Summer 6-23-2023

Requirements for Generating Learning Environments for Autonomous Systems Behavior in a Digital Continuum

Anike Murrenhoff
Fraunhofer Institute for Material Flow and Logistics, anike.murrenhoff@iml.fraunhofer.de

Requirements for Generating Learning Environments for Autonomous Systems Behavior in a Digital Continuum

Anike Murrenhoff
Fraunhofer-Institute for Material Flow and Logistics IML
Dortmund, Germany
anike.murrenhoff@iml.fraunhofer.de

***Abstract*— Autonomous systems in material handling are increasingly prevalent in logistics, offering benefits such as flexibility, adaptability, robustness, and sustainability. To fully harness these advantages, a novel paradigm, the Digital Continuum, is proposed for the development and operation of such systems. A critical component of the Digital Continuum is a deeply integrated digital system model, which serves as a simulation, training, and test environment for virtual agents corresponding to physical robots. To ensure robust performance in learned behavior, a large number of learning environments is needed, thus highlighting the importance of an automated generation process. This process can significantly reduce modeling effort and is yet to be developed. This paper presents the derivation of requirements for an automated learning environment generation approach, unifying elements from Digital Continua, intralogistics, and robotics domains. Furthermore, the paper briefly discusses the research gap in the context of existing procedural content generation and domain randomization approaches. By addressing these requirements and bridging the research gap, a generation approach has the potential to profoundly facilitate the development and operation of autonomous systems in logistics.**

Keywords—autonomous systems, Digital Continuum, intralogistics, learning environments, simulation-based artificial intelligence, procedural generation.

I. Motivation and Background

Over the past few decades, there has been a trend towards increasing autonomy in logistics systems, driven by rapid technological advances in Industry 4.0 [1]. Especially recent advances in artificial intelligence (AI) in combination with physically correct simulation environments have fueled new possibilities for autonomous decision-making in logistics.

The fundamental characteristic of autonomous systems is that they can achieve a given goal and independently adapt to a given, unknown situation accordingly, without human control or detailed programming [2]. In general, there are two types of autonomous systems: virtual and physical [3]. Virtual autonomous systems operate solely in the virtual realm, such as the internet. They are capable of taking autonomous action, such as defending against cyber-attacks or negotiating contracts between companies. Physical autonomous systems, on the other hand, have a tangible impact on the physical world, either as individual systems or as part of a networked cyber-physical system (CPS). Examples include collaborative robots, drones, and surgical robots.

Both types of autonomous systems have multiple benefits in logistics systems that often operate under demand fluctuations and uncertainties. Their independent decision-making capabilities enable autonomous systems to adapt to changes in supply networks independently and to scale according to the current demand. This makes these systems flexible, robust, and sustainable, as they can adjust themselves to the current and possible future circumstances. A very apparent instance of this trend is the evolution of automated guided vehicles (AGV) into autonomous mobile robots (AMR) in intralogistics. Where AGV depend on centralized planning and control systems and operate on predefined paths, AMR have the potential to enable self-contained, decentralized planning, execution, control, and optimization of internal material and information flows [4].

The development towards autonomy forces a paradigm shift from centralized, hierarchical organizations towards networked and autonomous systems which presents several challenges. Firstly, decentralized decision-making in general requires new methods and concepts to find appropriate overall system behavior. Due to the stochastic dependencies of the independent interaction of AMR with each other and with their environment, it is hardly possible to estimate the behavior of the system using analytical methods, especially if their behavior adapts depending on the situation [1]. Secondly, AMR systems that already support decentralized decision-making in practice oftentimes are not as autonomous as they could be. Their autonomy is limited to specific systems or tasks, and the respective planning and control decisions are hard coded into the decentralized decision logic of the systems [5]. This leads to the fact that the control software is hard to synchronize to changes in the physical system and adaptions always involve human intervention, which prohibits AMR from fulfilling the entirety of their potential benefits. Overall, new methods and concepts are required to find appropriate and adaptable system behavior through decentralized decision-making.

A technology supporting this evolution is simulation-based AI. AI techniques allow to solve multi-objective optimization and to include complex dependencies, which is required when learning to find good overall system behavior through decentralized decision-making [4]. Combined with simulation, AI techniques can be used to train autonomous agents in digital system models whose behavior can be transferred to the physical world. On a robotic level, there have been numerous studies that have shown how autonomous agents in a simulation can learn behavior that can be transferred to the real world (see [6]). A major advancement for this field has been the emergence of deep reinforcement learning (RL). However, current approaches of deep RL in robotics are still limited to lab settings, and far from being able to train AMR that can deal with the complexity and diversity of tasks and environments in the real world [7].

A possible solution to this challenge could be provided by the industrial metaverse. The industrial metaverse is a concept that involves creating digital representations of reality, which allow for interactions and decision-making with the power to impact the physical world, thereby merging physical and digital reality. However, most current use cases of the industrial metaverse are focused on the interaction of humans with virtual models and with each other in the virtual reality [8] and lack solutions for the specific requirements of AI-based autonomous systems, especially regarding continuously adapting and improving autonomously made decisions, while ensuring conformity and alignment with the intentions of the system designers. Developers of generative AI, such as OpenAI, are currently addressing this issue of the alignment problem in AI for their future releases, in this case of models for ChatGPT [9]. A potential solution is seen in deploying continuously improved models incrementally, instead of large new releases. This approach follows a continuous feedback-deployment loop between physical and digital reality, as in the DevOps principle.

An open question is, how the principles of the metaverse and ensuring alignment of AI-based decisions can be applied to developing and operating autonomous systems in a way that allows to reach the full potential of autonomy in logistics systems. This paper introduces the paradigm of a Digital Continuum (DC) for logistics to address this question.

The DC for logistics is based on the idea of an increasingly close connection between the physical and digital reality, forming a control loop between these worlds that incorporates AI methods to continuously optimize and adapt the systems behavior. The result is high-frequency logistics where decisions and transactions are made autonomously and in real time. In this context, digital systems models are increasingly evolving from representations to drivers of physical processes. They are an essential prerequisite to serve as digital twins and, above all, as learning and test environments for autonomous systems, such as AMR in intralogistics. By learning beneficial behavior in the digital reality, the learning process can be sped up, parallelized and new behavior tested without damage in the physical world. Achieving robust results in these digital training environments requires a large number of different learning and test environments to cover a wide variety of real-world circumstances, before the trained models can be applied to the physical world and improve the behavior of autonomous systems there.

Digital learning environments are expensive to model manually. For DC to become a reality and for AMR to reach their full potential, automatically generating learning and test environments for autonomous systems becomes inevitable. In the field of procedural content generation (PCG) and domain randomization (DR) approaches for automatically generating varying digital content have been developed for several decades, yet until now there is no approach to generate learning environments for autonomous systems that fulfills all the requirements from DC, logistics and robotics.

In conclusion, autonomous systems have the potential to revolutionize logistics and supply chain management, but further research and development are needed around paradigms to develop and operate these systems to allow complete autonomy and how the required digital learning environments for these systems can be generated automatically.

In the following sections, the concept of DC is introduced in more detail and the need for different learning and testing environments for autonomous systems identified. The requirements for an approach to generate these environments for AMR in intralogistics, as an example of autonomous systems in logistics, are then derived and compared at a high level with existing approaches of PCG in the following section. Finally, the conclusion summarizes the need for further research.

II. Digital Continua in Logistics

The paradigm of the DC in logistics describes how through the seamless connection of the physical and the digital world as well as development and operations, and through leveraging advanced methods in AI, it becomes possible for autonomous systems in logistics to reach new levels of autonomy.

A. *Concept of a Digital Continuum*

A DC emerges when physical and digital reality interact with each other and form an unbroken control loop that incorporates methods of AI to continuously optimize itself (see Fig. 1) [10].

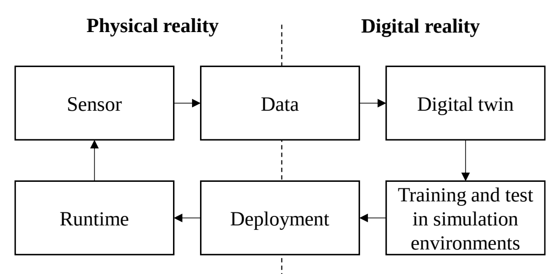

Fig. 1. Digital Continuum through the seamless merging of the physical and digital worlds.

Information on the physical environment is transmitted to a digital system model, that serves as the digital twin. By ensuring that the digital model captures the physical model as close to reality as necessary for the use case, the model can be seen as a digital reality. Within this digital reality, simulation and training runs can be conducted and improved system behavior can be determined. After testing the behavior in the digital reality and ensuring the alignment with the system designers' intentions, the adjusted code encompassing the behavior, e.g., weights of neural networks, can be delivered to the physical reality. The impact of the newly deployed control logic in the physical system is again perceived by sensors and an update is given to the digital world. A continuous highly complex and dynamic feedback loop emerges allowing for the system to find increasingly good behavior and increasingly accurate digital system models.

This process allows for autonomous systems to incorporate more and more complex decisions, until they reach a new level of autonomy. The crucial decisions for a systems behavior and reactions to changes are made in the digital world, therefore elevating digital twins to digital reality and digital models to drivers of physical processes.

The DC comprises all scopes of logistics use cases, from physical hardware and near-hardware applications, that are combined with digital development, to strategic decisions of real and digital supply chains, connecting the planning, tender, dispatching, and operations elements of supply chain management in the real world with their digital counterparts.

B. *Characteristics of Digital Continua*

Independent of the area of application or use case, every DC exhibits the same characteristics, potentially in different manifestations.

1) *Computational continuity*

The computational continuity property deals with the necessary architecture of computing capacities for DC. In order for DC to emerge, a continuous data processing infrastructure must be established between the physical and virtual system through distributed computing power.

Computational continuity is a key aspect of what has previously been defined as a "digital continuum" in the high performance computing community (see [11]). However, the DC in logistics presented in this paper includes additional aspects, that are described in the following characteristics.

Autonomous entities at the IoT device level present in the physical system are equipped with data processing capacities, as they are able to independently convert input signals from the perceived environment into appropriate actions and send data from the perceived environment to their digital counterpart. Before arriving at the data centers where the digital reality of the systems is operated, the data passes through various nodes of the network, where it can be aggregated and converted, each of which equipped with processing capacities. Large (potentially infinite) computing capacities are available in data centers to determine computationally intensive decisions and behaviors. These decisions and behaviors are then (re-)transmitted to the computing units and to the actors of the physical system.

Therefore, the environments in which software modules are deployed and data is processed in DC range from edge and fog nodes to large data centers. These environments form a federation of systems and functions with unified communication and management mechanisms for all participating systems, creating a data processing continuum.

2) *Data continuity*

In order for a DC to form the base for a new level of autonomy of logistic actors, the data and information that is generated in the physical and the digital world must not only be seamlessly available from "shop floor to cloud", as the computing continuity describes, but also from "source to sold", and from "cradle to grave".

To achieve data continuity, logistics actors, i.e., enterprises, cyber-physical systems (CPS) and individual IoT devices, must be connected through data spaces. All actors have access to the same, unique data set (single point of truth) and organize data spaces while respecting data sovereignty. Data spaces can contain historical, current, and predictive data and can be held centralized or decentralized. It is possible to continuously integrate and organize data and information in real-time across entire supply chains and down to the physical level of material handling or robots, making the data required for decision-making or learning situations available at the right level of abstraction.

Data continuity also means, that the same data must be available for decision algorithms in both, the physical and the digital world. The mapping continuity property elaborates on this idea.

3) *Mapping continuity*

For the decision-making algorithms operating within a DC, it must be indistinguishable whether they are operating in the physical or digital world. This is a necessary condition for using the digital reality of a system as an environment for testing and learning alternative behaviors that can be transferred to the physical world.

To achieve the state of continuous mapping, also called mapping continuity, it is necessary to consider two conditions:

Appropriate choice of the level of detail: The accuracy with which the digital representation of the system corresponds to reality should be chosen reasonably. For use cases at a hardware-level, a detailed representation of reality in terms of physics and appearance of objects proves to be of crucial importance(closing the sim-to-real gap, see e.g. [12]). At the supply chain level, where potentially the behavior of entire companies is determined the essential elements of realistic representation are more concerned with inventory quantities, transportation times, weather conditions, etc.

Maintain the twin property between the physical and digital realities: The digital representation of the system must always be kept up to date with the physical system, and the physical system must be able to adapt in real-time to the decisions and behaviors determined in the digital world.

4) *Development continuity*

Development continuity is the characteristic that defines the essential benefits of a DC and distinguishes it from the idea of the industrial metaverse. It integrates all three characteristics defined above and is achieved by applying the ideas and practices of the DevOps approach in software development to the development and operation of autonomous logistics systems.

In a digital reality, advantageous decisions about the behavior of autonomous systems can be learned continuously and at any given time. The learned behavior is then tested in instances of the digital reality that the autonomous decision algorithms have not previously encountered. This ensures that the learned behavior is consistent with the intent of the system designer. If it passes the quality test, the learned behavior is continuously applied to the physical reality in small incremental changes. This creates a closed loop of learning, testing, execution, and feedback, where changes in the physical world lead to changes in the digital world, and vice versa. It enables the acceleration of the process from development to delivery of software and hardware adjustments to the production environment, ensuring alignment of AI-based decisions and enabling continuous learning, improvement, and adaptation.

Starting the process of developing an autonomous system in a DC means that the process of becoming operational is not a sudden and significant implementation but can be integrated into the continuous loop and achieved incrementally. During the development phase the autonomous system can be trained in various digital learning environments for situations that it may encounter in the physical, operational system. Once the system is deployed in the physical world, the pretrained algorithms can be increasingly adapted to the system in which they eventually operate, and during the operational phase the systems remain adaptable to future changes.

The characteristics of DC underline the importance of digital system models for learning and testing beneficial behavior before it is applied to the physical world. Finding the right policy to adapt to a given set of environmental states through training in a digital model, instead of the real world has several advantages such as the training being faster, cheaper, and more scalable [7]. To find behaviors that are robust to the large number of real-world circumstances, a large variety of learning environments is required that sufficiently covers possible combinations of environmental states. However, building such models is expensive, especially for robotic applications, such as AMR in intralogistics, which require physically accurate models to adequately represent sensors and actuators. Therefore, approaches that automatically generate a large number of different digital learning and test environments are necessary to realize the vision of DC. In the next section, the requirements for the generation of digital learning and testing environments for AMR in intralogistics are derived.

III. Derivation of Requirements

The requirements that an approach to generate learning and testing environments for AMR in a DC should fulfill can be derived from the characteristics of DC, characteristics of AMR in intralogistics and policy learning in robotics (see Fig. 2).

A. *Requirements of Digital Continua*

DC and in particular their central property of development continuity provide the fundamental motivation for creating digital learning and testing environments for autonomous systems. It is used in both the development and the operation phases of autonomous systems and imposes different requirements on the digital environments in each of them (see Fig. 2 A.).

In the development phase, since the exact logistics system in which the autonomous robots will operate is not yet determined, a variety of different digital learning environments are helpful to increase the generalizability of the learned behaviors and to achieve good results even in unfamiliar digital and physical environments. Therefore, learning environments should allow autonomous systems to be trained and tested in a variety of environments to prepare them for as robust as possible deployment in different real-world systems.

Similarly, in the operational phase of autonomous systems in intralogistics, the ability to handle unknown and changing environments is necessary, as environmental parameters such as the physical structure of the system or the workload to be processed may change on short notice. To automatically adapt the system to the new situation without manual intervention in the programming of the control logic, digital learning and testing environments with varying environmental parameters are also required in this phase. In contrast to the development phase, the requirement in the operational phase is not to generate completely new digital models, but to be able to adjust specific environmental variables in an existing one.

A requirement for the learning and testing environments that is relevant for both, the development and operation phases, is that the learned behaviors are applicable in the physical reality based on the training in the digital reality. Only in this way is it possible to make use of the advantages of learning in digital worlds for the physical handling of material flows.

Finally, a requirement of DC for learning environments for autonomous systems is that the level of detail in the learning environment should be tailored to the learning task. In order to minimize the computational effort for the generation the learning environments as well as for the learning process itself, the principle of relevance from the principles of proper modeling applies to the appropriate level of abstraction: Only those facts should be modeled that are relevant for the underlying modeling purpose which means that the learning environment should not be as precise as possible, but as accurate as necessary [13].

The learning tasks that are relevant for AMR in intralogistics and therefore dictate the appropriate level of abstraction for the learning environments for AMR in a DC depend on their characteristics and tasks.

B. *AMR in intralogistics*

There are two main characteristics of AMR in intralogistics that define the learning tasks in digital environments: decentralized decision-making and autonomous execution of transports (see Fig. 2 B.).

Autonomous systems are designed to make decentralized decisions and take actions based on the information they receive about the state of their environment. Each entity of the autonomous system forms its own decision-making unit, which can be flexibly added or removed from the system and can react to the environment independently. This fundamental characteristic is crucial for the benefits of autonomous systems in intralogistics, as it ensures that the systems are highly scalable and adaptable. In order to take advantage of this feature in the physical reality for as many planning and control tasks as possible, and thus to make the systems as flexible as possible, the same computing and decision-making architectures as in the physical reality must be replicable in the digital reality. Therefore, decentralized decision-making mechanisms of digital agents must be trainable in a learning environment for AMR in intralogistics.

The decisions to be made the entities of autonomous systems in intralogistics systems are typically related to the execution of transports of one or more handling units, e.g., for put away, picking, or replenishment [4]. The way that these transports are

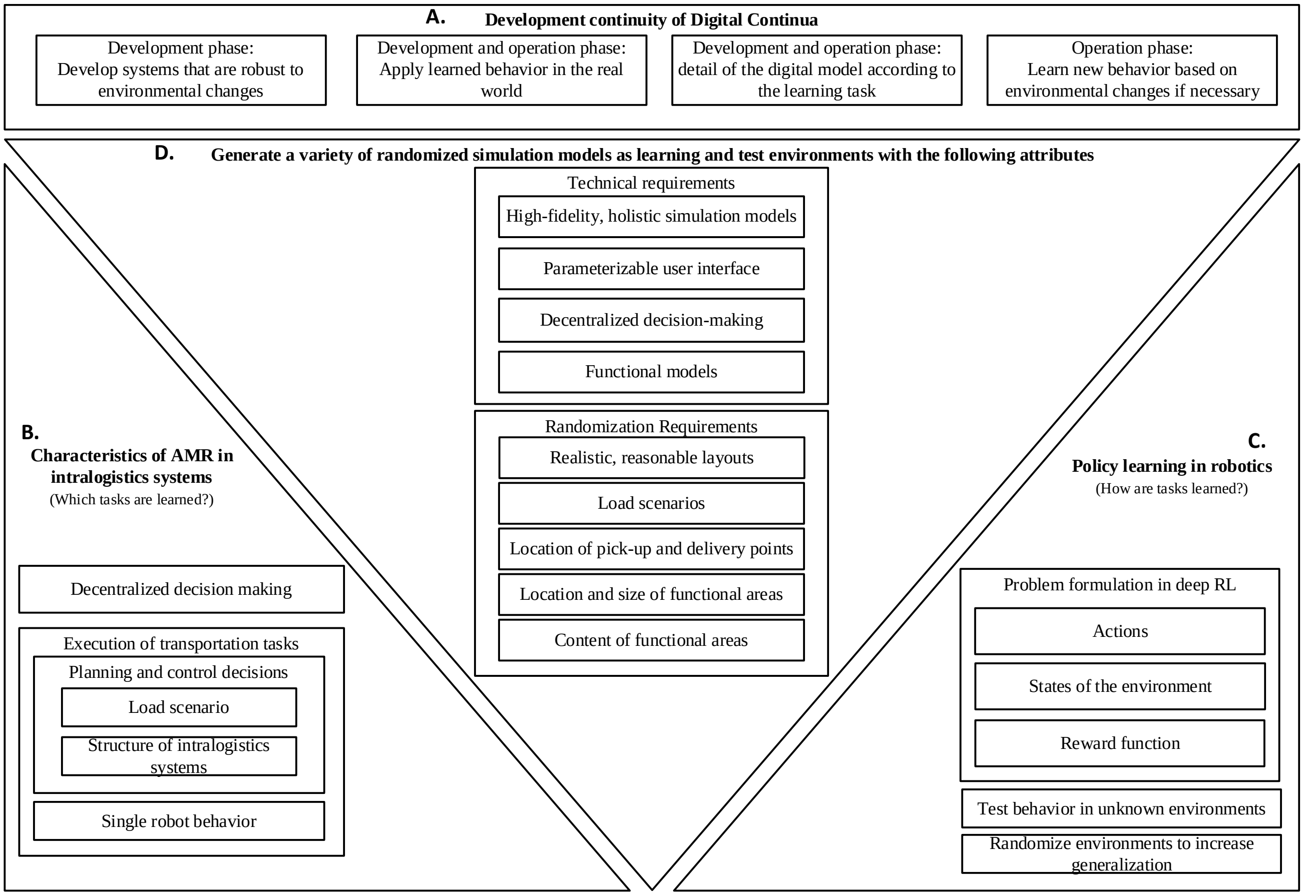

Fig. 2. Requirements for an approach to generate learning environments for AMR in intralogistics

performed determines the overall system behavior and is defined by planning and control decisions as well as the behavior of each of the AMR at the robotic level. Therefore, these are the levers that determine advantageous system behavior and represent the learning tasks for the learning environments.

The planning and control decisions of AMR fleets that affect the execution of transports include the following: resource management, scheduling, dispatching, path planning, collision avoidance, orientation in dynamic environments, and failure handling. The authors of [4] and [14] have identified various studies that have been conducted on each of these individual decision tasks. They have noted that several studies have been conducted in the recent past to address several of these planning and control problems simultaneously and that deep RL has the potential to address these problems successfully. These studies suggest that there is a trend towards integrating the individual decision problems to identify the optimal behavior that encompasses all decisions at once. Therefore, a learning environment for AMR in intralogistics should enable autonomous agents to learn the best behavior that includes these planning and control decisions. The environment should include the variables, that these decisions are usually made upon.

The system variables on which planning and control decisions in intralogistics typically depend are the number of transports to be executed between sources and destinations of a system, and the paths to be taken to make these moves. The number of transports to be executed depends on the respective load scenario under which the system is operating, which may vary, for example, due to seasonal changes, days of the week, or even throughout the day if the inbound and outbound material flows of a system are divided into different shifts. The flexible adaptability of autonomous systems becomes particularly interesting when the load scenario changes abruptly due to an unexpected disruption in the supply chain. To find the most advantageous system behavior under different and rapidly changing load scenarios, different intensities of material flows should be representable in a learning environment.

The paths taken to execute transports depend on the physical structure of the intralogistics system and the locations where transports need to be picked up and delivered. The physical structure of the system refers to the arrangement of functional areas and technical equipment within the functional areas. One of the differences between AMR and traditional AGVs is that AMR do not rely on predetermined paths and can navigate individual paths through their autonomous perception and navigation capabilities. In order to prepare autonomous systems to find their paths even in unfamiliar or changing environments, the structure of the system should be changeable in learning environments.

In addition to planning and control decisions based on the load scenario and structure of an intralogistics system, the robotic behavior of each entity in an autonomous system also determines the overall system behavior. Typical learning tasks at the robotic level include learning acceleration and braking behavior, object recognition, navigation and path finding, e.g., depending on masses, friction coefficients, and surface appearance properties. These robotic aspects determine the times and the paths that AMR in the physical world take to perform a transport. Therefore, these aspects should be included in a digital learning environment to determine the overall system behavior based on realistic assumptions.

After defining the learning tasks, the next section explores the requirements derived from how autonomous agents learn in digital environments.

C. Policy learning in robotics

Recently, deep RL has been shown to be key to training advantageous behaviors of autonomous agents in digital environments. A learning environment for autonomous agents should, therefore, be able to simulate the elements that define a RL task, which include perceiving states of the environment, executing actions based on the perception, and receiving rewards based on the quality of the chosen action [15] (see Fig. 2 C.). The environmental states that the agents must perceive and based on which they decide for a possible action, as well as the reward function, are closely related to the learning task and thus to the tasks of autonomous systems in intralogistics. The mapping between environmental states and actions according to the reward function is called a policy which is the central element that is eventually applied in the physical environment to achieve the same behavior of a digital agent on a robot in the real world.

Before a learned policy is applied in the real world, it must be tested in a variety of digital environments to ensure that it has not been overfitted during training and is consistent with the indented behavior. Research has shown that using a variety of different training environments can improve an agents' ability to generalize and there is some early evidence that highly diverse training environments can promote the emergence of meta-learning in recurrent neural networks, allowing adaption to situations not seen during training [7]. The main idea of randomized learning environments is therefore to increase the diversity of a dataset by adding modified versions of already existing data and thus increasing generalization [6]. For autonomous agents this means training an agent in many simulated learning environments, where certain properties are different in each environment. The goal is to learn a single policy that works well in all of them to increase robustness to changes in the environment and to facilitate the transfer of policies to the physical world real world, since no digital learning environment can perfectly capture the complexity and variability of the physical environment. [12]. An approach to generating learning environments for AMR should therefore include randomizations of the environmental aspects that affect the learning tasks.

D. Summary of requirements

The requirements for learning environments for AMR in intralogistics systems in a DC, derived from the areas of development continuity, AMR in intralogistics, and policy learning in robotics, emphasize the need for diverse, randomized simulation models as learning and test environments (see Fig 2. D.). These requirements can be divided into technical requirements and randomization requirements.

1) Technical Requirements:

1. **High-fidelity, holistic simulation models:** To ensure that the learned behavior is transferable to the real world, the learning environments must be high-fidelity, holistic simulation models that include a physics engine to accurately reproduce physics effects. These models should provide an appropriate level of detail to support learning tasks ranging from sensor and actor level for individual robot behavior to planning and control decisions that depend on the detailed perception of the environment.

2. **Parameterizable user interface:** Users should be able to interact with the generation algorithm by specifying the elements of an environment to be randomized and the number of learning environments required. This fulfills the DC requirement to generate learning environments during the operational phase of an autonomous system and in specific test cases during the development phase.

3. **Decentralized decision-making:** To support policy learning through deep RL, learning environments must allow for the behavior of autonomous agents to be executed in a decentralized manner. This allows agents to make decentralized decisions about actions based on their individual perception.

4. **Functional models:** The generated learning environments should be functional and allow the intended processes to occur. For AMR learning environments, this means that the models must allow agents to perform transports, handle realistic objects, and access all locations where transports need to be picked up or dropped off.

2) Randomization Requirements:

1. **Realistic and reasonable layouts and contents:** The generated environments should represent realistic and reasonable layouts and contents of intralogistics systems, following general design principles. This ensures the transferability of learned behavior to the physical world. All functional areas required for processes from goods-in to goods-out should be included.

2. **Varied learning environments:** To ensure that the learned behavior is generalizable and robust to changes in the real system, the learning environments should be varied.

To create an approach that can produce a wide variety of different and effective learning environments, the following elements should be considered:

- Randomization of location and size of functional areas: This allows agents to learn more robust navigation and adapt to unknown and changing environments.

- Randomization of pick-up and delivery points for transport orders: This ensures that agents learn to perform transports in unknown and changing environments, without focusing on a particular setting of environmental characteristics.

- Randomization of content in functional areas: This aspect improves the agents' perception of the environment and increases robustness to different technologies being used in the functional areas and objects that may block the path.
- Variation of intensity of transport relationships: To prepare autonomous systems for different performance requirements and future load scenarios, the intensity of transport relationships between sources and sinks should be varied.

In conclusion, DC for intralogistics require digital, high-fidelity learning environments that serve as training and test beds for the beneficial behavior of autonomous agents. For the learned behavior to be transferable to the real world, a variety of randomized learning environments is essential. The content that is randomized depends on the parameters of the learning environment on which autonomous decisions are based. The manual generation of different complex learning environments is time consuming and costly, so an approach to generate them automatically is needed. In the field of PCG and DR, approaches for automatically generating varying digital content have been developed for several decades.

IV. Procedural content Generation and Domain Randomization

PCG has its origins in the field of video game development [16]. It refers to a computational technique that involves the use of algorithms to automatically generate content, without the need for direct human input. In addition to video game design, it is now widely used in fields such as art, music, and even scientific simulations. DR can be seen as a simple form of PCG, with its origins in the machine learning community, and as a way to counter overfitting in machine learning. While originating in different domains, PCG and DR both subsume approaches to generating data or content for video games or simulations using algorithms, with the goal of creating variability and unpredictability in the generated content ([7], [17]). In the following, the term PCG will be used to represent both PCG and DR. For more information on specific approaches to PCG see e.g., [7] and [18].

The following section presents relevant work related to the generation of training environments for autonomous systems in a DC. In particular, generation approaches from the application areas of facility layout design, material flow simulation models, and digital environments in robotics are considered.

In the field of facility layout design a variety of PCG approaches have been applied in industry. For example, Kaiser et al. have applied answer set programming to generating highly constrained warehouse layouts [19]. The results are 2D warehouse models that lack high-fidelity attributes and functionality. Dannapfel et al. [20] have applied evolutionary algorithms to facility layout planning and have developed a concept for randomly arranging functional areas in a given space, but have not included the technical equipment in the functional areas and the functionality in the model. Qian at al. have explored general layout design via ML [21] but did not apply their method to high-fidelity modeling of intralogistics systems. López at al. developed a RL based framework to generate virtual system models and validated it through a workstation layout [22]. Their approach does not include functional features.

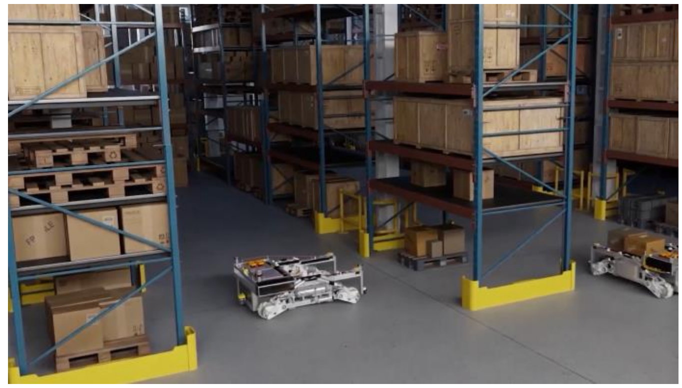

Fig. 3. Warehouse model as digital learning enviroment by Nvidia and Fraunhofer IML [23]

Finally, Nvidia has developed Isaac Sim™, a robotics simulation and data-generation tool, that allows high-fidelity modeling of warehouse environments that can be used for learning use cases at a robotic level. In collaboration with Fraunhofer IML they have included digital robot models of physical prototypes that have been developed at Fraunhofer IML for logistic use cases (Fig. 3) [23]. Isaac Sim™ also provides a toolkit that uses a wavefunction-collapse algorithm to generate randomized high-fidelity warehouse models equipped with pallet racking [24]. While Isaac Sim™ fulfills many of the requirements for learning environments for AMR in DC, it does not support the automatic generation of a wide variety of layouts randomized on a functional area and equipment level yet and is not planned to include the functionality requirement.

PCG has also been used to generate material flow simulations in discrete manufacturing (see [25] for a survey) and warehouse simulation models [26]. These models focus on discrete-event simulation of the material flow therefore not fulfilling the high-fidelity requirement of learning environments for AMR.

In terms of robotics learning environments, for instance, OpenAI used a Progressive PCG (PPCG) approach to train a robot in increasingly complex digital models to solve a rubics cube with one hand and then applied this knowledge in the real world [27]. Their approach demonstrates principles of robotic learning but lacks application to learning behavior of robots in intralogistics systems.

Overall, PCG has proven to be an effective tool for generating learning environments (and data) for training autonomous systems, enabling the creation of expansive, diverse, and modifiable environments. This capability is essential for the proficient development and deployment of AI-driven autonomous systems in the logistics domain. While there are promising approaches to PCG in the context of training environments for autonomous systems within a DC, thus far, no approach has yet addressed all the relevant requirements comprehensively. This is due to the complexity of the requirements and the need for a coherent and efficient integration of the different components.

V. CONCLUSION AND OUTLOOK

Autonomous systems have the potential to fundamentally transform the logistics industry. However, in order to fully realize this potential, adequate and comprehensive approaches for the development and implementation of such systems are needed. The DC paradigm offers a promising approach by leveraging the seamless fusion of physical and digital reality to achieve a new level of autonomy through AI-based methods. Extending the industrial metaverse with an AI-based autonomy approach enables continuous optimization and adaptation of these systems in many use cases.

In this regard, the creation of suitable digital learning and testing environments is crucial. Due to the necessary mapping of physical complexity to virtual models, the scale and the high number of required variations, automatic generation of these environments should be pursued. However, current PCG approaches are still insufficient to generate learning and testing environments that meet the diverse requirements of AI-based autonomous systems in DC. This reveals a research gap that needs to be closed in order to fully exploit the potential of AI-based autonomous systems. Only within a fully continuous development process can AI-based autonomous systems AMRs be effectively deployed in intralogistics systems.

In summary, the concept of the DC in combination with AI-based autonomous systems has enormous potential for the future of the intralogistics industry. As PCG approaches evolve within high-fidelity simulation environments, digital learning and testing environments will become the central tool to revolutionize the continuous development of intralogistics systems even during operation. Successful deployment of the DC could finally achieve a level of automated adaptivity at the system level that is needed to allow highly autonomous systems, resulting in a more resilient and adaptive global supply chain.

REFERENCES

[1] M. ten Hompel and S. Kerner, “Logistik 4.0: Die Vision vom Internet der autonomen Dinge,” *Inform.-Spektrum*, vol. 38, no. 3, pp. 176–182, Jun. 2015, doi: 10.1007/s00287-015-0876-y.

[2] W. Wahlster, “Künstliche Intelligenz als Grundlage autonomer Systeme,” *Inform.-Spektrum*, vol. 40, no. 5, pp. 409–418, Oct. 2017, doi: 10.1007/s00287-017-1049-y.

[3] R. Adler, “Autonome Systeme: großes Potenzial für die digitale Zukunft - Blog des Fraunhofer IESE,” *Dependable AI - Autonome Systeme*, Jun. 10, 2021. https://www.iese.fraunhofer.de/blog/autonome-systeme/ (accessed Mar. 29, 2023).

[4] G. Fragapane, R. de Koster, F. Sgarbossa, and J. O. Strandhagen, “Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda,” *Eur. J. Oper. Res.*, vol. 294, no. 2, pp. 405–426, Oct. 2021, doi: 10.1016/j.ejor.2021.01.019.

[5] J. Fottner *et al.*, “Autonomous Systems in Intralogistics – State of the Art and Future Research Challenges,” *LogisticsResearch*, vol. 14, no. 2, Feb. 2021, doi: 10.23773/2021_2.

[6] F. Muratore, F. Ramos, G. Turk, W. Yu, M. Gienger, and J. Peters, “Robot Learning From Randomized Simulations: A Review,” *Front. Robot. AI*, vol. 9, 2022.

[7] S. Risi and J. Togelius, “Increasing generality in machine learning through procedural content generation,” *Nat. Mach. Intell.*, vol. 2, no. 8, pp. 428–436, Aug. 2020, doi: 10.1038/s42256-020-0208-z.

[8] D. Laß, “Fakt oder Fiktion? Technologien und Use Cases für das (Industrial) Metaverse,” Fraunhofer-Verbund IUK-Technologie, Berlin, 2022.

[9] Greg Brockman [@gdb], “The underlying spirit in many debates about the pace of AI progress…,” *Twitter*, Apr. 12, 2023. https://twitter.com/gdb/status/1646183424024268800 (accessed Apr. 16, 2023).

[10] M. Schmidt, “"Simulation-based AI” and the ‘digital continuum,’” *Jahresber. 2022 Fraunhofer-Inst. Für Mater. Logist. IML*, 2022.

[11] G. Antoniu *et al.*, “A blueprint for the new Strategic Research Agenda for High Performance Computing.” European Technology Platform for High-Performance Computing ETP4HPC, European eXtreme Data and Computing Initiative exdci, Apr. 2019.

[12] W. Zhao, J. P. Queralta, and T. Westerlund, “Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey,” in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec. 2020, pp. 737–744. doi: 10.1109/SSCI47803.2020.9308468.

[13] J. Becker, W. Probandt, and O. Vering, *Grundsätze ordnungsmäßiger Modellierung: Konzeption und Praxisbeispiel für ein effizientes Prozessmanagement*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. doi: 10.1007/978-3-642-30412-5.

[14] M. Wesselhöft, J. Hinckeldeyn, and J. Kreutzfeldt, “Controlling Fleets of Autonomous Mobile Robots with Reinforcement Learning: A Brief Survey,” *Robotics*, vol. 11, no. 5, Art. no. 5, Oct. 2022, doi: 10.3390/robotics11050085.

[15] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*, Second edition. in Adaptive computation and machine learning series. Cambridge, Massachusetts: The MIT Press, 2018.

[16] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup, “Procedural content generation for games: A survey,” *ACM Trans. Multimed. Comput. Commun. Appl.*, vol. 9, no. 1, pp. 1–22, Feb. 2013, doi: 10.1145/2422956.2422957.

[17] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games*. in Computational Synthesis and Creative Systems. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-42716-4.

[18] R. M. Smelik, T. Tutenel, R. Bidarra, and B. Benes, “A Survey on Procedural Modelling for Virtual Worlds: A Survey on Procedural Modeling for Virtual Worlds,” *Comput. Graph. Forum*, vol. 33, no. 6, pp. 31–50, Sep. 2014, doi: 10.1111/cgf.12276.

[19] P. Kaiser, A. Thevapalan, C. Reining, G. Kern-Isberner, and M. ten Hompel, “Generating Highly Constrained Warehouse Layouts Using Answer Set Programming,” *Vol. 2021*, p. Issue 17, 2021, doi: 10.2195/LJ_PROC_KAISER_EN_202112_01.

[20] M. Dannapfel, S. P. Vierschilling, S. Losse, and O. Matzke, “Generative Design In Factory Layout Planning: An Application Of Evolutionary Computing Within The Creation Of Production Logistic Concepts,” *Conf. Prod. Syst. Logist. CPSL 2020*, 2020, doi: 10.15488/9656.

[21] C. Qian, R. K. Tan, and W. Ye, “An adaptive artificial neural network-based generative design method for layout designs,” *Int. J. Heat Mass Transf.*, vol. 184, p. 122313, Mar. 2022, doi: 10.1016/j.ijheatmasstransfer.2021.122313.

[22] C. E. Lopez, O. Ashour, and C. S. Tucker, “Reinforcement Learning Content Generation for Virtual Reality Applications,” presented at the ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Nov. 2019. doi: 10.1115/DETC2019-97711.

[23] S. Kerner and J. Eßer, “Towards a Digital Reality in Logistics Automation: Optimization of Sim-to-Real,” presented at the NVIDIA GTC Digital Spring, Mar. 2022. Accessed: Apr. 20, 2023. [Online]. Available: https://www.nvidia.com/en-us/on-demand/session/gtcspring22-s42559/

[24] NVIDIA, “8. Replicator SceneBlox tutorial — Omniverse Robotics documentation,” Mar. 23, 2023. https://docs.omniverse.nvidia.com/app_isaacsim/app_isaacsim/tutorial_replicator_sceneblox.html (accessed Mar. 23, 2023).

[25] H. Reinhardt, M. Weber, and M. Putz, “A Survey on Automatic Model Generation for Material Flow Simulation in Discrete Manufacturing,” *Procedia CIRP*, vol. 81, pp. 121–126, Jan. 2019, doi: 10.1016/j.procir.2019.03.022.

[26] F. Kallat, J. Pfrommer, J. Bessai, J. Rehof, and A. Meyer, “Automatic Building of a Repository for Component-based Synthesis of Warehouse Simulation Models,” *Procedia CIRP*, vol. 104, pp. 1440–1445, 2021, doi: 10.1016/j.procir.2021.11.243.

[27] OpenAI *et al.*, “Solving Rubik’s Cube with a Robot Hand.” arXiv, Oct. 15, 2019. doi: 10.48550/arXiv.1910.07113.