Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 130 (2024) 1632–1637

www.elsevier.com/locate/procedia

57th CIRP Conference on Manufacturing Systems 2024 (CMS 2024)

An extension of the Core Manufacturing Simulation Data standard to enhance the interoperability for discrete event simulation

Jiaqi Zhao[a,b,*], El-Houssaine Aghezzaf[a,b], Johannes Cottyn[a,b]

[a]Department of Industrial Systems Engineering and Product Design, Ghent University, Technologiepark 46, Gent-Zwijnaarde 9052, Belgium
[b]Industrial Systems Engineering (ISyE), Flanders Make, www.FlandersMake.be, Belgium

* Corresponding author. Tel.: +32-484-968-390; fax: +32-484-968-390. *E-mail address:* Jiaqi.Zhao@UGent.be

Abstract

Discrete event simulation are vital in building a manufacturing system. The emergence of the industrial metaverse increases the collaboration needs between simulation engineers. Data reuse between different simulation software applications could significantly lower manual modeling efforts and shorten the overall time of system construction. To achieve this, interoperability in discrete event simulation is a prerequisite. Core Manufacturing Simulation Data (CMSD) is a promising standard for exchanging discrete event simulation models. However, the current CMSD standard is insufficient to achieve full-scope interoperability between discrete event simulation software applications. In this paper, a case-driven iterative approach is applied to extend the CMSD standard. An initial extension is developed based on a first iteration. A method for exchanging models between different discrete event simulation software applications via the extension is proposed. With this method, the interoperability performance of the extension is evaluated by exchanging a representative set of discrete event simulation models between two simulation software applications, FlexSim and Visual Components, via self-developed "Import" and "Export" plug-ins. The result shows that 57.5% more attributes can be converted via the extension than the original CMSD standard.

Keywords: Interoperability; Discrete event simulation; Core Manufacturing Simulation Data; Data exchange.

1. Introduction

With the evolution of Industry 4.0, Digital Twin (DT) is developing rapidly in the smart manufacturing field [1]. Virtual Commissioning (VC) is a method that could contribute to the development and integration of DTs [2]. Based on a virtual model, VC detects potential errors in the early phase, significantly shortening the real commissioning time of system construction [3, 4]. As the emergence of the industrial metaverse increases the demand for 3D visualization, 3D model-based VC (3D VC) are likely to become increasingly pivotal [5]. Based on the hierarchy of manufacturing systems, 3D VC can be divided into three levels: machine, cell and plant [6, 7]. For the machine level, real-time emulation is conducted to synchronously mimic the execution of mechatronic components in connection with external Programmable Logic Controller (PLC) programs. A 3D emulation model is usually physics-based and in high fidelity. For the plant level, the target shifts to optimize a manufacturing plant's operations and logistics. Cycle time and setup time in the manufacturing process are key influencing factors. Physics-related information is no longer necessary, and geometric fidelity is significantly reduced as it is only used for 3D shape and animation visualization. At this level, Discrete Event Simulation (DES) is executed instead of real-time emulation.

The development of Virtual Reality (VR) and industrial metaverse boosts the collaboration of end users with different 3D software applications [8]. Reusing software-specific 3D models will enable (semi)automated model data exchange between different simulation software applications,

10.1016/j.procir.2024.10.293

significantly reducing manual remodeling work [9]. However, interoperability between different 3D software applications is still a challenge to be resolved in the DT domain [10, 11]. To the authors' best knowledge, neutral data exchange formats, such as Standard for the Exchange of Product Data (STEP) and JT, are used to exchange models between 3D simulation software applications. However, only 3D geometry can be reused in most cases, which means the rest of the information must be modeled manually. This creates a heavy workload for simulation engineers.

AutomationML (AML) is influential in modeling and exchanging data in the engineering phase [12]. For 3D-based emulation data exchange, the current AML standard has been extended towards an "AML-VC extension", which converts 70% more attributes than the original AML standard during data exchange between different 3D emulation software applications [13]. In terms of interoperability in DES, CMSD is proved to be a promising data exchange standard, as CMSD converts more DES-related attributes than AML [9]. However, only 26.4% of total attributes can be converted via CMSD, which needs to be improved [9]. In this case, the authors suggest extending the current CMSD standard towards full-scope interoperability between different DES software applications.

The remainder of the paper is structured as follows. The state-of-the-art is introduced in Section 2. Section 3 elaborates on developing an initial CMSD extension based on a first iteration. A method is proposed in Section 4 to achieve interoperability between two DES software applications via the CMSD extension, and an example is given to verify the feasibility of this method. The interoperability performance of the CMSD extension is evaluated and discussed in Section 5. Finally, conclusion and outlook are described in Section 6.

2. State-of-the-art

This section investigates the history of DES and the current status of DES interoperability. Besides, the concept of CMSD and related work on its extension are also studied.

2.1. The evolution of DES and the interoperability between DES software applications

DES simulates the dynamics of the real world through a series of events, contributing to decision-making based on the generated performance report [14]. In the manufacturing domain, DES is widely used to optimize scheduling and logistics for production lines [15]. Since the 1960s, various DES software applications have been developed and applied in many sectors [16]. However, these software applications are invented by different software companies and with mutual-independent software kernels [17]. Software-specific DES models cannot be exchanged with "one button play", which creates a tremendous obstacle for different software end users to collaborate [18]. In this case, a generic data model for automatically generating and exchanging DES models could be beneficial.

To date, several neutral data exchange formats or industrial standards have the potential to achieve automated simulation model generation [9, 19]. Studies have also been conducted in the manufacturing domain to achieve automated DES model generation [9]. However, a comprehensive data model for full-scope interoperability in DES has yet to be found [9].

In previous research, the authors compared representative neutral data exchange formats (industrial standards) for 3D-based interoperability to find a starting point towards full-scope data exchange between different DES software applications. It shows that CMSD (SISO-STD-008) and AML (IEC 62714) have more concrete attribute libraries for data exchange in the field of DES in comparison with other formats (standards), such as ISA-95 (IEC 62264), STEP (ISO 10303) and JT (ISO 14306) [9]. The interoperability performance of CMSD and AML was subsequently evaluated by exchanging ten DES models between FlexSim (FLS) and Visual Components (VCO). The result shows that 26.4% of DES-related attributes can be converted via CMSD, while the attribute conversion rate is 19.3% via AML. Based on this, CMSD was recommended for further extension towards full-scope interoperability between different DES software applications [9].

2.2. The CMSD standard and related work on its extension

CMSD is an industrial standard for interoperability between simulation systems and manufacturing applications, such as Enterprise Resource Planning (ERP) and Manufacturing Execution Systems (MES). CMSD integrates layout, part, resource, production operation and planning information in one interconnected data structure [20]. In CMSD, information can be represented in two different forms: Unified Modeling Language (UML) (graphical) [21] and Extensible Markup Language (XML) (textual) [22]. XML enhances the practical application capability of the CMSD standard in data exchange scenarios [23].

Though CMSD provides a fundamental information model and data format for simulation-related data exchange, it is still inadequate for automated DES model generation. Lee et al. [20] enlarged the CMSD standard by adding conveyor speed, elevator load and unload speed, staff numbers allocated to each workstation, resource cycle time, Mean Time To Failure (MTTF) and Mean Down Time (MDT). Bergmann et al. [24] suggested adding worker current location, conveyor current speed and acceleration, and job remaining processing times into the CMSD standard. He also mentioned that resource capacity, buffers, decision and routing rules, resource availability and reliability should be included [25]. Besides, Steinbacher et al. [26] extended the CMSD standard by proposing a UML-based ontology for automated material flow simulation model generation in third-party logistics. In this ontology, elements such as dispatcher, buffer, anchor point, physics object, process link, transporter moving speed have been integrated into the CMSD standard.

These studies have added necessary attributes to the CMSD standard to enable automated DES model generation for specific use cases. However, the comprehensiveness of the CMSD standard still needs to be ensured. In summary, extending the current CMSD standard towards a "CMSD-VC extension" for full-scope interoperability between different DES software applications is still an open point.

3. The development of the CMSD-VC extension

A case-driven iterative approach is used to gradually evolve towards the CMSD-VC extension. Various DES software applications are used as cases. During each iteration, the CMSD-VC extension is continuously enriched by taking the union of all DES-related attributes of the DES software application. A naming rule is applied when a new attribute has a similar definition to an existing attribute in the CMSD-VC extension. A detailed description of the case-driven iterative approach is elaborated in previous work [13].

A first iteration is conducted by using two DES software cases, namely FLS and VCO ("Process" Module). 2180 new attributes of 48 elements have been added to CMSD.

An example list of new elements and attributes in the CMSD-VC extension is presented in Table 1.

Table 1. An example list of new elements (marked as *) and new attributes added to the CMSD standard after a first iteration.

Class	Element	Added attributes
Basic types	Resource type	Source, sink, buffer, elevator, articulated robot, gripper, mobile robot, warehouse rack, dispatcher, …
Resource	Connection	From resource frame, to resource frame, ...
Resource	Source*	Feed mode, time interval, arrive at time 0, limit, part creation probability, …
Resource	Buffer*	Buffer mode, maximum content, …
Resource	Conveyor	Visualization customization, capacity, speed, acceleration, deceleration, a series of frames for representing the path on the conveyor, …
Resource	Machine	Visualization customization, capacity, Failure mode, Mean Time Between Failures (MTBF), Mean Cycles Before Failure (MCBF), Mean Time To Repair (MTTR), ...
Resource	Transporter	Capacity, maximum speed, maximum acceleration, maximum deceleration, …
Resource	Articulated robot*	Inverse kinematics, maximum speed, acceleration, deceleration of each joint, joint axes, base frame, Tool Center Point (TCP) frame, tool connection interface, …
Resource	Gripper*	Kinematics of joints, open value, close value, TCP depth, robot connection interface, …
Resource	Mobile robot*	Exclude obstacle components, is obstacle avoidance, pick offset, place offset, move speed, move speed loaded, move speed approach, move speed avoidance, power and charge settings, …
Resource	Warehouse rack*	Visualization customization, quantity and dimension definition of bay, slot and level, slot assignment strategy, slot stacking order, pick and place Y offset, ...
Resource	Path	Is two way, accumulation type, …
Resource	Dispatcher*	Resource priority, queue strategy, ...
Part	Part	3D file path, part color, Stock Keeping Unit (SKU), ...
Process	Process	Load time, unload time, pick time, place time, …
Job	Event	On creation, on arrival, on entry, on exit, on task start, on task finish, on state change, …
Layout	Geometry and kinematics*	Geometry and kinematics-related attributes in Collada (1.5.0 version) (including "up axis" and "model unit").

As listed in Table 1, more DES-related resource types (elements) with attributes have been added, such as source, sink, buffer, articulated robot, mobile robot, warehouse rack, and dispatcher. For example, the "time interval" attribute is added to the "Source" element to define the time interval between two adjacent flow item creations. Frames (such as base frame and TCP frame) and kinematics information (like maximum speed, acceleration and deceleration of each joint) are added to the "Articulated robot" element. For the "Mobile robot" element, obstacle avoidance, condition-specific moving speed, power and charge-related attributes are included.

Moreover, new attributes are added to the original CMSD resource types. For instance, "capacity", "speed" and "acceleration" are added to the "Conveyor" element. Failure-related information, such as MTBF, MCBF, and MTTR, is attached to the "Machine" element. For the "Process" class, more time-related attributes like load time and unload time are added. Events like "on creation" and "on arrival" are included in the CMSD-VC extension to refine the "Job" class. Besides, entry point frames and exit point frames in the resource connections are also vital. These frames indicate entry and exit locations on the resources while transporting a flow item. Therefore, the "from resource frame" and "to resource frame" attributes are added to the "Connection" element of Class "Resource".

Furthermore, it has been found that no 3D kinematics-related information is included in the CMSD standard. As 3D data are indispensable in VR and the industrial metaverse, the extension of the CMSD standard to include 3D geometry and kinematics description will contribute to the interoperability not only in DES but also between 3D emulation and simulation. Collada is a generic XML-based open-source data format for exchanging 3D models with geometry, physics and kinematics information [27]. If Collada is integrated into CMSD, 3D shape and animation information in DES will be exchangeable, which could be an added value. Therefore, Collada is integrated into the CMSD-VC extension. It is noteworthy that attributes like "up axis" and "model unit" must be mentioned in the Collada format, as these parameters vary from software applications. This may cause a 3D model to display in the wrong size or up-direction if these parameters are not mentioned correctly.

The element and attribute enrichment of the CMSD standard takes a meaningful first step towards full-scope interoperability between different DES software applications.

4. A CMSD-VC extension-based method for data exchange between different DES software applications

A CMSD-VC extension provides a generic information model for interoperability in DES. However, as the data modeling structures of different DES software applications are heterogeneous, the DES model can only be edited and saved in a software-specific data format. Currently, the CMSD standard is not supported by commercial DES software applications. In this case, a CMSD-VC extension-based method for data exchange between different DES software applications must be developed. Based on this newly developed CMSD-VC extension, a method to exchange DES models between FLS and VCO is created, which is depicted in Fig. 1.

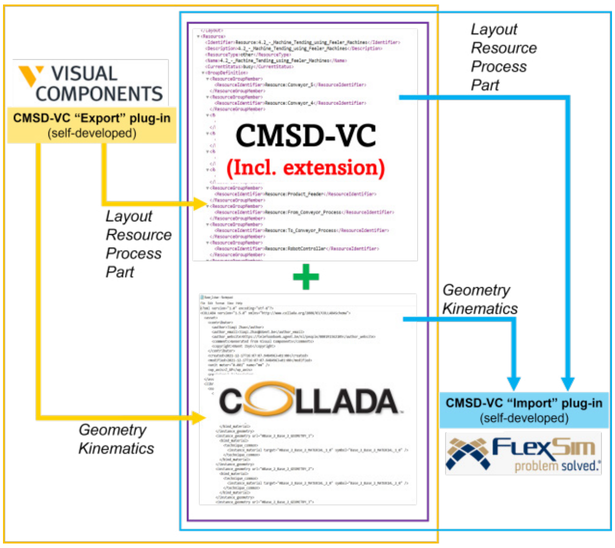

Fig. 1. A method for transmitting DES models from VCO to FLS via the CMSD-VC extension.

As is shown in Fig. 1, a method for CMSD-VC extension-based data transmission from VCO to FLS is illustrated. As neither software application supports the CMSD standard, “Import” and “Export” plug-ins are created in both software applications respectively. The Application Programming Interface (API) in VCO and FLS is used to develop the plug-ins in both software applications. C# .NET is used to create the plug-ins in VCO, while FlexScript is used in FLS. With this method, a data package of an XML-based CMSD-VC extension file integrated with Collada files is generated by clicking the self-developed “CMSD-VC Export” button of VCO. Layout, resource, process and part-related information is stored in the CMSD-VC extension data structure, while geometry and kinematics information is stored in Collada files. Every Collada file is linked to the CMSD-VC extension via a Uniform Resource Identifier (URI) of the “FileName” attribute under the “Graphic” node of the “Layout” class. Then, the CMSD-VC extension data package is imported into FLS via the self-defined “CMSD-VC Import” button of FLS. It is worth noting that if kinematics information is contained in a Collada file, the Collada file must be separated into several Collada subfiles during the import process. This is because kinematics information is defined in the “Animation” module of FLS, and each link of the kinematics must be referenced separately in this module. It is also important to note when importing an articulated robot into FLS, the vector of the joint revolute axis in VCO and FLS might differ. In this case, the forward direction of the link must be changed during the import to ensure the vectors are consistent with the corresponding software application. After the import of the CMSD-VC extension, the 3D DES model is generated in FLS. The data transmission from FLS to VCO is a reverse process of the described method in Fig. 1.

To verify the feasibility of the method, a demonstration is conducted by transmitting a Machine Tending Work Cell (MTWC) DES model from VCO to FLS. The original VCO-based MTWC model is shown in Fig. 2 (a). In this model, a product feeder creates a cylinder flow item in a specific time interval. The flow item is then transferred on a conveyor line and enters into a machine tending mechanism. An articulated robot then picks up the cylinder and places it into a Computer Numerical Control (CNC) machine. The CNC machining then converts the cylinder to a block. After that, the same robot transfers the block from the CNC machine to another conveyor line, and the flow item is transferred away via the conveyor transportation mechanism. In the end, the block sinks at the end of the conveyor line.

Based on the data exchange method described in Fig. 1, the MTWC model-based verification process is executed. First, an MTWC model is created in VCO. Then, a CMSD-VC extension-based data package is generated by clicking the “CMSD-VC Export” button of VCO. After this, the CMSD-VC extension is imported into FLS by pressing the self-developed “CMSD-VC Import” button of FLS, and a 3D MTWC model is generated in FLS. The generated MTWC model is shown in Fig. 2 (b). This demonstration verifies that the method for exchanging 3D simulation models via the CMSD-VC extension between FLS and VCO is feasible.

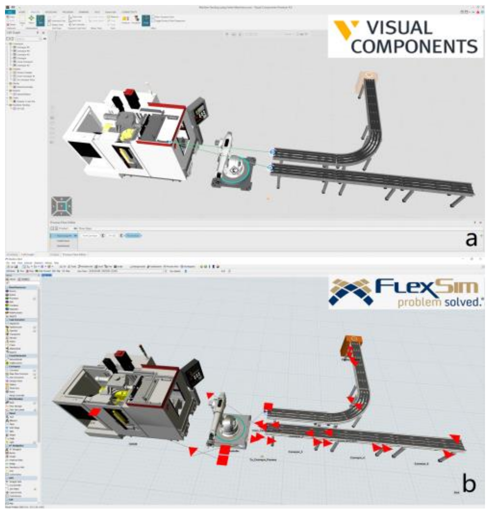

Fig. 2. The original VCO model (a) and the generated FLS model (b) with CMSD-VC extension-based data exchange method.

5. Interoperability performance evaluation of the CMSD-VC extension

Based on the CMSD-VC extension-based data exchange method and self-developed “Import” and “Export” plug-ins, a case study is conducted to evaluate the interoperability performance of the CMSD-VC extension by exchanging a set of DES models between FLS and VCO. The DES models must be as diverse as possible, covering most categories of DES-related functions in FLS and VCO. According to this rule, five FLS models and five VCO models are selected. The data exchange of these ten DES models is subsequently conducted between FLS and VCO via the "Import" and "Export" plug-ins. The entire methodology of the interoperability performance evaluation of a generic VC data model is described in detail in previous work [28].

During the implementation of CMSD-VC extension-based data exchange between FLS and VCO, the attribute quantities and conversion rates of these ten models are calculated in each procedure. The result of the five DES models transmitted from FLS to VCO is listed in Table 2, while that from VCO to FLS is outlined in Table 3. To show the contribution of the CMSD-VC extension intuitively, the attribute quantities and conversion rates of these models via the original CMSD standard are also displayed.

Table 2. Attribute quantities and conversion rates of five FLS DES models during data exchange from FLS to VCO based on CMSD and CMSD-VC extension.

DES model number	1	2	3	4	5	Average
Attribute quantity in FLS	813	1229	1415	927	559	/
Attribute quantity in CMSD	162	239	277	170	148	/
Attribute quantity in CMSD-VC extension	557	896	1000	683	367	/
Attribute quantity in VCO converted via CMSD	130	203	246	133	114	/
Attribute quantity in VCO converted via CMSD-VC extension	270	418	456	294	201	/
Conversion rate from FLS to CMSD (unit: %)	19.9	19.4	19.6	18.3	26.5	20.8
Conversion rate from FLS to CMSD-VC extension (unit: %)	68.5	72.9	70.7	73.7	65.7	**70.3**
Conversion rate from FLS to VCO via CMSD (unit: %)	16.0	16.5	17.4	14.3	20.4	16.9
Conversion rate from FLS to VCO via CMSD-VC extension (unit: %)	33.2	34.0	32.2	31.7	36.0	**33.4**

Table 3. Attribute quantities and conversion rates of five VCO DES models during data exchange from VCO to FLS based on CMSD and CMSD-VC extension.

DES model number	6	7	8	9	10	Average
Attribute quantity in VCO	987	335	526	209	288	/
Attribute quantity in CMSD	266	73	112	72	53	/
Attribute quantity in CMSD-VC extension	885	305	485	183	258	/
Attribute quantity in FLS converted via CMSD	257	68	97	71	53	/
Attribute quantity in FLS converted via CMSD-VC extension	349	116	183	93	83	/
Conversion rate from VCO to CMSD (unit: %)	27.0	21.8	21.3	34.4	18.4	24.6
Conversion rate from VCO to CMSD-VC extension (unit: %)	89.7	91.0	92.2	87.6	89.6	**90.0**
Conversion rate from VCO to FLS via CMSD (unit: %)	26.0	20.3	18.4	34.0	18.4	23.4
Conversion rate from VCO to FLS via CMSD-VC extension (unit: %)	35.4	34.6	34.8	44.5	28.8	**35.6**

As is listed in Table 2, the average attribute conversion rate from FLS to the CMSD-VC extension and that from FLS to VCO via the extension are 70.3% and 33.4%. In comparison, the conversion rate from FLS to CMSD and that from FLS to VCO via CMSD are 20.8% and 16.9%. Similarly, according to Table 3, the attribute conversion rate from VCO to the CMSD-VC extension is 90.0%, and that from VCO to FLS via the CMSD-VC extension is 35.6%. Likewise, the attribute conversion rate from VCO to CMSD is 24.6%, and that from VCO to FLS via CMSD is 23.4%.

The result shows that the average attribute conversion rate between the DES software application and the CMSD-VC extension is 80.2%, while the rate between FLS (or VCO) and CMSD is 22.7%. It means that 57.5% more attributes can be converted via the CMSD-VC extension than CMSD. It indicates that the CMSD-VC extension significantly improves the interoperability performance between FLS and VCO.

As the CMSD-VC extension is created by taking the union of all DES-related functions and attributes in FLS and VCO, you could argue that the attribute conversion rate between FLS (or VCO) and the CMSD-VC extension should actually be 100%. However, some attributes are defined based on the script, and the flexibility and irregularity in the script syntax definition make it extremely difficult to recognize and extract relevant attributes. This inconvenience decreases the attribute conversion rate between the DES software application and the CMSD-VC extension. As FLS contains more script-based attributes, the conversion rate between FLS and the CMSD-VC extension is 19.7% lower than that between VCO and the extension.

It is worth noting that the average attribute conversion rate between FLS and VCO via the CMSD-VC extension is 34.5%, while the rate via the original CMSD standard is 20.2%. Though the rate raises 14.3% compared to only via CMSD, the conversion rate has a 45.7% drop after the import of the CMSD-VC extension into the DES software application. The rate decrease is due to two reasons. One reason is the difference in the software capability and data structure. It has been found that FLS focuses more on defining rules (such as dispatching, routing and scheduling rules), while VCO has more comprehensive 3D component libraries. This difference brings out a problem: some attributes in one DES software application cannot be reused in another. The difficulty in converting script-based attributes from one DES software application to another is also an important reason for the decrease in the attribute conversion rate.

Furthermore, it should be noted that the openness of DES software applications is also indispensable for the CMSD-VC extension-based data exchange. Software-specific "Import" and "Export" plug-ins must be developed as most commercial DES software applications do not support the CMSD standard. The accessibility and comprehensiveness of the APIs make it possible to build an interoperability bridge between different DES software applications. This CMSD-VC extension information model can be a reference for software engineers who want to develop "Import" and "Export" plug-ins for their DES software applications.

6. Conclusion and outlook

This paper extends the current CMSD standard towards an initial "CMSD-VC extension" for the interoperability between different DES software applications. The extension is developed based on a case-driven iterative approach, and 2180 new attributes have been added by a first iteration via two software applications, namely FLS and VCO. The CMSD classes, such as "Resource", "Part", "Process", "Job" and "Layout", have been comprehensively enriched with new DES-related resource types, elements and attributes during the iteration. Furthermore, the Collada format has been integrated into the CMSD-VC extension data structure to represent 3D geometry and kinematics information, significantly strengthening the 3D representation capability of the CMSD standard. To enable the interoperability between these two software applications, a method for exchanging DES models between FLS and VCO via the CMSD-VC extension has been proposed. A case study has been conducted by exchanging ten DES models between FLS and VCO. It shows that 80.2% of attributes can be converted between FLS and VCO via the CMSD-VC extension, which is 57.5% more than that via the original CMSD standard. The development of the CMSD-VC extension significantly enhances the interoperability between different DES software applications and paves the way towards the interoperability between 3D-based emulation and simulation.

For future work, script-based attributes will be analyzed, and critical parameters will be extracted from the script and integrated into the CMSD-VC extension. Besides, the data reuse potential between 3D emulation and simulation models will also be explored.

Acknowledgements

This research is financially supported by the China Scholarship Council (CSC).

References

[1] Tao F, Zhang H, Liu A, Nee AY. Digital twin in industry: State-of-the-art. IEEE Transactions on industrial informatics 2018;15(4):2405-2415.

[2] Barbieri G, Bertuzzi A, Capriotti A, Ragazzini L, Gutierrez D, Negri E, Fumagalli L. A virtual commissioning based methodology to integrate digital twins into manufacturing systems. Production Engineering 2021;15(3):397-412.

[3] Schamp M, Hoedt S, Claeys A, Aghezzaf E-H, Cottyn J. Impact of a virtual twin on commissioning time and quality. 16th IFAC Symposium on Information Control Problems in Manufacturing 2018. p. 1047-1052.

[4] Hoffmann P, Schumann R, Maksoud TM, Premier GC. Virtual commissioning of manufacturing systems a review and new approaches for simplification. ECMS 2010;175-181.

[5] Striffler N, Voigt T. Concepts and trends of virtual commissioning – A comprehensive review. Journal of Manufacturing Systems 2023;71:664-680.

[6] Schamp M, Demasure T, Huysentruyt S, Lamote J, Aghezzaf E-H, Cottyn J. Multi-level approach to virtual commissioning: a reconfigurable assembly system case. 10th IFAC Conference on Manufacturing Modelling, Management and Control 2022. p. 3208-3213.

[7] Reinhart G, Wünsch G. Economic application of virtual commissioning to mechatronic production systems. Production engineering 2007;1:371-379.

[8] Bellalouna F, Puljiz D. Use case for the Application of the Industrial Metaverse Approach for Engineering Design Review. Procedia CIRP 2023;119:638-643.

[9] Zhao J, Aghezzaf E-H, Cottyn J. Interoperability performance evaluation for discrete event simulation models: A step towards multi-level data exchange. 34th CIRP Design Conference 2024. (Accept) Available at: https://drive.google.com/file/d/1ecL1orQmogI6rPw83PO4NwpgnbUdR44v.

[10] Qi Q, Tao F, Hu T, Anwer N, Liu A, Wei Y, et al. Enabling technologies and tools for digital twin. Journal of Manufacturing Systems 2021;58:3-21.

[11] Kerrouchi S, Aghezzaf E-H, Cottyn J. Production digital twin: a systematic literature review of challenges. International Journal of Computer Integrated Manufacturing 2024;1-26.

[12] Zhao J, Schamp M, Hoedt S, Aghezzaf E-H, Cottyn J. AutomationML in industry 4.0 environment: A systematic literature review. Stuttgart Conference on Automotive Production 2021. p. 162-169.

[13] Zhao J, Aghezzaf E-H, Cottyn J. An AutomationML extension towards interoperability of 3D virtual commissioning software applications. International Journal of Computer Integrated Manufacturing 2023;1-20.

[14] Law AM, Kelton WD, Kelton WD. Simulation modeling and analysis: Mcgraw-hill New York 2007.

[15] Aretoulaki E, Ponis ST, Plakas G, Tzanetou D. (DT4Smart) discrete event simulation and Digital Twins in warehouse logistics: a bibliometric and content analysis-based systematic literature review. International Journal of Computer Integrated Manufacturing 2024;1-28.

[16] Babulak E, Wang M. Discrete Event Simulation: State of the Art. In: Aitor G, editor. Discrete Event Simulations. Rijeka: IntechOpen; 2010.

[17] Rashidi Z, Rashidi Z. Evaluation and Ranking of Discrete Simulation Tools. Journal of Electrical and Computer Engineering Innovations 2016;4(1):69-84.

[18] Collins AJ, Sabz Ali Pour F, Jordan CA. Past challenges and the future of discrete event simulation. The Journal of Defense Modeling and Simulation 2023;20(3):351-369.

[19] Reinhardt H, Weber M, Putz M. A survey on automatic model generation for material flow simulation in discrete manufacturing. 52nd CIRP Conference on Manufacturing Systems 2019. p. 121-126.

[20] Lee Y-TT, Riddick FH, Johansson BJI. Core Manufacturing Simulation Data – a manufacturing simulation integration standard: overview and case studies. International Journal of Computer Integrated Manufacturing 2011;24(8):689-709.

[21] Organization SIS. Standard for: Core Manufacturing Simulation Data - UML Model. 2010.

[22] Organization SIS. Standard for: Core Manufacturing Simulation Data - XML Representation. 2012.

[23] Bergmann S, Stelzer S, Wüstemann S, Strassburger S. Model generation in SLX using CMSD and XML stylesheet transformations. Proceedings of the 2012 Winter Simulation Conference 2012. p. 1-11.

[24] Bergmann S, Stelzer S, Straßburger S. Initialization of simulation models using CMSD. Winter Simulation Conference 2011. p. 2223-2234.

[25] Bergmann S, Strassburger S. On the use of the Core Manufacturing Simulation Data (CMSD) standard: experiences and recommendations. Fall Simulation Interoperability Workshop 2015.

[26] Steinbacher LM, Düe T, Veigt M, Freitag M. Automatic model generation for material flow simulations of Third-Party Logistics. Journal of Intelligent Manufacturing 2023.

[27] Barnes M, Finch EL. COLLADA–digital asset schema release 1.5.0. Specification, Khronos Group, Apr. 2008.

[28] Zhao J, Aghezzaf E-H, Cottyn J. A framework for evaluating a generic virtual commissioning data model. 33rd CIRP Design Conference 2023. p. 284-289.