertise
- Process Supervision
- Stigmergy

The most straightforward design pattern we could identify is Agent-Artifact Coupling. This pattern consists in assigning one Agent to one Artifact for the whole lifetime of a system. In particular, an Agent may be responsible for managing a single Workstation or Transportation Device, but other Artifacts may as well be considered.

We refer to the second design pattern as Product Model Expertise. It consists in exposing various Product models in a Dataset (a catalog) and assigning one Agent to each Product model. This Agent would then be capable of planning a particular Manufacturing Process given Product model requirements and available Workstations on a Shop Floor.

The next design pattern, Process Supervision, relates to a specific Manufacturing Process. Such a process starts with placing an order and results in a final Product to be shipped. The simplest way to guarantee continuity across all steps in the process is to assign one Agent to the whole process. This Agent ensures process steps are executed in the right order while delegating the particulars of each step to specialized Agents. Alternatively, for more complex processes, a coalition of Agents may instead be formed, such that Agents collaborate in an unsupervised way with each other, but not with other Agents, to execute the process. A Process Supervision Agent may rely on a Product Model Agent to plan process execution in an optimal way.

The last design pattern we identified, mostly used in bio-inspired approaches, is Stigmergy. Direct interactions (i.e. one or more Interact Actions) are replaced by sequences of updating and checking information in a Dataset. This form of interaction is a computationally efficient substitute for Agent negotiation. It is also possible to implement “physical” stigmergy, where the state of Artifacts themselves are updated and checked, but none of the reviewed publications explicitly mentions this possibility.

IV. EVALUATION OF THE MOSAIK MODEL

To evaluate the MOSAIK domain model, two aspects must be considered. First, to show that no model element is missing to fully capture domain knowledge (in other words, that MOSAIK is *sufficient*), one must ensure that all systems described in our corpus can be properly instantiated. To that end, it is enough to map all concepts shown on Fig. 1 to MOSAIK model elements or to the design patterns introduced in the previous section. Second, one must show that none of the MOSAIK model elements are superfluous (that is, that MOSAIK is *necessary*). Once concepts are mapped to MOSAIK, it is straightforward to verify that all model elements are indeed used at least once.

In the following, we go once again through the concepts of Fig. 1 and detail how each concept maps to the MOSAIK domain model, thus showing that MOSAIK is necessary and sufficient for the manufacturing domain. Then, given that our corpus does not cover all possible instantiations of MOSAIK, we identify research opportunities through novel design patterns for self-organizing systems.

A. Mapping to Concepts in the Corpus

The two main functionalities sought in a manufacturing system respectively refer to optimally ordering Control Actions or full Manufacturing Processes (task scheduling) and to ordering Move Actions (shop floor navigation). The other functionality mentioned in our review, asset tracking, refers to the property of all Artifacts to be cyber-physical, including e.g. Storage Spaces, by providing a digital interface to Agents. Note, however, that the communication network of a system is not captured in MOSAIK, given that communication is transparent to Agents.

The quality of manufacturing systems we are most concerned with in this paper is self-organization. The A&A architecture featured in MOSAIK provides us with a clean formulation for self-organization.

Definition 5. Self-organization *is the ability of a single set of Agents to operate on a broad range of configurations of Artifacts and according to a global objective. A configuration, in this context, is a given set of Artifacts at given locations.*

The above statement defines self-organization only with respect to Agents. Indeed, Artifacts being reactive components, they may not evolve or adapt by themselves. However, Artifacts may appear on a Shop Floor (e.g. when modifying its layout) or disappear from it (e.g. if defective), hence the importance of self-organization. The Agents’ objective is mostly dictated by product orders received by the factory, which can be captured in MOSAIK as a Dataset (an order book).

To be able to properly map the various architectures that we reviewed, we refer to the MOSAIK design patterns. Indeed, all patterns but Stigmergy capture the commonalities between PROSA and ADACOR. As mentioned in Sec. II, PROSA and ADACOR share many characteristics. Particularly, both approaches include ‘product holons’, which translate to Agents implementing the Product Model Expertise design pattern in MOSAIK. While PROSA specifies ‘resource holons’ and ‘order holons’, ADACOR redefines them as ‘operational holons’ and ‘task holons’. In both cases, these are special instances of the Agent-Artifact Coupling design pattern and the Process Supervision design pattern, respectively. The definitions of holons in these architectures mostly differ in the fourth kind they each define: PROSA defines ‘staff holons’, which can be seen as generic helpers that may interact with other holons, while ADACOR defines 'supervision holons', which are responsible for monitoring global patterns in the factory. Both kinds of holons correspond to Agents that perform only Interact Actions. Besides PROSA and ADACOR, the most common patterns across our corpus are Agent-Artifact Coupling [2], [4], [15], [21], [22], [24], [26], [27] and Process Supervision (in which the Agent is often called 'product agent') [4], [15], [22], [25]–[27]. The last design pattern, Stigmergy, captures the specificity of biological manufacturing. Exchange in social manufacturing and coalition formation among agents, the remaining MAS-related concepts of Fig. 1, both translate to Interact Actions in MOSAIK. The other concepts from the architecture branch imply greater numbers of Datasets in MOSAIK instances. We will come back to this aspect in the next section.

The last part of the concept map, on specific approaches, is only partially covered by MOSAIK for the following reason: as a domain model, MOSAIK aims at providing a synthetic view on a given system instance, as opposed to providing an exhaustive account of all its components. We therefore decided not to include a model of agent behaviors in MOSAIK, in order for instantiation to remain simple. In other words, MOSAIK only provides an observer's view on a system instance, which should, however, be enough to recognize qualities such as self-organization. As a consequence, approaches that refer to the specification of individual agent behaviors (rule-based control, deep learning, service composition and some optimization techniques) are not part of MOSAIK. Other approaches (ant colony optimization and swarm intelligence) are included through the Stigmergy design pattern.

B. *Research Opportunities*

Given our definition of self-organization in MOSAIK terms (Def. 5), it is clear that the capabilities of Agents depend on those of the Artifacts they control. Yet, research on designing Artifacts goes beyond the scope of this paper, limited to self-organization and thus, to Agents. We identified 3 novel design patterns that are potentially of interest with respect to Agents.

First, when considering a higher level of independence of Agents towards Artifacts, the problem of *attention* emerges. Indeed, to detect changes on the Shop Floor, e.g. the delivery of spare parts (new Products), Agents must not only focus on what they act on but also "look around" and adapt their attention span accordingly. This pattern can be broadly formulated as defining an artificial perception range for Agents, such that Agents only act on those Artifacts within range.

Second, an issue that is rarely considered in our corpus is the problem of limited data access. Most CPS, IoT and SOA architectures from our corpus feature a global and constant access to Datasets on Cloud platforms (even RFID readings, which are forwarded to gateways). However, it is unlikely that all data required by Agents will eventually reach the Cloud in real-world systems, for scalability reasons. The notion of location- and time-dependent Datasets must therefore be integrated to future manufacturing systems, i.e. Datasets for which Update Actions and Check Actions are only possible in certain Artifact configurations.

Last, what could be the highest level of self-organization is to let design patterns emerge from a collection of generic Agents under reinforcement learning. Agents are indeed the most versatile if patterns such as Agent-Artifact Coupling, Product Model Expertise or Process Supervision are not hard-coded in an Agent's behavior specification but if they instead result from interacting with Artifacts and learning using feedback loops. While such an approach is likely not to be feasible on physical Artifacts, advanced Artifact simulations may help reach such a level of self-organization. The pattern involved here consists in instantiating a system with identical Agents of the most abstract type in its initial state.

The publications we reviewed from 2018 or later include a promising idea to address all three challenges, consisting in using the architecture of the Web to separate Agents from Artifacts [24]. The latter are mirrored on the Web by Web servers, exposing a symbolic representation of the physical world to (Web) Agents. The perception of Agents may be constrained by following a subset of the hyperlinks exposed by Artifacts; Artifact configurations themselves dictacte what hyperlinks Artifacts should expose to Agents, turning the problem of availability of Datasets into a reachability problem; generic Agents may learn new capabilities at runtime, by retrieving Web ontologies referenced by Artifacts. This architectural principle, also called the Web of Things (WoT) [31], [32], may therefore be an interesting baseline to improve self-organization in future manufacturing systems.

V. CONCLUSION

After reviewing a selected corpus on self-organization in manufacturing, we could summarize past research in the field with few concepts, as our concept map illustrates (Fig 1). In turn, all publications under review could be seen as instances of a single model for self-organizing manufacturing systems that includes no more than 50 model elements. This model, the MOSAIK model, is the main outcome of our review (Fig. 2). The classes of Agent and Artifact are the central elements of the model.

What our review further highlights is recurring patterns across our corpus of publications that can be defined as MOSAIK design patterns. In this review, we introduced four: Agent-Artifact Coupling, Product Model Expertise, Process Supervision and Stigmergy. Three of these four design patterns show the dependency of an Agent to some Artifact. The last one, Stigmergy, shows a dependency between two or more Agents.

The distinction between an Agent space and an Artifact space in MOSAIK suggests a definition of self-organization as the ability of a single set of Agents of operating on a broad range of Artifact configurations, that may change because of machine failures, order modifications or changes in stock. The versatility of Agents with respect to Artifacts—the condition to their self-organization—suggests new system architectures for self-organizing systems, in particular architectures based on the Web.

ACKNOWLEDGMENT

This work was funded by the German Federal Ministry of Education and Research through the MOSAIK project (grant no. 01IS18070A-E)

REFERENCES

[1] P. Leitão, J. Barbosa, and D. Trentesaux, "Bio-inspired multi-agent systems for reconfigurable manufacturing systems," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 5, pp. 934 – 944, 2012.

[2] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for Industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, pp. 18 – 23, 2015.

[3] Weiming Shen, Lihui Wang, and Qi Hao, "Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 36, no. 4, pp. 563–577, 2006.

[4] R. Frei, G. Di Marzo Serugendo, and J. Barata, "Designing self-organization for evolvable assembly systems," in *2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, 2008, pp. 97–106.

[5] J. Barbosa, P. Leitão, E. Adam, and D. Trentesaux, "Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution," *Computers in Industry*, vol. 66, pp. 99–111, 2015.

[6] P. Leitão, S. Karnouskos, L. Ribeiro, J. Lee, T. Strasser, and A. W. Colombo, "Smart agents in industrial cyber-physical systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1086–1101, 2016.

[7] V. Charpenay, "A bibliography on self-organization in manufacturing systems," Oct. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4067814

[8] Y. Sallez, T. Berger, and D. Trentesaux, "A stigmergic approach for dynamic routing of active products in FMS," *Computers in Industry*, vol. 60, no. 3, pp. 204 – 216, 2009, intelligent Products.

[9] G. Q. Huang, Y. Zhang, and P. Jiang, "RFID-based wireless manufacturing for real-time management of job shop WIP inventories," *The International Journal of Advanced Manufacturing Technology*, vol. 36, no. 7-8, pp. 752–764, 2008.

[10] H. A. El Maraghy, "Flexible and reconfigurable manufacturing systems paradigms," in *Flexible Services and Manufacturing Journal*, vol. 17, no. 4 Special Issue. Springer New York LLC, 2006, pp. 261–276.

[11] P. Leitão and F. Restivo, "ADACOR: A holonic architecture for agile and adaptive manufacturing control," *Computers in Industry*, vol. 57, no. 2, pp. 121 – 130, 2006.

[12] H. Van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, and P. Peeters, "Reference architecture for holonic manufacturing systems: PROSA," *Computers in Industry*, vol. 37, no. 3, pp. 255 – 274, 1998.

[13] J. Barata and L. M. Camarinha-Matos, "Coalitions of manufacturing components for shop floor agility-the CoBASA architecture," *International journal of networking and virtual organisations*, vol. 2, no. 1, pp. 50–77, 2003.

[14] K. Ueda, J. Vaario, and K. Ohkura, "Modelling of biological manufacturing systems for dynamic reconfiguration," *CIRP Annals*, vol. 46, no. 1, pp. 343 – 346, 1997.

[15] X. Xu, "From cloud computing to cloud manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 1, pp. 75 – 86, 2012.

[16] P. Jiang, K. Ding, and J. Leng, "Towards a cyber-physical-social-connected and service-oriented manufacturing paradigm: Social manufacturing," *Manufacturing Letters*, vol. 7, pp. 15 – 21, 2016.

[17] F. Tao, Y. LaiLi, L. Xu, and L. Zhang, "FC-PACO-RM: A parallel method for service composition optimal-selection in cloud manufacturing system," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2023–2033, 2013.

[18] P.-C. Chang, J.-C. Hieh, and T. W. Liao, "Evolving fuzzy rules for due-date assignment problem in semiconductor manufacturing factory," *Journal of Intelligent Manufacturing*, vol. 16, no. 4-5, pp. 549–557, 2005.

[19] T. Chen, "A hybrid SOM-BPN approach to lot output time prediction in a wafer fab," *Neural Processing Letters*, vol. 24, no. 3, pp. 271–288, 2006.

[20] Hadeli, P. Valckenaers, M. Kollingbaum, and H. Van Brussel, "Multi-agent coordination and control using stigmergy," *Computers in Industry*, vol. 53, no. 1, pp. 75 – 96, 2004.

[21] D. Kai, L. Jingyuan, C. T. S. Felix, H. Jizhuang, Z. Fuqiang, and W. Yan, "Hidden Markov model-based autonomous manufacturing task orchestration in smart shop floors," *Robotics and Computer-Integrated Manufacturing*, vol. 61, Feb 2020.

[22] F. Ocker, I. Kovalenko, K. Barton, D. Tilbury, and B. Vogel-Heuser, "A Framework for Automatic Initialization of Multi-Agent Production Systems Using Semantic Web Technologies," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4330–4337, Oct 2019.

[23] B. Caesar, F. Grigoleit, and S. Unverdorben, "(Self-)adaptiveness for manufacturing systems: challenges and approaches," *SICS Software-Intensive Cyber-Physical Systems*, vol. 34, no. 4, pp. 191–200, dec 2019.

[24] A. Ciortea, S. Mayer, and F. Michahelles, "Repurposing Manufacturing Lines on the Fly with Multi-agent Systems for the Web of Things," in *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems(AAMAS 2018)*, 2018.

[25] I. Kovalenko, D. Tilbury, and K. Barton, "The model-based product agent: A control oriented architecture for intelligent products in multi-agent manufacturing systems," *Control Engineering Practice*, vol. 86, pp. 105–117, MAY 2019.

[26] 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. 12 746–12 754, 2018.

[27] F.-S. Hsieh, "Design of scalable agent-based reconfigurable manufacturing systems with Petri nets," *International Journal of Computer Integrated Manufacturing*, vol. 31, no. 8, SI, pp. 748–759, 2018, 26th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM), Korea Sci & Technol Ctr, Seoul, South Korea, Jun 27-30, 2016.

[28] G. Skulj, A. Sluga, D. Bracun, P. Butala, and R. Vrabic, "Energy efficient communication based on self-organisation of IoT devices for material flow tracking," *CIRP Annals-Manufacturing Technology*, vol. 68, no. 1, pp. 495–498, 2019.

[29] S.-J. Shin, Y.-M. Kim, and P. Meilanitasari, "A Holonic-Based Self-Learning Mechanism for Energy-Predictive Planning in Machining Processes," *Processes*, vol. 7, no. 10, Oct 2019.

[30] A. Omicini, A. Ricci, and M. Viroli, "Artifacts in the A&A meta-model for multi-agent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 17, no. 3, pp. 432–456, 2008.

[31] M. Kovatsch, R. Matsukura, M. Lagally, T. Kawaguchi, K. Toumura, and K. Kajimoto, "Web of things (WoT) architecture," April 2020. [Online]. Available: https://www.w3.org/TR/wot-architecture/

[32] S. Kaebisch, T. Kamiya, M. McCool, V. Charpenay, and M. Kovatsch, "Web of things (WoT) thing description," April 2020. [Online]. Available: https://www.w3.org/TR/wot-thing-description/

AUTHORS

Victor Charpenay is Maître de conférence (associate professor) at Mines Saint-Étienne. He worked on MOSAIK while at the Friedrich-Alexander University (FAU) Erlangen-Nuremberg, after he obtained his Ph.D. at the University of Passau in 2019. His research topics include the Web of Things, agent-based cyber-physical systems, knowledge representation and knowledge exchange on the Web. He also is a contributor to the W3C Web of Things standards. E-Mail: `victor.charpenay@fau.de`.

Daniel Schraudner received his B.Sc. (2016) and M.Sc. (2019) in Computer Science from FAU Erlangen-Nuremberg. He currently is doing his Ph.D. at the Chair of Technical Information Systems at the same university. His research interests include multi-agent systems, Linked Data, the Semantic Web and knowledge representation. E-Mail: `daniel.schraudner@fau.de`.

Thomas Seidelmann is a researcher at the Otto-von-Guericke University (OvGU) Magdeburg. In 2019 he completed his M.Sc. in Computational Visualistics at the same university. His research is focused on multi-agent systems, especially in terms of mechanistic individual based simulations and swarm behavior. In his work towards an interdisciplinary Ph.D. thesis, Thomas further applies this knowledge to species coexistence theory. E-Mail: `thomas.seidelmann@ovgu.de`.

Torsten Spieldenner acquired his B.Sc. in Computer Science in 2009, and his M.Sc. in Visual Computing in 2012 at Saarland University in Saarbrücken, Germany. He is now working as a researcher at the German Research Center for Artificial Intelligence (DFKI) in Saarbrücken and pursues a Ph.D. in the field of Semantic Web and autonomous software in the Web of Things. E-Mail: `torsten.spieldenner@dfki.de`.

Jens Weise is a researcher at the OvGU Magdeburg. He received his B.Sc. degree in computer systems in engineering and his M.Sc. in medical systems engineering at OvGU Magdeburg, in 2013 and 2017 respectively. His current research includes many-objective pathfinding and route planning using evolutionary multi-objective optimization methods. He also conducts research on methods of self-organization in the field of factory planning. E-Mail: `jens.weise@ovgu.de`.

René Schubotz leads the Agents and Simulated Reality research team at DFKI, Saarbrücken. He received his diploma degree in Computer Science from Saarland University in 2007. His research concerns Semantic Web technologies and their application to robotics, simulated environments and classical AI. E-Mail: `rene.schubotz@dfki.de`.

Sanaz Mostaghim is a professor of computer science at OvGU Magdeburg, Germany. She holds a Ph.D. degree in Electrical Engineering and Computer Science from the University of Paderborn, Germany. She has worked as a postdoctoral fellow at ETH Zurich in Switzerland and as a lecturer at Karlsruhe Institute of technology (KIT), Germany. Her research interests are in the area of evolutionary multi-objective optimization and decision-making, swarm intelligence, and their applications in robotics, science and industry. Sanaz is as an associate editor for the *IEEE Transactions on Evolutionary Computation* and *IEEE Transactions on Artificial Intelligence*. E-Mail: `sanaz.mostaghim@ovgu.de`.

Andreas Harth currently holds the Chair of Technical Information Systems at FAU Erlangen-Nuremberg and also leads the department Data Spaces and Internet of Things Solutions at Fraunhofer SCS in Nuremberg. He previously worked at the institute AIFB at KIT until 2018. In 2010, he completed his Ph.D. at DERI at the National University Ireland, Galway. His research interest is centered around Linked Data and concerns large-scale interoperation in hyperlinked networked systems. E-Mail: `andreas.harth@fau.de`.