Towards employing ABM and MAS integrated with MBSE for the lifecycle of sCPSoS

Moharram Challenger*
University of Antwerp and Flanders Make, Belgium
moharram.challenger@uantwerpen.be

Hans Vangheluwe
University of Antwerp and Flanders Make, Belgium
hans.vangheluwe@uantwerpen.be

ABSTRACT

Cyber-Physical Systems (CPSs) are natural evolutions of embedded systems consisting of embedded computing devices and networks interacting with physical processes and possibly with a human. By introduction of Internet of Things (IoT), and Industry 4.0, CPSs are used in an interconnected way each of which may belong to a different stakeholder, building a complex System of Systems (SoS) called CPSoS. These systems need to employ some techniques to transform the collected data to knowledge with which the system can make better decisions. These capabilities can create smart CPSoS or sCPSoS.

However, these systems are highly complex from both structural and behavioural point of views. Naturally, there is a need for multiple abstraction levels using different paradigms to model these system, called Multi-Paradigm modeling (MPM).

In this paper, the challenges and opportunities of using agent technologies, including intelligent agents, Agent-based modeling and Simulation (ABM), and Multi-Agent Systems (MAS) (or Agent-oriented Software Engineering (AOSE)) in an integrated way with Model-based System Engineering (MBSE) techniques are discussed to cover the whole lifecycle of sCPSoS, from simulation and analysis, to development, operation and monitoring.

CCS CONCEPTS

• **Software and its engineering** → **Model-driven software engineering**; • **Computing methodologies** → **Multi-agent systems**; • **Computer systems organization** → **Embedded and cyber-physical systems**.

KEYWORDS

Agent-based modeling, Multi-Agent Systems, Agent-Oriented Software Engineering, Model-based System Engineering, Multi-Paradigm modeling, Smart Cyber-Physical System of Systems

ACM Reference Format:
Moharram Challenger and Hans Vangheluwe. 2020. Towards employing ABM and MAS integrated with MBSE for the lifecycle of sCPSoS. In *ACM/IEEE 23rd International Conference on Model Driven Engineering Languages and Systems (MODELS '20 Companion), October 18–23, 2020, Virtual Event, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3417990.3421439

*Corresponding author, Tel: +32 485 94 6229

MODELS '20 Companion, October 18–23, 2020, Virtual Event, Canada

ACM ISBN 978-1-4503-8135-2/20/10…$15.00
https://doi.org/10.1145/3417990.3421439

1 INTRODUCTION

Cyber-Physical Systems (CPSs) [21] consist of tightly integrated and coordinated computational and physical elements. They are evolutions of embedded systems to a higher level of complexity by focusing on interaction with highly uncertain environment (such as human interaction or wear and tear of devices). In these systems, embedded computers and networks monitor (through sensors) and control (through actuators) the physical processes, usually with feedback loops where physical processes and computations affect each other. The computational part of these systems plays a key role and needs to be developed in a way that can handle (mostly in real-time) the uncertain situations with the limited resources (including computational resource, memory resource, communication resource, etc.) [12].

By introducing Internet of Things [3, 18] and Industry 4.0 [17, 22], the CPSs are connected to each other with various networking approaches (e.g. Wireless Sensor Networking, Wi-Fi, Bluetooth, etc.) to meet the emerging more complex requirements. These interconnected CPSs constitute an even more complex system called Cyber-Physical System of Systems (CPSoSs) in which not only the environmental uncertainty and resource limitations are available, but also there may be emergent behaviour, lack of central control, dynamic structure, and need for autonomy.

According to Jamshidi [9, 16], a CPSoS is a CPS which exhibits the features of System of Systems (SoS) including:

- Large and spatially distributed physical systems with complex dynamics
- Distributed control, supervision and management
- Partial autonomy of the subsystems
- Dynamic reconfiguration of the overall system on different time-scales
- Possibility of emerging behaviours
- Continuous evolution of the overall system during its operation

To design and manage CPSoS, there is a need for theories and tools from different domains. The behaviour of the large coupled physical part of the system can be modelled, simulated and analysed, e.g. using agent-based large-scale simulation, stability analysis, and design of stabilizing control laws.

A key point in a CPSoS is to obtain knowledge out of the information, collected by monitoring the environment, using computational intelligence techniques. This knowledge can improve the control and feedback mechanism. This capability leads to the next generation of CPSoS with timely and more accurate decisions and actions called Smart CPSoS (sCPSoS). These smart systems [1] can analyze a situation and make decisions based on the available data in an adaptive manner, to perform smart actions. The smart-ness of these systems can be attributed to autonomous operation based on control process. Therefore, sCPSoSs are large-scale software intensive and pervasive systems, which by combining various data sources (both from physical and virtual elements), and applying intelligence techniques, are able to efficiently manage real-world processes and offer broad range of novel applications and services. In this way, self-managing, self-configuring, and ultimately self-aware systems can be built to improve quality of CPSoS, helping to address a number of social and environmental issues. However, such intelligent techniques put yet additional complexity to the systems, specifically to the computational part.

Thus, these systems of the future have a high complexity (both from structural and behavioural points of view) throughout their lifecycle, including modeling and simulation (M&S), design and implementation, validation and verification (V&V), deployment, execution and monitoring, and maintenance and evolution.

One of the approaches to address the complexity of engineering systems is to remove the extra details and have an abstract model/representation of the system where we can do some tasks (e.g. analyze, comprehend, develop, etc.) which are difficult or sometimes impossible to do on the original system. modeling a system represents the properties of interest in that system which can be used for different purposes. There can be different models with specific paradigms and formalisms for a complex system such as a sCPSoS in which each of the models represents one aspect of the system. This approach is called Multi-Paradigm Modeling (MPM) [2]. The modeling approach can be used for different purposes, such as model based engineering [34] which is a software and systems development paradigm that emphasizes the application of modeling principles and best practices throughout the System Development Life Cycle; and M&S in which the models are basis for simulation of the system to develop data utilized for analysis and decision making for the system.

This vision paper aims to elaborate the lifecycle of sCPSoS using MPM [2] (both for M&S and development), to use proper paradigms/formalisms with the philosophy of modeling different aspects/phases of sCPSoS lifecycle explicitly in an appropriate level of abstraction (in design and/or run time). The idea of using different paradigms and formalisms in the lifecycle of sCPSoS can be challenging, as the integration of different paradigms may be a hard problem, if it is not possible at all in some cases. More details on the fundamentals of this research including research challenges are given in the sections.

How we can build the foundations to use the advantages of Model-driven Engineering (MDE), Multi-Agent System (MAS), and Agent-based Modeling (ABM) for CPS? The scientific research gaps/questions and future industrial needs related to this field. This awareness is the result of working and collaborating with researchers related to this field and also by studying the industrial reports (i.e. Gartner's reports for hype cycle and technical trends which expect emerging technologies such as Internet of Autonomous Things and Digital Twin). Finally, the results of conducting research on sCPSoS can be applied in different domains, such as Automotive platooning or flocking, Smart Manufacturing (Cyber-Physical Production Systems), etc..

The rest of the paper is organised as follows: Section 2 discusses the related work and related academic activities. The conceptual architecture for the integration of agent-based technologies with MBSE techniques is presented in Section 3. Then, in Section 4, the challenges and research questions in this integration for sCPSoS are elaborated. Finally, the paper is concluded in Section 5.

2 RELATED WORK

There were some recent academic events and activities on the multi-Paradigm modeling of cyber-Physical systems (MPM4CPS), such as the COST Action 1404 MPM4CPS [35] and the International workshop for MPM4CPS held as part of MODELS conference. Some studies have been reported related to using agent technologies along with MBSE for CPS. For instance, in [11], the authors discuss challenges of automating adaptive abstraction in the agent-based modeling of car traffic. They describe the adaptive abstraction process in terms of a MAPE-K (Monitor-Analyze-Plan-Execute over a shared Knowledge) control loop, particularly, adaptively selecting a candidate over multiple abstractions.

As another study, the work presented in [7] discusses how an agent development framework called SEA_ML++ is used for the design and implementation of an agent-based garbage collection system (as a cyber-Physical system). MDE techniques are used in SEA_ML++ to generate part of the system software. The conducted study demonstrates how this CPS can be designed according to the various viewpoints of SEA_ML++ and then implemented on JASON agent execution platform.

The challenges of using MASs in real-time systems to enable IoT and CPS are discussed in [6]. The authors define the contributions of MAS for CPS and IoT systems, see Figure 1. Internet of Things is a network-oriented system focusing on heterogeneous devices uniquely addressable. Therefore, it can be integrated into a huge and ever growing network wrapped by several frameworks interacting with the related communication standards. CPS implicitly involve features such as adaptability, scalability, safety, and resilience. Therefore, they stimulated huge interests producing considerable contributions from both artificial intelligence and embedded systems communities. Finally, according to the authors' report, the following features of MAS support their adoption in CPS and IoT systems: Scalability, reactivity, sociability, autonomy, modularity, context awareness, flexibility, reconfigurability etc. 1.

Furthermore, considering the other related work, in "Road2CPS EU support Action" [14], a roadmap and recommendations for future deployment of CPSs are proposed [29]. Similarly in "CPSoS EU Support Action" [10], the challenges posed by engineering and operation of CPSoS are defined in [9, 33] and a research and innovation agenda on CPSoS are presented. Also, in [19, 23, 28], a group of researchers (in the scope of an international research project [15]) have studied the application of agents to address the problems in Cyber-Physical Production Systems. They focused on quality control in manufacturing by proposing an agent architecture (which presents a distributed intelligence) for distributed analysis of the CPS and tackling the defects in multi-stage manufacturing. However, none of these studies and research projects

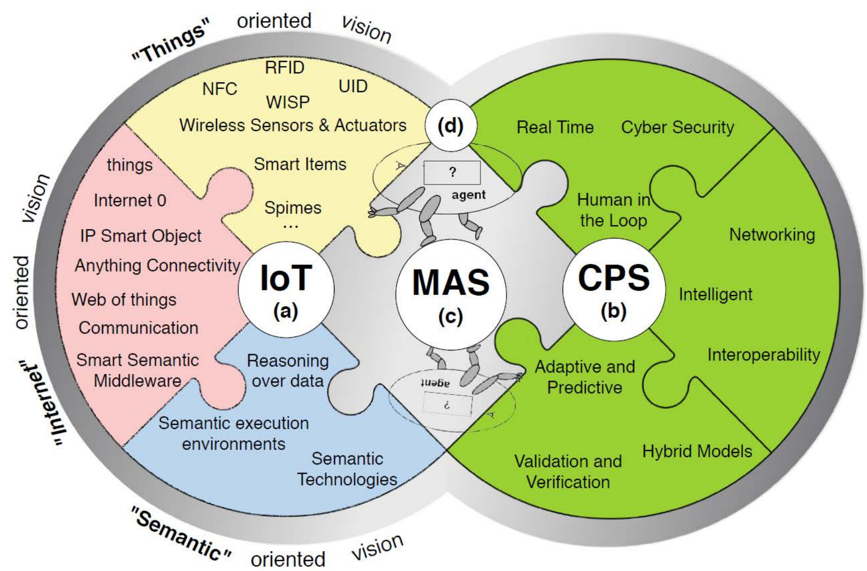

Figure 1: Relation between MAS, CPS, and IoT [6]

address the different levels of agent-based sCPSoS throughout its lifecycle considering M&S, design and development, and execution and monitoring. There are many fundamental research questions which need to be addressed to make the sCPSoS available for the engineers, some of which are discussed in the scope of this paper.

3 CONCEPTUAL ARCHITECTURE

To imagine how the agent technologies, as ABSE, (including Intelligent Agent (IA) [32], MAS [20], ABM [11] and AOSE [25]) can be used along with the MBSE techniques, we present a conceptual architecture for their possible orthogonal integration.

First of all, let us define what are the things that agent technologies can do in general. An IA refers to an autonomous entity which acts, directing its activity towards achieving goals, upon an environment using observation through sensors and consequent actuators. Intelligent agents may also learn or use knowledge to achieve their goals [31]. A MAS is a system including multiple interacting IAs which can solve difficult problems that individual agents or monolithic systems cannot solve. Intelligence may include methodic, functional, procedural approaches, algorithmic search or reinforcement learning [4]. An agent-based model is a model for simulating the actions/interactions of autonomous agents (both individual or collective entities such as organizations or groups) with a view to assessing their effects on the system as a whole [13, 27]. Finally, AOSE is a software engineering paradigm that apply best practice in the development of complex systems by focusing on the use of agents, and organizations of agents as the main abstractions [36].

In general, the MBSE techniques can be used in an orthogonal way with Agent-based system engineering approaches, see Figure 2.

As one can see in Figure 2, to orthogonally combine MBSE and ABSE, the CPS designers need to first decide about the formalisms, the languages (with their syntax and semantics suitable for the application domain), workflow/process [8], the architecture, the views, and abstractions which can be used in the development of the system. The second choice is the for ABSE technologies including mobile/distributed agents, intelligent agents, MASs, agent-oriented system engineering and/or ABM. Finally, there is a need to combine the selected MBSE techniques with chosen ABSE technologies.

One approach to do this, in the formalism level, is embedding the physical models in the agents. This means that some of the agents are going to use the models of physics and represent the behaviour of physics, for example a flow-based physical model like Bond graph [5].

Although, it is OK to encapsulate the physical systems or their models in agents (as if these physical system were agents), we also need to consider some physical characteristics as well, such as aggregation of the models of physics (as it happens in the physical phenomena). For example, once there is a connection between the agents for two Bond graph models, one need to make them as a one big Bond graph. This needs for a dynamic abstraction change (increasing the abstraction by aggregation). There are some studies for adaptive abstraction in ABM [11] which can be used as the starting point for this purpose.

4 CHALLENGES AND RESEARCH QUESTIONS

As it is discussed earlier, the idea is to improve the lifecycle of sCPSoS by using MPM. To this end, different layers with different abstraction levels are considered for these complex systems. This can be used in the analyzing phase (by M&S), development phase (by step-wise refinement of design models) or at the operation phase (with adaptive abstraction and/or monitoring). For this purpose the foundations and challenges of integrating agent-based system engineering with model-based system engineering are addressed.

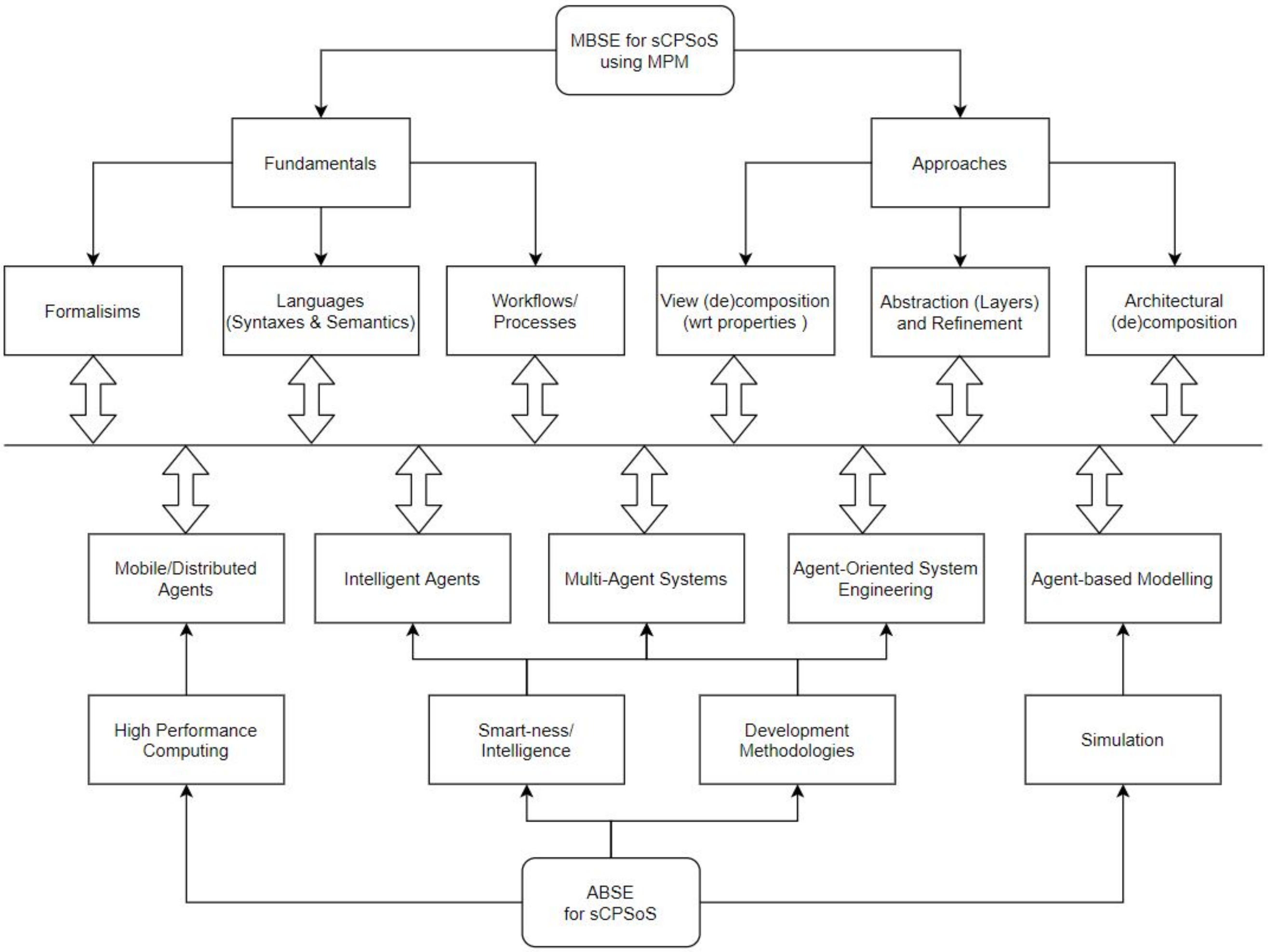

Figure 2: Architecture for orthogonal integration of MBSE and ABSE

A sCPSoS is a SoS interconnecting different CPSs each of which has a computational part interacting with physical elements/processes. These systems are working in uncertain environments, have resource limitations, may have no central control, and need to present smart behaviour, all of which make the system and its lifecycle very complex. Components of a sCPSoS must have a high degree of autonomy while cooperating with each other in a robust, scalable and decentralized way. These components and their main challenges are summarized in Table 1. Methods and tools from computer science should be adopted for: modeling of these systems, verification and testing, assume-guarantee methods, contract-based assertions etc. This is to capture both the behaviour on the low level (discrete control logic, communication, effects of distributed computing) and global effects, in the latter case based on abstract models of complete subsystems.

These challenges involve in the engineering of systems of systems over its lifetime, coordination and optimization of autonomously managed subsystems, modeling, simulation and model management, tools and methods for validation and verification on the system level, humans in the loop, and systems integration [29].

Agent-based system engineering (including intelligent agents, MASs, ABM, and agent oriented system engineering) can be used to address some of the challenges in lifecycle of sCPSoS.

According to Russell and Norvig [31], an agent is anything that can be considered to be able to perceive its environment through sensors and act on this environment through actuators. Also, based on Macal definition [24], an agent needs to have the following features: (1) identifiable discrete individuals with rules that can govern behaviour and decision-making capacity (Intelligent); (2) Interaction with environment with other agents (reactive and social); (3) goal-driven (proactive); (4) self-contained; and (5) ability to adapt its behaviour through time-based experiences (learn) [30].

A MAS is defined as a loosely coupled network of problem-solving entities (agents) that work together to find answers to problems that are beyond the individual capabilities or knowledge of each entity (agent). With the MAS characteristics such as autonomy,

Table 1: sCPSoS Challenges and possible solutions from Agent-based System Engineering

Component/Aspect	Challenge/Limitation	Agent-based System Engineering
Cyber	Resource Constraints	Multi-level Organization
Physical	Environmental uncertainty Physical constraints (validity)	Adaptive behaviour (abstraction)
System of Systems	No central control Autonomy of subsystems Dynamic structure and behaviour Emerging behaviours	Distributed and Mobile Autonomous Dynamic ABM and Monitoring
Smart	Intelligence techniques Explainable and interpretable	Autonomous XMAS, Goal-directed reasoning

sociability, reactivity and proactivity, these systems can used along with MBSE to target the challenges of sCPSoS such as smart-ness, uncertainty, and distribution (Figure 1). In this sense, the agent paradigm can be integrated with MBSE paradigm to also address the abstraction and adoption (as well as adaptive abstraction) for sCPSoS.

Agents can be used in ABM, which states a computer-based method for studying the (inter)actions of a set of autonomous entities. These systems refer to individual-based model or a self-sufficient computing method. The objective of an ABM is to search for descriptive insights into the agents' (not necessarily intelligent) collective behaviour following simple rules (typical of natural systems) rather than solving particular engineering problems (as it is in MAS) [27]. Apparently, ABM is very suitable for analyzing and designing complex SoS such as sCPSoS.

The idea is to include both ABM and MAS in the process of developing sCPSoS to address M&S as well as implementation and monitoring the sCPSoS. To this end, MBSE techniques can be used to bring the gap between the M&S with ABM and implementation with MAS together. The formal specification of both ABM agents and MAS agents need to be provided and based on the commonality and variability analysis, a MBSE process needs to be defined for the development of sCPSoS. This can provide a system level/high level consideration in the sCPSoS lifecycle which is very crucial for complex systems. Therefore, this paper follows the MPM approach to study the foundations of integrating Agent- and Model- based paradigms to address some of the sCPSoS problems and challenges.

In addition to addressing the challenges, involving agents in the process of design, development, implementation, and execution of sCPSoS can bring the following advantages for these systems:

- **Collaborative Intelligence and Agent-based Computing:** Multi-agent distributed systems with a cognitive architecture such as goal-directing reasoning empowers the MAS so that each agent (or its representing machine) uniquely presents autonomy to contribute to the problem-solving mechanism.
- **Pervasive and Ubiquitous Intelligence:** MAS can help to have ubiquitous intelligence where intelligence (provided by agents collaborations) is made to appear anytime and in any agent (any CPS).
- **Pervasive Simulation:** Some of the agents in the proposed architecture can, as part of its tasks, simulate a part of the system M&S (for example as part of co-simulation).
- **Explainable and interpretable intelligence (Explainable MAS and Agency):** Agents and MASs can have contributions for eXplainable Artificial Intelligent (XAI). Explainable MAS and explainable agency have recently gained attention of the researchers, such as EMAS Workshop (organized as part of AAMAS Conference). This is an essential characteristic for intelligent agents in sCPSoS, as the CPSs involved in these systems mostly handle critical tasks and their actions need to be explainable and interpretable. This is not only important in the human-agent interaction, but also in the agent teams, as succeeding in agents collaboration necessitates a mutual understanding of the status of other agents including their capacities and limitations.
- **Proactive behaviour for sCPSoS reactive systems:** Using goal-directing behaviour in the intelligent agents, the agents and their representing machine/element can follow its goals and take some proactive actions in addition to the classic reactive actions. This is the base for internal reasoning of the agent.

To benefit from the above mentioned advantages some fundamental challenges need to be addressed, some of which are discussed here:

- **Formulating the agents for sCPSoS lifecycle:** This is to come up with a correlational (a-causal) formulation/language with core elements of ABM and MASs to represent the structure and behaviour of a dynamic system (i.e. sCPSoS). These elements need to address both the design-time and run-time aspects of the system. In this way, the the formulation/language can support both M&S as well as MAS-based development and execution of the system.
- **Simulation and development techniques for agent-based sCPSoS:** Considering the agents' capabilities (i.e. sociability, autonomy, reactivity, and pro-activity), it is possible to provide an ABM and development and execution frameworks for complex, autonomous and dynamic-structured systems such as sCPSoS (these techniques can be also used for SoS,

Internet of Things (IoT), Internet of Autonomous Things (IoAT), Industry 4.0, and Digital Twin (DT).

- **Agent- and Model- based sCPSoS architecture and development Process:** Covering the whole sCPSoS lifecycle with different levels of abstraction (multi-layer architecture). In sCPSoS, the components are modified, added, the scope of the system may be extended or its specifications changed. So, engineering to a large extent has to be performed in a more dynamic manner. The traditional development paradigms, such as waterfall, is not applicable in its pure form to systems of systems where the requirements change during operation. Therefore, there is a need for a scientific foundation to handle multi-layer operations and life cycle management.
- **Adaptive Abstraction for sCPSoS M&S and operation:** When dealing with complex systems such as sCPSoS, there usually exist multiple abstractions, typically describing partially overlapping details of the system under study, and resulting in a hierarchy of abstractions. Adaptive abstraction leverages these levels with the aim of dynamically adapting the abstractions used during system simulation and execution. With a run-time symbolic manipulation mechanism and most importantly causality assignment for a modeling language such as Modelica, it is possible to support agents' complex behaviour which models the CPS. This capability is utter important because the dynamic structure (such as creation/deletion) of agents is very frequent, unpredictable, and there are too many variants to pre-compute all possible causality assignments.
- **MPM4sCPSoS: Fundamentals for Integration of Agent paradigm with MBSE:** The engineering of sCPSoS requires methods and frameworks that can be used seamlessly during design as well as operation (design-operations continuum). This needs to study the fundamentals of integrating the paradigm which is currently used in the development of these systems with the agent paradigm. This will pave the way to benefit the advantages of the proposed approach with the benefits of current state of the art and technology.
- **Modeling Framework and Automatic Synthesis (MBSE to ABSE and ABM to MAS transformations):** The proposed framework can be mapped from MPM language(s) used as part of MBSE such as Modelica and Simulink. This helps to model the multi-physics aspects of a sCPSoS (such as electrical, mechanical, hydraulic, biochemical, etc.) from one hand and computational aspects (control, signal processing, scheduling, etc.) from another in a single framework. In addition, the simulation models in ABM can be transformed to implementation and execution models in MAS using MDE transformation techniques. This framework can also be combined (using modular language engineering) with rule-based (graph) transformation to describe structural changes, e.g. using ProMoBox [26], to engineer the (temporal) pattern languages.

Some of the specific research questions could be:

(1) How intelligent agents (as computationally intensive software components) can be deployed on cyber-Physical systems with their resource constraints (processing or memory) and hard/soft real-time deadlines?
(2) How to seamlessly integrate ABM and MAS with the other paradigms and formalisms (and their supporting models, languages, frameworks and tools) used to engineer sCPSoS?
(3) Which agent architectures are most suitable for different layers and application domains of sCPSoS?
(4) How to use agent-based systems to deal with dynamic systems such as sCPSoS, for example in the operating environment or with human in the loop?
(5) How to scale with the complexity of real-world sCPSoS?
(6) How can ABM and MAS improve development of sCPSoS (and its applications in IoT and Digital Twins)? Which development tools and frameworks are needed and how this process can be automated?
(7) Which processes and methodologies can integrate the agents and sCPSoS and provide a disciplined approach to rapid yet high-quality developmentment?
(8) How the formal semantics of ABM and MAS for development/monitoring of sCPSoS can be specified?
(9) What are the logical systems for modeling, specification, analysis and synthesis of agents for sCPSoS
(10) What are the mechanism to automate the adaptive abstraction of ABM and MAS for sCPSoS in an efficient way?
(11) How goal-directing behaviour can provide an explainable, transparent and understandable intelligent agents?
(12) How can intelligent agents help in the development of self-aware explainable smart sCPSoS in which the proactive actions can be explained and justified?

5 CONCLUSION

Agents and MASs with their different types and capabilities can be used to address the development of complex systems such as sCPSoS throughout their lifecycles. This paper discusses the challenges, research questions, and opportunities on employing agent technologies for CPS and sCPSoS. These challenges and research questions need to be addressed by researchers to pave the way for using agent to develop intelligent systems in an efficient way.

As the next steps, we plan to work on implementing a CPS using different types of agent in different layers of the system, including controller, edge, and cloud.

ACKNOWLEDGMENTS

The authors would like to acknowledge Flanders Make, a strategic center for the manufacturing industry in Flanders, Belgium.

REFERENCES

[1] Georges Akhras. 2000. Smart materials and smart systems for the future. *Canadian Military Journal* 1, 3 (2000), 25–31.

[2] Moussa Amrani, Dominique Blouin, Robert Heinrich, Arend Rensink, Hans Vangheluwe, and Andreas Wortmann. 2019. Towards a Formal Specification of Multi-Paradigm Modelling. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. IEEE, 419–424.

[3] Luigi Atzori, Antonio Iera, and Giacomo Morabito. 2010. The internet of things: A survey. *Computer networks* 54, 15 (2010), 2787–2805.

[4] Mihal Badjonski, Mirjana Ivanović, and Zoran Budimac. 1999. Agent oriented programming language LASS. In *Object-oriented technology and computing systems re-engineering*. Horwood Publishing, Ltd., 111–121.

[5] Jan F Broenink. 2020. Bond Graphs: A Unifying Framework for Modelling of Physical Systems. In *Foundations of Multi-Paradigm Modelling for Cyber-Physical Systems*. Springer, 15–44.

[6] Davide Calvaresi, Mauro Marinoni, Arnon Sturm, Michael Schumacher, and Giorgio Buttazzo. 2017. The challenge of real-time multi-agent systems for enabling IoT and CPS. In *Proceedings of the international conference on web intelligence*. 356–364.

[7] Moharram Challenger, Baris Tekin Tezel, Vasco Amaral, Miguel Goulao, and Geylani Kardas. 2020 (in press). Agent-based Cyber-physical System Development with SEA_ML++. In *Multi-Paradigm Modelling Approaches for Cyber-Physical*, V. Amaral B. Tekinerdogan and H. Vangheluwe (Eds.). Elsevier Publication, Chapter 8, 1–28.

[8] Moharram Challenger, Ken Vanherpen, Joachim Denil, and Hans Vangheluwe. 2020. FTG+ PM: Describing Engineering Processes in Multi-Paradigm Modelling. In *Foundations of Multi-Paradigm Modelling for Cyber-Physical Systems*. Springer, 259–271.

[9] Sebastian Engell, Radoslav Paulen, Michel A Reniers, Christian Sonntag, and Haydn Thompson. 2015. Core research and innovation areas in cyber-physical systems of systems. In *International Workshop on Design, Modeling, and Evaluation of Cyber Physical Systems*. Springer, 40–55.

[10] EU FP7. 2014-2016. CPSoS - Towards a European Roadmap on Research and Innovation in Engineering and Management of Cyber-Physical Systems of Systems. https://www.cpsos.eu/

[11] Romain Franceschini, Moharram Challenger, Antonio Cicchetti, Joachim Denil, and Hans Vangheluwe. 2019. Challenges for Automation in Adaptive Abstraction. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. IEEE, 443–448.

[12] Richard Fujimoto, Conrad Bock, Wei Chen, Ernest Page, and Jitesh H Panchal. 2017. *Research challenges in modeling and simulation for engineering complex systems*. Springer.

[13] Volker Grimm and Steven F Railsback. 2005. *Individual-based modeling and ecology*. Vol. 8. Princeton university press.

[14] EU H2020. 2015-2017. Road2CPS - Strategic action for future CPS through roadmaps, impact multiplication and constituency building. http://www.road2cps.eu/

[15] EU H2020. 2016-2019. Agent oriented zero defect multi-stage manufacturing. http://go0dman-project.eu/

[16] Mo Jamshidi. 2008. *Systems of systems engineering: principles and applications*. CRC press.

[17] Nasser Jazdi. 2014. Cyber physical systems in the context of Industry 4.0. In *2014 IEEE international conference on automation, quality and testing, robotics*. IEEE, 1–4.

[18] Sabina Jeschke, Christian Brecher, Tobias Meisen, Denis Özdemir, and Tim Eschert. 2017. Industrial internet of things and cyber manufacturing systems. In *Industrial Internet of Things*. Springer, 3–19.

[19] Jonas Queiroz. and Paulo Leitão. and José Barbosa. and Eugénio Oliveira. 2019. Distributing Intelligence among Cloud, Fog and Edge in Industrial Cyber-physical Systems. In *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO,*. INSTICC, SciTePress, 447–454. https://doi.org/10.5220/0007979404470454

[20] Geylani Kardas, Zekai Demirezen, and Moharram Challenger. 2010. Towards a DSML for semantic web enabled multi-agent systems. In *Proceedings of the International Workshop on Formalization of Modeling Languages*. 1–5.

[21] Edward A Lee. 2008. Cyber physical systems: Design challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. IEEE, 363–369.

[22] Jay Lee, Behrad Bagheri, and Hung-An Kao. 2015. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing letters* 3 (2015), 18–23.

[23] Paulo Leitão, José Barbosa, Carla AS Geraldes, and João P Coelho. 2018. Multi-agent System Architecture for Zero Defect Multi-stage Manufacturing. In *Service Orientation in Holonic and Multi-Agent Manufacturing*. Springer, 13–26.

[24] Charles M Macal. 2016. Everything you need to know about agent-based modelling and simulation. *Journal of Simulation* 10, 2 (2016), 144–156.

[25] Viviana Mascardi, Danny Weyns, Alessandro Ricci, Clara Benac Earle, Arthur Casals, Moharram Challenger, Amit Chopra, Andrei Ciortea, Louise A Dennis, Álvaro Fernández Díaz, et al. 2019. Engineering Multi-Agent Systems: State of Affairs and the Road Ahead. *ACM SIGSOFT Software Engineering Notes* 44, 1 (2019), 18–28.

[26] Bart Meyers, Romuald Deshayes, Levi Lucio, Eugene Syriani, Hans Vangheluwe, and Manuel Wimmer. 2014. ProMoBox: A Framework for Generating Domain-Specific Property Languages. In *Software Language Engineering*, Benoît Combemale, David J. Pearce, Olivier Barais, and Jurgen J. Vinju (Eds.). Springer International Publishing, Cham, 1–20.

[27] Muaz Niazi and Amir Hussain. 2011. Agent-based computing from multi-agent systems to agent-based models: a visual survey. *Scientometrics* 89, 2 (2011), 479.

[28] Jonas Queiroz, Paulo Leitão, José Barbosa, and Eugénio Oliveira. 2019. Agent-Based Approach for Decentralized Data Analysis in Industrial Cyber-Physical Systems. In *International Conference on Industrial Applications of Holonic and Multi-Agent Systems*. Springer, 130–144.

[29] Meike Reimann, Carsten Ruckriegel, Sarah Mortimer, Steve Bageritz, Michael Henshaw, Carys E Siemieniuch, Murray A Sinclair, Paul J Palmer, John Fitzgerald, Claire Ingram, et al. 2017. *Road2CPS priorities and recommendations for research and innovation in cyber-physical systems*. © Steinbeis-edition.

[30] Jorge Rocha, Inês Boavida-Portugal, and Eduardo Gomes. 2017. Introductory Chapter: Multi-Agent Systems. In *Multi-Agent Systems*. IntechOpen.

[31] Russell Stuart and Norvig Peter. 2016. *Artificial intelligence-a modern approach 3rd ed.* Berkeley.

[32] Baris Tekin Tezel, Moharram Challenger, and Geylani Kardas. 2016. A metamodel for Jason BDI agents. In *5th Symposium on Languages, Applications and Technologies (SLATE'16)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[33] H Thompson, R Paulen, M Reniers, C Sonntag, and S Engell. 2015. Analysis of the state-of-the-art and future challenges in cyber-physical systems of systems. *EC FP7 project* 611115 (2015).

[34] Bert Van Acker, Joachim Denil, Hans Vangheluwe, and Paul De Meulenaere. 2015. Managing heterogeneity in model-based systems engineering of cyber-physical systems. In *2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*. IEEE, 617–622.

[35] Hans Vangheluwe, Vasco Amaral, Holger Giese, Jan Broenink, Bernhard Schätz, Alexander Norta, Paulo Carreira, Ivan Lukovic, Tanja Mayerhofer, Manuel Wimmer, et al. 2016. MPM4CPS: multi-paradigm modelling for Cyber-Physical Systems. In *CEUR workshop proceedings*, Vol. 1675. 40–47.

[36] Gerhard Weiss. 2013. *Multiagent Systems, 2nd edition*. MIT Press.