Applying IEEE 1278.1-2012 Concepts to Support Integration of Digital Twins in Industrial Applications

1st Anna Florea
Automation Technology and Mechanical Engineering
Tampere University, Tampere, Finland
Department of Mechanical and Industrial Engineering
Norwegian University of Science and Technology
Trondheim, Norway
https://orcid.org/0000-0001-6734-1203

2nd Andrei Lobov
Department of Mechanical
and Industrial Engineering
Norwegian University
of Science and Technology
Trondheim, Norway
https://orcid.org/0000-0003-2729-489X

3rd Tatiana Minav
Automation Technology and
Mechanical Engineering
Tampere University
Tampere, Finland
tatiana.minav@tuni.fi

***Abstract*—Digital twins serve as a source of new insights about industrial processes and systems performance and enable the use of cutting-edge technologies for their optimization. Solution vendors offer a variety of tools for digital twin implementation. As the amount of such solutions grows, the need to integrate and navigate the variety of digital twins in large-scale systems arises. The complexity of modern industrial systems requires an approach, where the integration process will happen in an organized way, allowing engineers to make informed decisions and communicate clearly project goals and transfer them into actual design and solution.**

This work presents results from investigating the applicability of concepts, building blocks, and engineering processes for Distributed Interactive Simulation (DIS) standards to perform such tasks. It provides an example of integration between two simulation tools used for digital twin development and illustrates how DIS aligns with such a development process.

***Index Terms*—distributed interactive simulation, digital twin, warehouse operations**

I. INTRODUCTION

Modern industrial enterprises represent complex landscapes, composed of heterogeneous information sources, that must be dynamically integrated for near real-time decision making. For handling this degree of complexity, one requires an adequate information system with short reaction times. Such technology is being developed around the concept of Digital Twins (DT) [1].

Implementing the true two-way communication between physical and virtual entities, and across other information systems [2] is essential for DTs. Ultimately, DT is seen as an Artificial Intelligence (AI) assisted agent-driven socio-technical platform, capable of sensing, optimizing, learning, simulating, and allowing end-user engagement, while providing high integration of life-cycle and supply-chain information [3]. Such complexity calls for a revised approach to DT design and implementation. In order to address these needs, a structured approach to DT management, including interactions, integration, and composition, is required [4].

Potential instruments to address this challenge can be found in the field of system of system engineering [5]. Some prominent tools for design and implementation of such systems are collected under a group of IEEE standards and practices for Distributed Interactive Simulations (DIS) [6], [7].

Earlier research found these tools to have unnecessarily strict requirements for the case of manufacturing [8], and the gap between IT standards and industry-specific software being too big for structured approach to distributed plant simulation and control [9]. However, since then, the complexity of systems increased dramatically, as well as capabilities of modern industrial information systems, built with state of the art technology. Therefore, it is important to re-evaluate applicability of these standards in current context.

This paper presents results from evaluation of suitability of standards for Distributed Interactive Simulations for implementation of digital twin integration projects in industrial context. The rest of the paper is organised as follows: Section II provides brief overview of DIS and history of its application in industrial context; Section III introduces evaluated standards and their purpose, as well as case study and related tools; Section IV provides the overview of Distributed Simulation Engineering and Execution Process (DSEEP) and its relation to DIS, as well as mapping between the two and the actual process of case study development; Section V describes building blocks of DIS and how the case study can be expressed using those blocks, followed by Section VI, presenting the implemented solution based on APIs available from chosen simulation tools, and how DIS could support engineering process in practice. Paper concludes with Section VII summarising the findings, challenges, and providing directions for future work.

This work is partially supported by K.F. and Maria Dunderberg Foundation and Neles OY 30 Years Foundation grants.

II. DIGITAL TWIN INTEROPERABILITY AND COMPOSITION

Increasing amount of DT implementations brings new opportunities as well as challenges. One of which is integration or, rather, composition of digital twins. To make the integration between DTs possible, it must be embedded into design of an individual DT.

Realising the importance of this feature, Digital Twin Consortium [10] released in 2021 a white paper titled "Digital Twin System Interoperability Framework" [11]. The framework suggests seven concepts to assist scalability and interoperability of future digital twin systems: *(1)System-Centric Design*, *(2)Model-Based Approach*, *(3)Holistic Information Flow*, *(4)State-Based Interactions*, *(5)Federated Responsibilities*, *(6)Actionable Information*, and *(7)Scalable Mechanisms*. Some of the principles proposed in the document, are inspired by High Level Architecture (HLA) framework [12], developed by US Department of Defence to support consistent implementation of component-oriented distributed simulations [9].

When considering industrial applications, RAMI4.0 [13] plays central role in providing structured view on Industry4.0, characterised, among other, by cross-layer communication and distributed nature of function implementation. In context of DT research, RAMI4.0 is seen as a tool helping to construct DTs capable to incorporate changes in Industrial Internet of Things (IIoT) layer along systems life cycle [14], thus supporting implementation of communication between physical and virtual entities.

Both RAMI4.0 and HLA, were conceived as reference models/architectures, and capture a wide range features and operation principles necessary to be followed in modern systems to meet domain's needs. However, they provide little insight into practical aspects of implementation of such systems.

The gap is recognised, and addressed by research and industrial community in a variety of ways. For example, [14] proposes to use a familiar and widely adopted Enterprise Architecture Framework for practical implementation of RAMI4.0. A lot of research effort is also centered around the practical approach to Asset Administration Shell (AAS) [15], virtual representation of I4.0 assets necessary for establishing communication between I4.0 components.

HLA, and its potential use in industrial context was first evaluated in 2000s. Bandinelli et al. [9] identified several challenges, preventing adoption of HLA for industrial systems at the time, including the following:

- the information routing for most cases was expected to be broadcast, and industrial communication networks were not ready for such amount of data;
- while industrial systems were shifting from centralised to distributed architectures, managing information ownership posed a great challenge, lacking both technical and organisational tools to assist the process;
- finally, off-the-shelf software would often lack the means to implement or incorporate distributed and heterogeneous solutions.

In 2007, Iannone et al. [8], proposed an architecture for multi-model synchronisation in distributed simulations for supply chain management. It found HLA not suitable for manufacturing and supply chain simulation, since it was intended for cases where thousands of elements were expected to reliably exchange data over network in real time, while such a requirement did not exist for the above mentioned systems at the time.

Set of standards for Distributed Interactive Simulations (DIS) is mentioned in [5] as one of the most complete and detailed implementation of principles suggested by HLA, covering engineering tools, communication, and application protocols. It is concerned with the simulation of warfare environments for military training exercises, making its transfer to a different field a challenging task.

However, recent developments in the field of digital twin composition suggest that DIS may be a powerful instrument to support implementation of DT engineering tools. For example, Autiosalo et al. [16], present an approach to DT integration by expressing component capabilities and system boundaries though APIs based on open web standards. The Digital Twin Capabilities Periodic Table (CPT) by Digital Twin Consortium [17] provides framework for requirements definition for DT development. By providing both engineering process recommendations and detailed descriptions of application protocols and component interactions in different scenarios, DIS supports structured approach to Digital Twin Integration and API Services, as understood in *Category 2: Integration*, *Capability 20* and *Capability 22* of CPT respectively.

The objective of present work was to look beyond domain terminology of DIS and see how core concepts and architectural components can be used to represent industrial scenarios, and identify how tools and techniques derived from DIS can be further incorporated into engineering tools for digital twin development.

III. APPROACH AND METHODOLOGY

The research was organised in two parallel tracks. The first one investigated DIS contents, and the second on concentrated on building a case study. The study was constructed to involve actual engineering tools and a scenario, allowing to illustrate the concepts and derive research road map for incorporating DIS into engineering tools for industrial digital twins. This section provides brief summary of the research design, selected tools and the case study.

A. IEEE Standards for DIS

The core documentation suite related to DIS consists of five documents providing both technical and organisational guidance for implementation of distributed interactive simulations. In this work, the main focus is placed on tools supporting technical design and engineering process that may provide practical support for developers involved in implementation of digital twin integration projects. For this purpose, IEEE Std 1278.1-2012 Distributed Interactive Simulation — Application Protocols [7] and IEEE Std 1730-2010 IEEE Recommended

Practice for Distributed Simulation Engineering and Execution Process (DSEEP) [6] were selected as primary targets for evaluation.

The evaluation process followed top-down approach starting with evaluation of general applicability of the concepts, on which the DIS design is built, its conceptual architecture and building blocks, followed by available scenarios framed as protocol families of [7]. The engineering practices defined in [6] were evaluated at this stage of research primarily to trace how conceptualised designs evolve into actual engineering artefacts, and what tools are suggested to assist the information flow in this process.

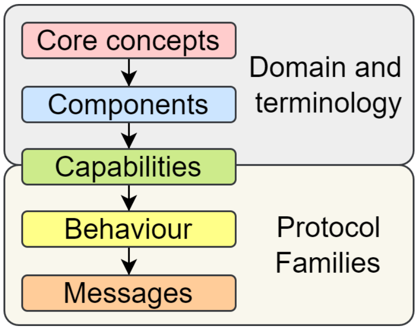


Fig. 1. Scope and structure of IEEE 1278.1-2012

Conceptually, the contents of DIS - Application protocols can be split in two major groups: *Domain Terminology* and *Protocol Families*, as illustrated in figure 1. The first category defines the domains core concepts, major components and their relations. The second category of information provided by the standards illustrates how various scenarios associated with the application domain (i.e. military training exercise) can be further implemented with said components through detailing underlying processes and necessary message exchange.

B. Simulation tools

In order to reflect diversity and heterogeneity of simulation tools and aspects of industrial applications, two different simulation tools were selected to conduct the study.

Figure 5 presents their simulation controls and visualization UI. Purpose and core features of each tool are described below:

- *Visual Components (VC)* [18]
  This manufacturing simulation tool allows to represent manufacturing process, where *processes* can be defined to mark actions upon materials and parts leading to creation of final product. Processes and transport operations are implemented via *resources*, and *flows* engage resources to move parts between processes(Figure 2(a)).
- *Mevea* [19]
  This suite consists of several applications, including Modeller and Solver. It allows physics based modelling of machines, reflecting how performed operations and interactions with the environment affect the performance of the constituent parts.

There are several features making these tools suitable for the case study:

- ability to connect to actual physical equipment via standard industrial communication protocols;
- availability of APIs enabling customisation of model components;
- availability of APIs for development of custom extensions and integration with 3rd party software.

Both tools provide 3D visualisation feature. However, while it is central to workflow of creating simulations in Visual Components, it is optional when running simulation in Mevea Solver, giving certain degree of flexibility in approach to simulation execution.

C. Case study

In order to illustrate how DIS concepts transfer into industrial applications, a case study has been developed where two digital twin development tools are integrated. The objective of the study was to evaluate readiness of API offered by DT development tools for such integration projects, and identify road map for implementation of DIS-based integration tools.

Case study is built around a human-driven forklift operating at a warehouse. Activity diagram, illustrating involved resources and their interactions is presented in Figure 3. A robot arm loads boxes onto a pallet, which is then transferred via conveyor to the pick up point. There it is picked by a human-operated forklift and transported to a shelf in storage area.

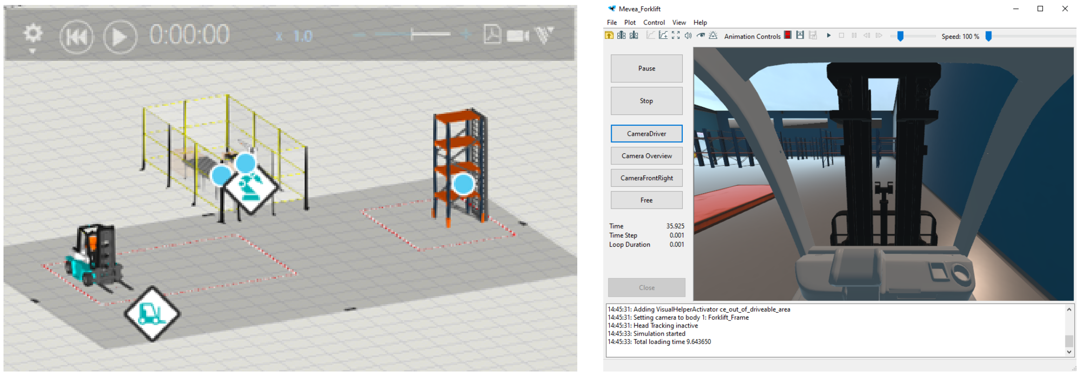
There are two simulations interacting with each other: one application represents the entire material handling process, while another focuses on dynamic simulation of the forklift. Such approach is expected to allow implement Key Performance Indicators (KPI) affected by transportation tasks in scalable and more accurate fashion. Current research phase focused on the case development and actual KPI implementation is reserved for the next stage.

IV. ENGINEERING PROCESS FOR DIS

IEEE 1730-2010 introduces a seven-step engineering process to support implementation of DIS projects. Each *step* contains two to four *activities*. Objectives, inputs and outputs, as well as actors are specified for each activity. The process is intended to be iterative with possibility to return and improve on any step or activity as project progresses until objectives are met.

Additionally to the general DSEEP, the document provides the mapping to support engineering in projects based on HLA and DIS accordingly. Table I provides summary of process steps for general DSEEP, its mapping to DIS and how these stages translate to the case study development.

The progress of the presented work in terms of DSEEP can be observed from the table. The implementation extended in several *test runs* to evaluate technical feasibility of the approach, but further iterations are necessary to achieve stage where gains in terms of business objectives can be presented and analysed.



```
Time          35.925
Time Step      0.001
Loop Duration  0.001
```

```
14:45:31: Adding VisualHelperActivator ce_out_of_driveable_area
14:45:31: Setting camera to body 1: Forklift_Frame
14:45:31: Head Tracking inactive
14:45:33: Simulation started
14:45:33: Total loading time 9.643650
```

(a) Visual Components. Resources in flow view

(b) Mevea Solver

Fig. 2. Simulation tools

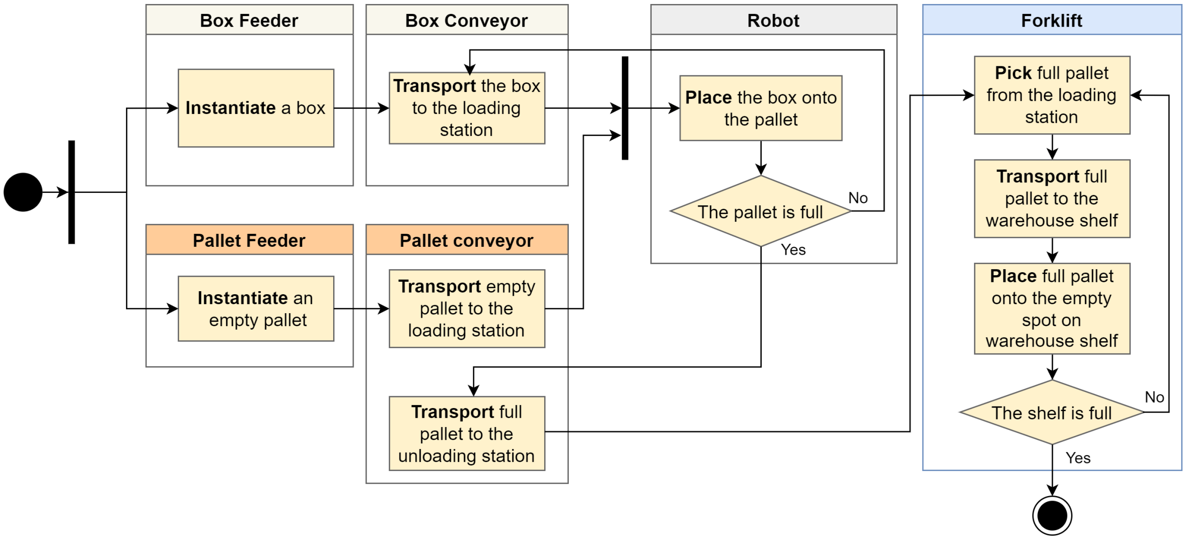


Fig. 3. Activity diagram of material handling implemented in Visual Components

Current progress towards case study implementation can be expressed mainly as early iterations over steps 1-6 of DSEEP or 1-3 of DIS. The content and scope of each phase is further illustrated in figure 4:

- **(1)DSEEP - (1)DIS – Scenario**
  This step can be thought of as business case. It defines the purpose and objectives of the DT integration project, which can be mapped to the DIS concept of *Simulation Exercise* (see Section V). There are two aspects to this phase. One is concerned with the business objectives, related KPIs, and how they can be implemented through the integration project. The second part is related to the actual execution, and how the daily working process around the KPIs can be implemented, and what tools the system user will use to accomplish the task.
- **(2-5)DSEEP - (2) DIS – Modelling&Implementation**
  - *Modelling* This phase concerned with inventory and development of models for the integration project. The models are the simulation models run by *simulation applications*. The modelling may involve development of necessary CAD models, parametrization and configuration. It may also include some basic scripting to define component behaviour. However, at this stage, the scripting is scoped to single simulation application. In iterative transitions between modeling and implementation stages the models gradually mature to adequately represent the process of interest.
  - *Implementation*
    In context of the case study it refers to the design and development effort related to extension of functionality of selected software to allow information exchange and synchronised execution. This phase is critical for developing structured approach to the technical implementation. DIS architecture and information model becomes a powerful tool, once extensibility of constituent applications, as well as completeness and limitations of their APIs is understood.
- **(6)DSEEP - (3)DIS – Test runs**
  This step represents the phase, where implemented system is taken into use and operates in real-life (production)

environment. Since the iterative nature of the process, in context of the case study, first test of implemented interaction between simulators can be linked to this step.

- **(7)DSEEP - (4-5)DIS – Feedback for case study development**
  DIS differentiates between exercise review and reporting to the decision makers. Step 4 refers to rapid assessment of the outcomes, and step 5 refers to elaborate reporting. In industrial context it would include both technical evaluation as well as degree to which business objectives where achieved.

TABLE I
DSEEP TOP LEVEL PROCESS VIEW

DSEEP	**DIS**	**Presented study**
1 Define Simulation Environment Objectives	1 Plan the exercise and develop requirements	Scenario. Customer case
2 Perform Conceptual Analysis	2 Design construct and test exercise	Modelling
3 Design Simulation Environment		Implementation
4 Develop Simulation Environment		
5 Integrate and Test Simulation Environment		
6 Execute Simulation	3 Conduct exercise	*Test runs*
7 Analyze data and Evaluate results	4 Conduct exercise review activity	*Evaluate technical feasibility*
	5 Provide results to decision maker	*Analyse results and refine case development*

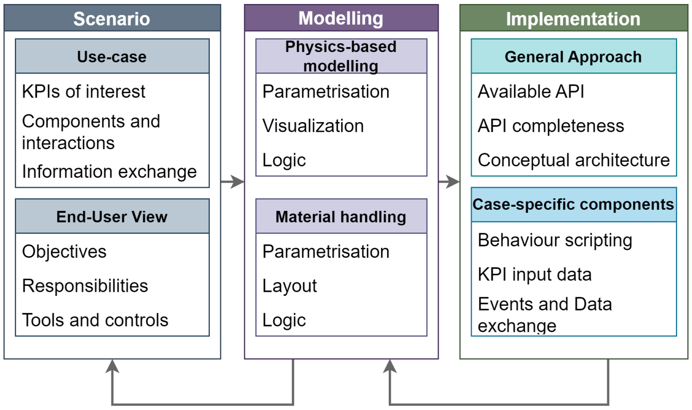


Fig. 4. Dimensions of case study development.

V. DIS: CONCEPTS AND BUILDING BLOCKS

This section introduces core concepts and building blocks of DIS and how they can be used to express the case study. This section focuses mainly on basic architectural principles of DIS as introduced in section 1.6.2 of IEEE 1278.1-2012 and definitions from section 3.1 of the standard.

Section 3.1 of IEEE 1278.1-2012 provides a total of 53 definitions. They can be grouped in the following categories: *fundamental concepts*, *architecture components*, *Live-Virtual-Constructed (LVC) triad*, *simulation exercise*, *simulation and simulation modes*, *entities (including operations, components and behaviour)*, *objects*, *data*, and *time*. At the current stage of research main focus was placed on the first two categories, and parts related to simulation exercise and simulation.

A. Core concepts

Distributed Interactive Simulation (DIS) is defined as follows (with parts relevant to DT integration for industrial applications emphasized by authors of this work):

> "A time and space coherent synthetic representation of world environments designed for linking the *interactive, free-play activities of people* in operational exercises. The *synthetic environment* is created through *real-time exchange of data units between distributed, computationally autonomous* simulation *applications* in the form of simulations, simulators, and instrumented equipment interconnected through standard computer communicative services. *The computational simulation entities may be present in one location or may be distributed geographically*." [7, p. 12]

The operational exercise can be seen as operations in industrial context, where different DTs will be present to represent systems realising the process in an interactive manner. The definition specifically mentions interactive free-play activities which reflects the nature of interaction sought after in digitalised industrial systems. As far as implementation is concerned, DIS definition describes how distributed nature reflects in organising computational resources and software components, where software components may be distributed over computational resources, and computational resources may be distributed geographically. It also important, that it relies on real-time exchange or data and use of standard communication interfaces. Overall nature of application scenario, degree of complexity and key characteristics of such an application (i.e. DIS) provides motivation for studying applicability of the IEEE 1278.1-2012 as basis for DT integration.

Another concept central to DIS, and being the main focus of the standard is *Protocol Data Unit (PDU)*, which is defined as:

> "A ... data message that is passed on a network between simulation applications according to a defined protocol." [7, p. 15]

This concept may serve the foundation for common approach to interaction between DT. To allow easy interaction between DTs, an individual implementation must consider which data it will expose, and which inputs would be available from other applications, and then implement those in some form of PDUs. The PDUs are grouped into protocol families implementing information exchange in different scenarios as defined by functional areas of DIS.

Some protocol families, such as *Simulation Management*, *Synthetic Environment*, and *Entity Management*, can be followed almost directly in industrial applications, while others (e.g. *Information Operations*) might be irrelevant in most cases. And there is a third group, lying in the middle, where covered scenarios resemble some of the processes and behaviours occurring in industrial context, but those are hidden behind application domain terminology (e.g. *Logistics*).

Additionally to the main definition of PDU, DIS introduces several additional types to ensure that all communication patterns and special cases foreseen in DIS are covered(i.e. *transient*, *state*, *supplemental* PDUs).

B. Elements of DIS architecture

Section 1.6.2 of IEEE 1278.1-2012 lists a total of six characteristics of DIS architecture, of which the four below are relevant in context of modern industrial applications:

> "a) No central computer controls the entire simulation exercise ..." [7, p. 3]
>
> "b) Autonomous simulation applications are responsible for maintaining the state of one or more simulation entities ..." [7, p. 3]
>
> "d) Changes in the state of an entity are communicated by simulation applications ..." [7, p. 3]
>
> "e) Perception of events or other entities is determined by the receiving application ..." [7, p. 3]

The five major concepts constructing a DIS system are: *simulation exercise*, *simulation environment*, *host computer*, *simulation application*, and *simulation entity*. An example of an exercise structure presented in terms of DIS building blocks is presented in Figure 5(a).

A simulation exercise is implemented as a collection of simulation applications running on host computers. A host computer can run one or more simulations simultaneously, belonging to same or different exercises. Simulation applications related to the same exercise share the same Exercise ID. Simulation applications represent a part, one or many simulation entities (a physical object represented in the synthetic environment) and exchange data about simulation entities via DIS PDUs. Simulation entities are being created and managed by simulation application and are affected by exchanged PDUs. Simulation environment refers to the conditions and operational environment around simulation entities.

In addition to the concepts above *simulation management* function and *simulation manager* role are important for implementation. Simulation management would assist with composition management at the highest level. It is necessary for control of simulation exercise. Simulation manager, in its turn, is the simulation application performing simulation management functions.

Through its concepts, design principles and building blocks DIS suite supports implementation of different aspects of Digital Twin System Interoperability Framework. For example, instruments for implementing *State-based Interactions* are provided trough PDU families such as *Simulation Management*, *Synthetic Environment*, and *Entity Information/Interaction*, as well as characteristics *b* and *d* of DIS, introduced in the beginning of the section. *Actionable Information* feature is supported primarily in terms of *contextual semantic interoperability* through characteristic *e* of DIS, reliability as part of *trust and security* is assured through PDU families defined for communication with reliability, *determinism* is supported in individual PDU families where necessary for scenario implementation. *Scalable Mechanisms* are part of DIS architecture, with *IEEE 1730.1-2013 IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process Multi-Architecture Overlay (DMAO)* [20] providing additional detail on approach to integration between distributed simulations implemented using different standards (i.e. HLA and Test and Training Enabling Framework(TENA)). Two alternative designs suggested are *Gateway* and *Middleware* configurations.

Figure 5(b) illustrates how DIS concepts transfer to integration between DT in the case study, which in its turn can be seen as simulation exercise. The two DT development tools can be considered as simulation applications, while some of the simulated objects can be perceived as simulation entities (e.g. forklift or robot). However, the boundaries for a simulation entity may be difficult to grasp for the DT developer: in case of Mevea Solver, forklift is modeled as kinematic chain of constituent parts, and while it is represented visually as a cohesive object, its representation is spread over multiple nodes of the model tree.

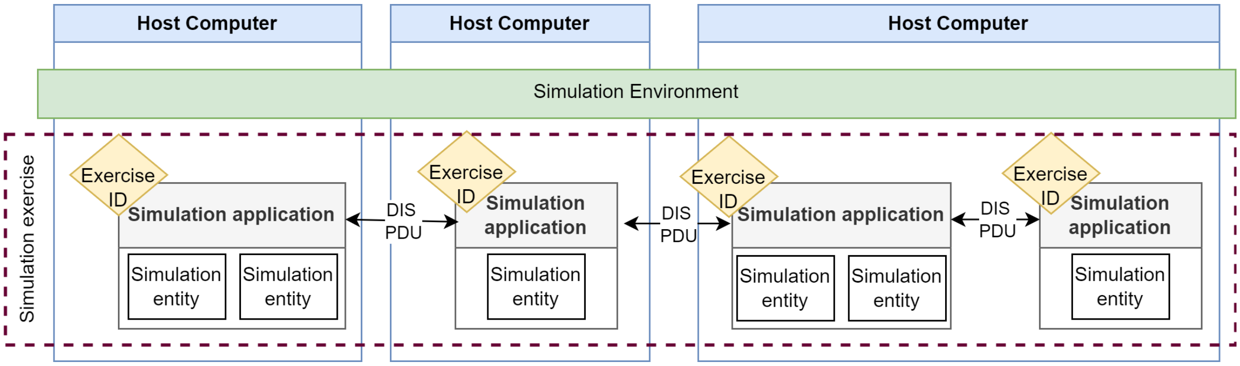
VI. IMPLEMENTATION

This section describes the implemented solution based on APIs available from chosen simulation tools and how DIS could support engineering process in practice.

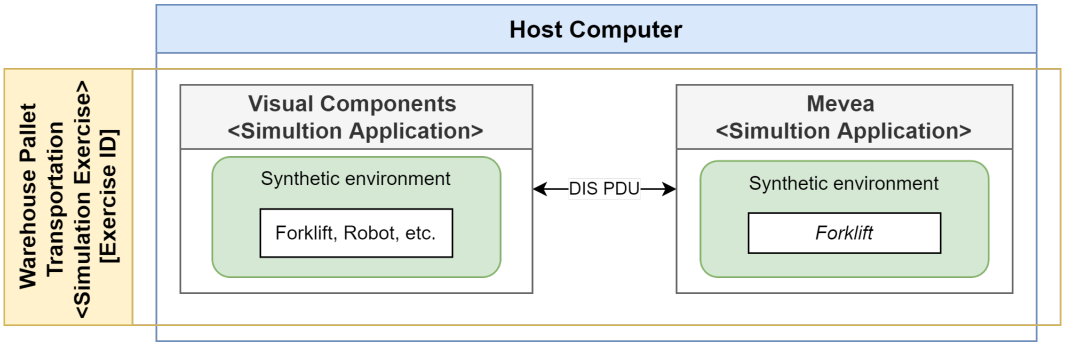
Figure 6 presents organisation of the components of interacting simulations. Each simulation uses a dedicated model representing the case study. The models are defined using VC and Mevea Modeller, and executed in VC and Mevea Solver.

Behaviour of some of the modeled components are customised with help of *Python API* available from each tool to assist implementation of accurate models and extend functionality beyond built-in components. For example, a script in forklift simulation is used to read virtual mass sensors of the forks to generate event notifying the VC simulation of successful pallet instantiation. In VC behaviour scripts are used to update custom parameters of the simulated resources (e.g. initializing and updating mass of the pallet as more boxes being added to it by the robot).

Interaction between simulations is managed via plugin developed for Visual Components using .NET APIs available from the software vendor. This API allows building custom extensions to the simulation tool itself, as opposed to behaviour scripts which belong to a specific simulation model. The plugin functionality relies on VC .NET API for accessing model components and manage simulation execution. Mevea Interface API provides built-in tools to establish TCP socket connection and exchange information with Mevea Solver during the simulation execution. The API provides functions to create, initialize and update the connection, as well as stop and close the connection.



(a) An example of distributed simulation in terms of DIS architecture components.



(b) Architecture of implemented system represented in DIS terms.

Fig. 5. DIS components

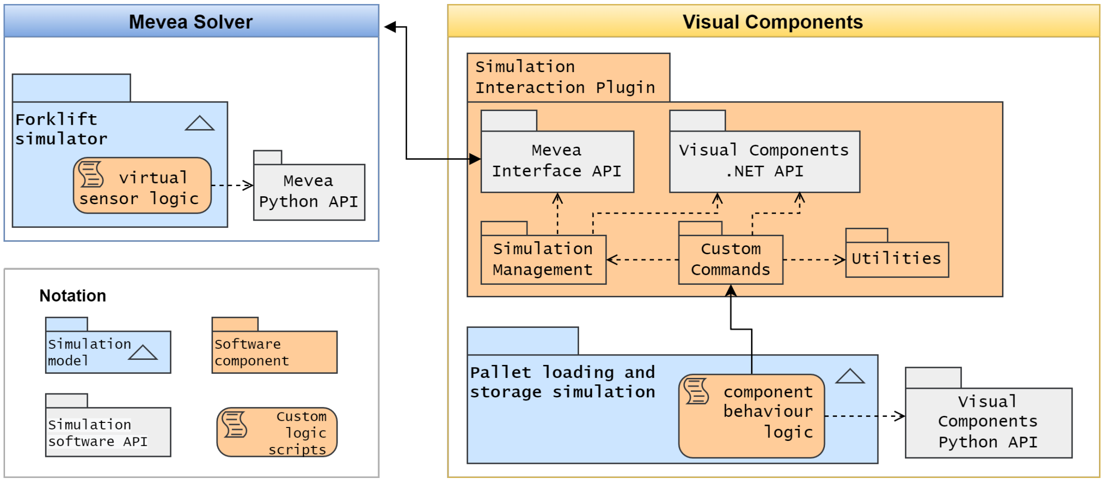


Fig. 6. Implemented integration solution based on software vendor APIs

Actions related to information exchange between simulations are wrapped as custom VC .NET commands, which can be called by the behaviour scripts of the components during simulation execution.This allows information exchange at relevant moments in response to key events marking progress of the process.

There are three distinct packages in the plugin, with well defined scope: *Simulation management*, *Custom commands*, and *Utilities*. Such clear separation of concerns allows identifying how DIS can help further advance the case study in structured manner. For example, *Simulation Management*, *Synthetic Environment*, and *Entity information/interaction* protocol families can help organise information exchange. The overall approach for framing protocol families, in combination with CPT supports implementation of custom commands.

Further work will be performed to refactor the implementation to reflect DIS PDU semantics as well as build better understanding of limitations of such approach, and how they can be addressed.

VII. CONCLUSIONS AND FUTURE WORK

Present work explored how standards for Distributed Interactive Simulations could support implementation of digital twin integration projects in industrial context. Core concepts and tools of IEEE 1278.1-2012 Application Protocols, and IEEE 1730-2010 Recommended Practice for Distributed System Engineering and Execution Process (DSEEP) were introduced and illustrated with a case study of forklift operating at a warehouse.

DSEEP provides support for engineering process, offering guidance for developing and sharing knowledge across units and among different stakeholders as project advances. The process can be adjusted to modern project management and development practices. Additional instrument to adapt to constrains and limitations emerging during integration is incorporated in Simulation Environment Agreement, resulting from Step 4 Activity 4.2 of DSEEP.

Core architectural components of DIS allow design and implementation of extensible systems with dynamic hierarchies, composed of diverse and distributed components to fulfil needs of modern industry.

DIS also provides strategies on approaching such challenging parts of the integration as time handling, representation of the environment and its change in response to events and operations progress.

Standard's approach to define functional areas and elaborate a tailored implementation of messaging in typical scenarios extensively and with much detail on message contents and structure can be followed to design and implement similar tools for industrial applications.

The standard provides very detailed implementation. However, its terminology is rather tight to the application domain. While core principles and architectural concepts can be applied directly in industrial context, the actual application of protocols implemented for various functional areas is a rather challenging task. Some protocol families can be followed almost directly in industrial applications, while others are irrelevant. And there is a third group, lying in the middle, where covered scenarios resemble some of the processes and behaviours occurring in industrial context, but those are hidden behind application domain terminology.

Digital Twin assumes bidirectional link between digital representation and actual physical entity. In a scenario, where two or more digital twin solutions need to be integrated, this would imply re-evaluation of the communication link implementation, and respective adjustments at physical and digital side. IEEE 1278.1-2012 primarily focuses on the digital (simulation) side of the solution, while taking into account diversity of data sources that could be fed into simulation.

Developed solution was implemented in a configuration, where two simulation applications run on same host computer. Further development is needed to evaluate scenarios where multiple applications run on the same computer, or in a more distributed manner. Such a scenario can be attained by scaling the case study to incorporate larger fleet of forklifts and increasing the capacity of the storage area.

Based on the findings from present case study, two major directions for future work were identified. The first concerns with in-depth exploration of DIS architecture and how it can assist the integration, taking into account diversity of approaches to extensibility in simulation tools. The second concerns with practical implementation of protocol families related to simulation and entity management, and further investigation of applicability of other domain specific scenarios.

REFERENCES

[1] M. W. Grieves, *Virtually Intelligent Product Systems: Digital and Physical Twins*, 2019, no. July.

[2] E. Negri, S. Berardi, L. Fumagalli, and M. Macchi, "MES-integrated digital twin frameworks," *Journal of Manufacturing Systems*, vol. 56, pp. 58–71, jul 2020.

[3] C. Boje, A. Guerriero, S. Kubicki, and Y. Rezgui, "Towards a semantic Construction Digital Twin: Directions for future research," p. 103179, jun 2020.

[4] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the Digital Twin: A systematic literature review," *CIRP Journal of Manufacturing Science and Technology*, vol. 29, pp. 36–52, may 2020.

[5] L. B. Rainey and A. Tolk, *Modeling and Simulation Support for System of Systems Engineering Applications*, L. B. Rainey and A. Tolk, Eds. Wiley, dec 2014.

[6] IEEE, *IEEE Std 1730-2010 (Revision of IEEE Std 1516.3-2003). IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP).* IEEE, 2011.

[7] ——, *IEEE Std 1278.1-2012 (Revision of IEEE Std 1278.1-1995). IEEE Standard for Distributed Interactive Simulation— Application Protocols.* IEEE, 2012.

[8] R. Iannone, S. Miranda, and S. Riemma, "Supply chain distributed simulation: An efficient architecture for multi-model synchronization," *Simulation Modelling Practice and Theory*, vol. 15, no. 3, pp. 221–236, mar 2007.

[9] R. Bandinelli and M. Tucci, "IT Solutions Advantages and Drawbacks for Different Level of Plant Distributed Simulation, Planning and Control," *IFAC Proceedings Volumes*, vol. 37, no. 4, pp. 455–460, apr 2004.

[10] (2020) Digital twin consortioum. [Online]. Available: https://www.digitaltwinconsortium.org/

[11] A. Budiardjo and D. Migliori, "Digital Twin System Interoperability Framework," 2021. [Online]. Available: https://www.digitaltwinconsortium.org/pdf/Digital-Twin-System-Interoperability-Framework-12072021.pdf

[12] IEEE, *IEEE Std 1516-2010 (Revision of IEEE Std 1516-2000). IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)– Framework and Rules.* IEEE, 2010.

[13] (2022) Rami 4.0 - isa. [Online]. Available: https://www.isa.org/intech-home/2019/march-april/features/rami-4-0-reference-architectural-model-for-industr

[14] C. Binder, W. Leiter, O. Joebstl, L. Mair, C. Neureiter, and A. Lüder, "Utilizing an enterprise architecture framework for model-based industrial systems engineering," in *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*, 2021, pp. 1–6.

[15] X. Ye and S. H. Hong, "Toward Industry 4.0 Components: Insights into and Implementation of Asset Administration Shells," *IEEE Industrial Electronics Magazine*, vol. 13, no. 1, pp. 13–25, mar 2019.

[16] J. Autiosalo, R. Ala-Laurinaho, J. Mattila, M. Valtonen, V. Peltoranta, and K. Tammi, "Towards Integrated Digital Twins for Industrial Products: Case Study on an Overhead Crane," *Applied Sciences*, vol. 11, no. 2, p. 683, jan 2021.

[17] (2022) Digital twin periodic table. [Online]. Available: https://www.digitalplaybook.org/index.php?title=Digital_Twin_Periodic_Table

[18] (2022) Visual components - 3d manufacturing simulation software. [Online]. Available: https://www.visualcomponents.com/

[19] (2022) Digital twin software and simulation solutions for ports, construction and mining — mevea. [Online]. Available: https://mevea.com/

[20] IEEE, *IEEE Std 1730-2010 (Revision of IEEE Std 1516.3-2003). IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process Multi-Architecture Overlay (DMAO).* IEEE, 2013.