A Comparison of the CSPI and CMSD Standards

Steffen Strassburger
School of Economic Sciences
Ilmenau University of Technology
Helmholtzplatz 3
98693 Ilmenau, GERMANY
steffen.strassburger@tu-ilmenau.de

Simon J E Taylor
ICT Innovation Group
Department of Information Systems and Computing
Brunel University
Uxbridge, Middx, UB8 3PH, UK
simon.taylor@brunel.ac.uk

Keywords:
Discrete-event Simulation, Distributed Simulation, COTS Simulation Package, Core Manufacturing Simulation Data, Interoperability, Standards.

ABSTRACT: *The SISO standards SISO-STD-006-2010 and SISO-STD-008-2010 are two standards focusing on the interoperability requirements of the manufacturing and logistics domains. In these domains, simulation studies are mostly performed using commercial-off-the-shelf (COTS) Simulation Packages (CSPs), such as Arena™, Anylogic™, Flexsim™, Plant Simulation™, Simul8™, SLX™, Witness™, etc. As both standards seem to address similar problems, namely interoperability issues when using such CSPs, we investigate the specific strengths of each of these standards and derive recommendations for their usage. In specific, we illustrate how the standard for Commercial Off-the-Shelf (COTS) Simulation Package Interoperability (CSPI) Reference Models (SISO-STD-006-2010) can facilitate interoperability between different CSPs. A use case from the commercial vehicle sector is presented in support of this discussion. Further, we illustrate how the standard for Core Manufacturing Simulation Data (SISO-STD-008-2010) can be used to exchange data between other information systems in manufacturing and CSPs. We outline a use case in which CMSD is used as the basis for simulation model generation in Plant Simulation. We also discuss the usage of CMSD to carry model initialization data and its potential to exchange data between different CSPs.*

1. Introduction

With the advent of the High Level Architecture for Modeling and Simulation (HLA) in the 1990s, the topic of simulation interoperability has become an important focus point for simulation researchers and practitioners. Although the success of the HLA in non-military applications has been controversially discussed [1, 2], and the military training simulation domain as one of HLAs ancestors has developed alternative solutions [3], HLA's positive impact on the attention towards simulation interoperability requirements outlasts.

In this article, we focus on simulation interoperability requirements of simulations conducted in the manufacturing sector. In this area, the most frequently used simulation paradigms are all variants of the well-known discrete event simulation paradigm [4]. The usage of commercial-off-the-shelf (COTS) Simulation Packages (CSPs) is commonplace. These CSPs offer intuitive modeling tools for creating material flow simulations of production facilities, supply chains, logistics processes, etc.

CSPs were traditionally designed for stand-alone use. A typical simulation study would follow one of the traditional simulation project life cycles [5], which typically include a data collection phase, a model creation phase, and an experimentation phase, following each other more or less sequentially. Already implied in such a life cycle is the disposal of the simulation model after the end of the study.

In today's world, such a use of simulation often does not reflect the requirements that one may have towards reducing the costs of the study (e.g. by re-using

existing simulation models/components) and the timeliness of simulation results (e.g. by feeding online data from real processes as input into the simulation). Towards these requirements, the standardization of interfaces for facilitating interoperability of simulation models is a crucial factor.
Both SISO standards SISO-STD-006-2010 and SISO-STD-008-2010 are standards focusing on these interoperability requirements. As both standards seem to address similar problems, this article investigates the specific strengths of each of these standards and derives recommendations for their usage.
The remainder of this article is structured as follows. Section 2 introduces both standards. Section 3 outlines two practical examples and derives recommendations towards the usage of both standards. Section 4 provides a critical discussion of both standards. Section 5 summarizes this paper.

2. Overview of the Standards

2.1 SISO-STD-006-2010

The SISO Standard for Commercial Off-the-Shelf (COTS) Simulation Package Interoperability (CSPI) Reference Models (SISO-STD-006-2010) [13] has been developed to support interoperability between two or more different CSPs.
As the aforementioned CSPs do not support converting models from one CSP's native format into another (say: a Plant Simulation model into an Arena model), a natural way of composing models from reusable model components would be to form a distributed simulation (possibly based on the HLA standard), where the individual models in their native formats and their respective CSPs would become federates and interact appropriately at runtime.
The SISO Standard for CSPI Reference Models was designed to support this process. In specific, it is supposed to help to define/capture the interoperability requirements for forming such a distributed simulation with multiple CSPs.
The interoperability reference models (IRMs) defined by the standard describe these interoperability requirements on a semantic level. They are not concerned with the details of achieving a syntactic interoperability, which is assumed to be provided by some kind of interoperability middleware, such as the HLA RTI.
The IRMs are based on the observation that the interaction between models and their CSPs follows some very typical use patterns. The most common type of interaction is the exchange of entities. An entity in these CSPs is typically a non-active item (e.g. an order, a car body, a customer) which is moved through the simulation model, occupies resources, is stored in a buffer, etc. Therefore the most common requirement when coupling different CSPs is to exchange such entities in an orderly fashion. The first set of interoperability reference models (IRMs) defined in SISO-STD-006-2010 therefore describes the different styles in which such an entity transfer can take place.
Overall, the standard defines four different types of IRM representing the most common interoperability scenarios. These are[1]:

Type A:	Entity Transfer
Type B:	Shared Resource
Type C:	Shared Event
Type D:	Shared Data Structure

IRM Type A Entity Transfer deals with the requirement of transferring entities between simulation models, such as an entity *Part* leaves one model and arrives at the next.
IRM Type B Shared Resource refers to sharing of resources across simulation models. For example, a resource R might be a pool of workers that is shared between two models. In this scenario, when a machine in a model attempts to process an entity waiting in its queue it must also have a worker. Only if a worker is available the processing can take place. If not then work must be suspended until one is available.
IRM Type C Shared Event deals with the sharing of events across simulation models. For example, when a variable within a model reaches a given threshold value (a quantity of production, an average machine utilisation, etc.) it should be able to signal this fact to all models that have an interest in this fact (to throttle down throughput, route materials via a different path, etc.).
IRM Type D Shared Data Structure deals with the sharing of variables and data structures across simulation models. Such data structures are semantically different to resources, for example a bill of materials or a common inventory.
To illustrate the idea behind the IRMs, let us now take a closer look at IRM Type A. IRMs can have different subtypes, providing more detail on the specific interoperability requirements of a certain set of models. The most generic case of entity transfers is described in IRM Type A.1 General Entity Transfer (Figure 1).
It allows capturing the input and output elements, the entity type and some timing constraints, e.g. a transport time which can be greater or equal to zero.

[1] The description of IRMs is shortened in this paper for conciseness and was adopted from [6].

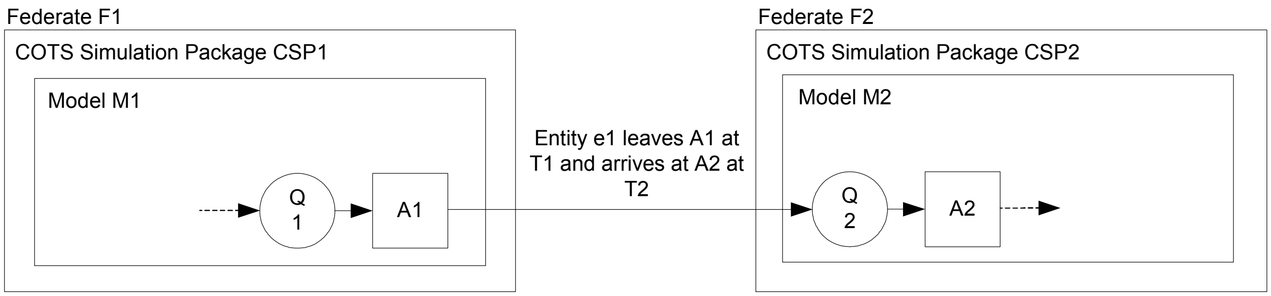

Figure 1: IRM Type A.1: General Entity Transfer [6]

However, there can be more interoperability requirements towards an entity transfer. For example, the receiving element (Q2) in M2 may be a bounded buffer. In this case, the transfer can only take place if there is sufficient buffer space, otherwise the exit element A1 in M1 cannot transfer the entity and must block. These requirements are captured in IRM Type A.2 Bounded Receiving Element (Figure 2).

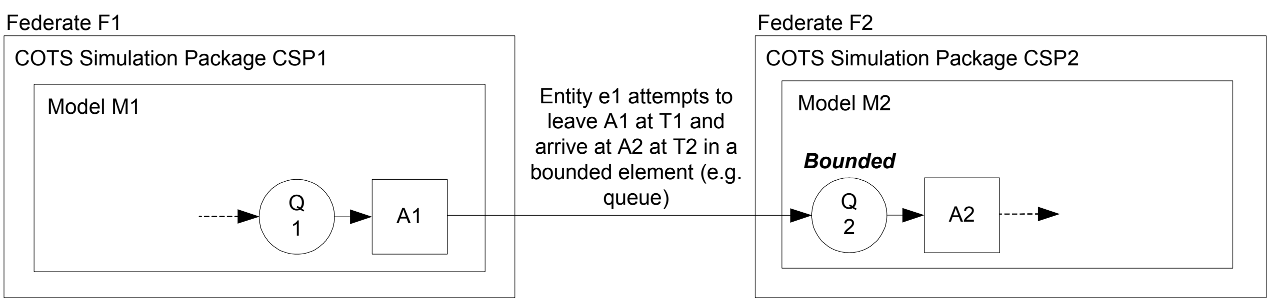

Figure 2: IRM Type A.2: Bounded Receiving Element [6]

A further requirement may occur when multiple models can deliver entities into the same buffer, potentially requiring a prioritization. This is captured in IRM Type A.3 Multiple Input Prioritization.

The list of IRMs defined in SISO-STD-006-2010 can be used in a cumulative way. They can, for instance be used as follows [6]:

- to clearly *identify* the model/CSP interoperability *capabilities* of an *existing* distributed simulation
  - e.g. The distributed supply chain simulation is compliant with IRMs Type A.1, A.2 and B.1
- to clearly *specify* the model/CSP interoperability *requirements* of a *proposed* distributed simulation
  - e.g. The distributed hospital simulation must be compliant with IRMs Type A.1 and C.1

Future extensions to the standard by adding further subtypes to these IRMs may be possible.

Current work in this area focusses on establishing guidelines and reference implementations how to “solve” the IRMs [11, 12].

In summary, SISO-STD-006-2010 is designed to describe interoperability *requirements* when connecting different CSPs at runtime into a distributed simulation. It can also be used to describe the *capabilities* of a specific technical solution for coupling CSPs. However, the standard does not suggest a specific way of *implementing* a solution to a specific IRM. It is envisioned that guidance in this area will be provided as a collection of best practices, e.g., of how to use a standard like HLA to implement a solution to a specific IRM. Work towards an entity transfer specification is an example for this effort [7, 8, 9, 10]

2.2 SISO-STD-008-2010

SISO-STD-008-2010 has been developed under the leadership of the National Institute of Standards and Technology (NIST).

While SISO-STD-006-2010 focusses on interoperability requirements *between different CSPs*, SISO-STD-008-2010 addresses the interoperability requirements *between CSPs and other information systems* in the manufacturing context.

The standard defines a so-called core manufacturing simulation data (CMSD) information model which provides a data specification for the efficient exchange of manufacturing data in a simulation environment [14]. Purposes of the CMSD standard are to:

- enable data exchange between simulation applications and other software applications
- support the construction of manufacturing simulators
- support the testing and evaluation of manufacturing software
- enable greater manufacturing software application interoperability [14]

Two different methods are used for representing the CMSD standard: the Unified Modeling Language (UML), and the XML representation based on RELAX NG and Schematron schema definitions. SISO-STD-008-2010 provides the UML representation, the XML representation is subject of another SISO standard which is currently in the balloting process.
The UML representation has been organized using packages shown in Figure 3.

simulation model generators for different CSPs have been implemented [17, 18].
CMSD is even capable of capturing initialization data appropriately (e.g. data describing the current system load) [17]. The latter is interesting in the context of online/symbiotic simulation, where a simulation model cannot start empty and idle, but must be initialized with the current state of the real system.

3 Case Studies

3.1 Usage of CSPI IRMs

Within the planning, design and redesigning of new production facilities at Deere & Co., a leading manufacturer for agricultural, forestry, and construction equipment, discrete event simulation models are routinely used to simulate the behavior of the systems under investigation [19].

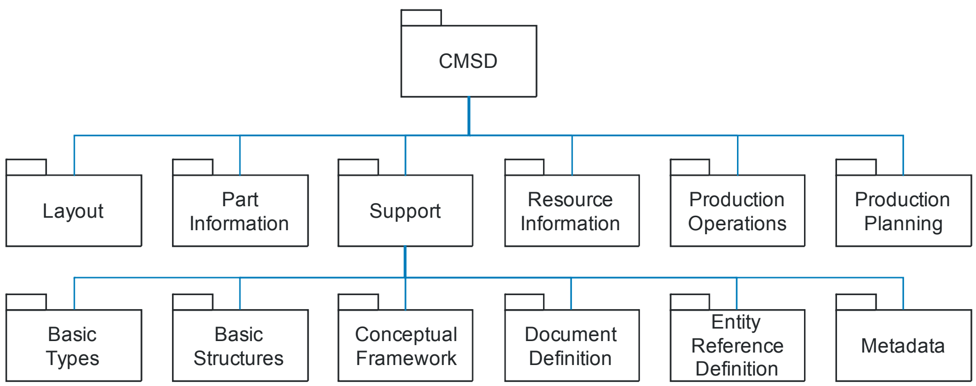

Figure 3: Packages of the CMSD Information Model [14]

The specific value of CMSD is its definition of a comprehensive data exchange format that suites both the needs of general manufacturing IT systems (e.g., ERP, MES) and a CSP [15, 16].
The CMSD standard provides data structures and an information model which were designed to firstly support the exchange of modeling information. To cover the complexity of production and logistic systems and a wide range of modeling approaches, the standard allows aspects of the system to be mapped in CMSD in multiple ways.
All CMSD classes which describe objects, like jobs, machines and so on can be extended by user defined properties.
With that, CMSD can easily be used for automatic simulation model generation based on data extracted from ERP systems such as SAP ERP.
CMSD standardizes all the data elements for describing the technical data and organizational data of a production system. Using this type of data, several

Dedicated solution sets exist for the simulation of certain types of production systems, like assembly lines or paint systems. These models are used for the planning of new factories as well as for supporting ongoing factory operations. The CSP used in the presented case study was SLX™.
The basis of the case study is a planned tractor factory in South America [11]. With a target production of 40 tractors per day, Monday through Friday, the production system under investigation consists of seven components, as shown in Figure 4: in total four pre-paint asynchronous assembly lines for chassis, transmissions and front axle assembly, two post-paint asynchronous assembly lines for cabs and tractors, and a wet-on-wet paint system. Each component of the production system consists of multiple manned work stations. The material flow of the overall production system is shown in Figure 4.
In the traditional application of simulation in the company, each of the production sections would be

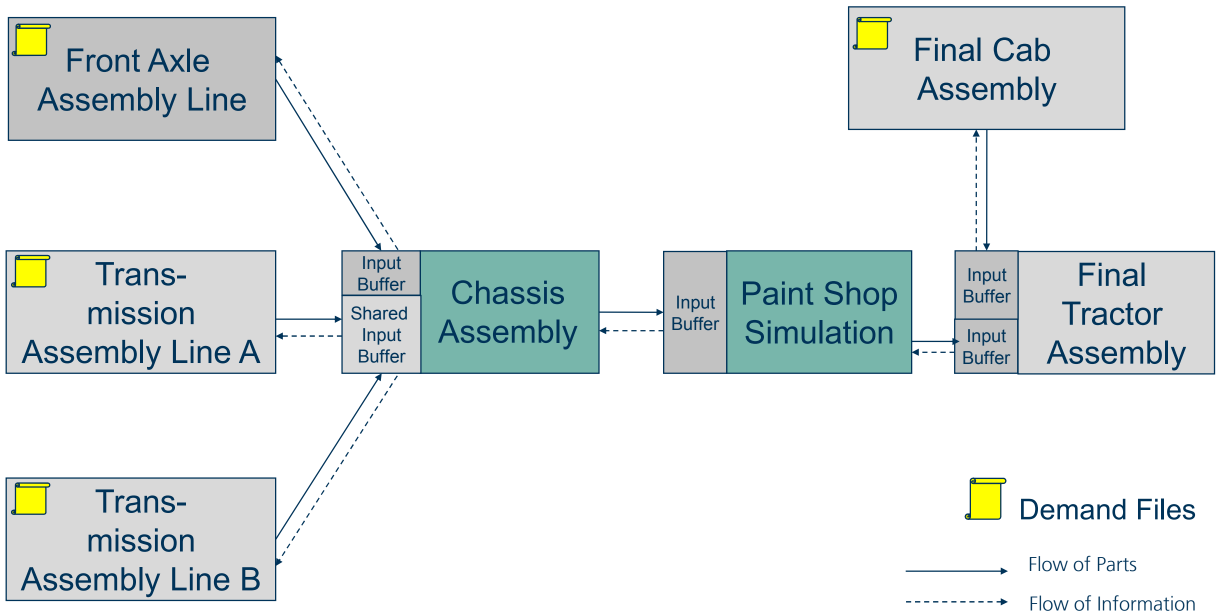

Figure 4: Material flow in the models of the distributed manufacturing simulation [11]

simulated using a separate simulation model using established simulation frameworks. This yields good results when each section is investigated separately.
It does, however, imply the usage of simplified assumptions about the input and output behavior of the different sections. In order to simulate and take into account interdependencies between the different production sections (like different shift regimes and the size of input buffers) correctly, an overall simulation of all relevant production sections was needed.
The motivation for using distributed simulation in this example was therefore the *integration* of independently developed existing models which cannot easily be combined within a single CSP for common execution.
The IRMs defined in [13] were used to identify the interoperability requirements in the feasibility analysis conducted between the simulation experts at Deere & Co. as the domain experts and the institution performing the integration of the models into a distributed simulation.
In this analysis it was determined that the main focus of the combination of the models was the correct implementation of the material flow. There are several types of entities which must be transferred between the models, all relating to parts which are the output of a certain system. Entity transfer had to take into account bounded input buffers in all of the production systems receiving parts. There is a travel time for parts between each of the sections. With one exception there is a unique input buffer for each of the sending models. In one case, there was a shared bounded input buffer into which two models deliver parts, but without any prioritization.
With the help of the IRMs it was therefore possible to define that the interoperability solution for the distributed manufacturing simulation had to be capable of implementing a IRM Type A.2 entity transfer with $T1<T2$ in all cases.
Details of the resulting interoperability solution can be found in [11].

3.2. Usage of CSMD for automatic model generation and initialization

To validate its applicability a CSMD based model generator for Plant Simulation (see Figure 5) was implemented [20].
The information flow was designed as closely as possible to real world scenarios of typical job shops. It is assumed that all information which is required to describe the production system and its state is captured in the relevant manufacturing applications (e.g., Enterprise Resource Planning Systems (ERP) and Manufacturing Execution Systems (MES)).
For the model generator certain classes from the most relevant CMSD packages were chosen. They include the resource class which stores information about machines and employees, the calendar class for storing shift and break information, the process plan class which stores detailed information about the manufacturing steps that are required for different part or job types. Furthermore the job class from the

productions operation package is used to model the production demand. CSMD also offers other ways to model production demand, e.g., by using the order class, but for simplicity we have chosen to model concrete jobs. A job contains a reference to its process plan, a release as well as a due date and therefore carries all information required to unambiguously simulate its flow through the production.

The decision about which CMSD class to use for which purpose somewhat depends on the specific needs and capabilities of the systems to be connected. A collection of best practices might be useful for future assistance in these cases.

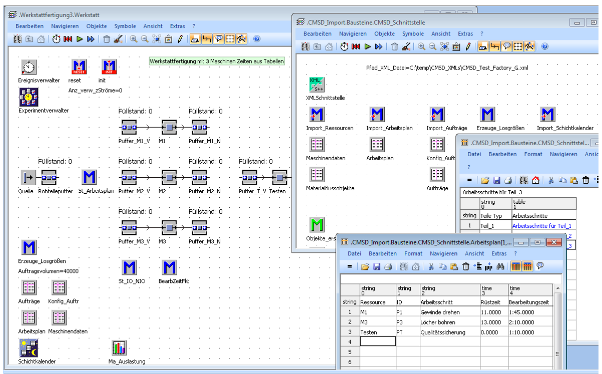

Figure 5: CMSD-based Model Generator for Plant Simulation

The developed model generator is furthermore capable of performing model initialization. This is mostly based on the same CMSD classes, but with the usage of additional (sometimes user-defined) attributes [17].

4. Discussion

The case studies discussed in section 3 clearly show the need for standards supporting interoperability of simulation in the manufacturing domain. However, one may be tempted to ask why interoperability of CSPs has to be so complicated. Do we really need more than one standard? Wasn't HLA designed as the solution to all these interoperability problems?

First of all, let us say in support of HLA that this standard solves many interoperability problems and that it has been a major step forward. By its impetus on not being limited to a specific niche of the simulation market, HLA has become a quite complex standard.

Several authors have therefore argued that many features of HLA are not needed for manufacturing simulation interoperability and pledged in favor of some self-developed proprietary solution. Although their argument may be true, we argue against the usage of proprietary solutions just to prevent having to deal with a complex standard.

The core problem of using HLA for creating true CSP interoperability is not its complexity as such, but its multitude of resulting options to solve a certain interoperability problem. A simple matter like implementing an IRM A.1 entity transfer can be performed in multiple ways in HLA. One solution might use HLA interactions modeling the entity transfer, another solution may be based on using HLA object instances and transferring them using HLA ownership management. Both solutions solve the interoperability problem, but are *not interoperable with each other*.

The original intent in creating SISO-STD-006-2010 was therefore to clearly identify the typical interoperability problems that must be solved when connecting different manufacturing or logistics models in different CSPs. In a second step, it was then envisioned to provide recommendations on the implementation of the resulting IRMs. This is now envisioned to be performed as a collection of best practice recommendations.

The preposition in creating this SISO standard was always based on the assumption that a distributed simulation is the only viable answer to solve the type of interoperability problems the group had in mind.

When we take a closer look at SISO-STD-008-2010 and its CMSD information model, one may be tempted to question this assumption. Wouldn't it be so much easier if we could simply save a certain native model (say: from Arena) into the CMSD exchange format and then recreate the model in the CSP of our choice (say: Plant Simulation)?

Unfortunately, life is not that easy. Using this approach may work for a certain percentage of the model. It may even work completely, but if and only if we can represent our entire model in CMSD. The same holds true for any other data exchange format. Experience with real-life applications, however, shows that most automatic model generation approaches will provide a nice 80-90% solution, but some manual model enhancement will very likely be needed.

This often relates to the dynamic of the model, as some behavior can be best represented using algorithmic descriptions. All model exchange formats known to the authors fall short of adequately expressing this dynamic behavior.

Therefore we must come to the conclusion that both interoperability standards discussed in this paper are not redundant. Rather, they complement each other as they address different aspects of interoperability in the manufacturing domain.

5. Conclusions

This paper has presented SISO standards SISO-STD-006-2010 and SISO-STD-008-2010 which both address interoperability issues in the manufacturing simulation domain.

SISO-STD-006-2010 with its interoperability reference models provides primarily help when interoperability between multiple CSPs is an issue and coupling of models in multiple CSPs by means of distributed simulation seems an appropriate solution.

SISO-STD-008-2010 with its CMSD information model has its specific strengths when interoperability between a CSP and other manufacturing IT systems is required. It provides a standardized data exchange format which is capable of storing and transporting data required for automatic model generation and initialization. CMSD may also, at least to some extent, be useful as a model exchange format for converting basic model information between the native formats of different CSPs. The requirement for that is the implementation of model import and export filters within the respective CSPs, a task which currently is not actively supported by CSP vendors.

Acknowledgments

The authors would like to thank the members of both PDGs involved in the production of the discussed SISO standards. We would also like to point out that significant parts of the discussions provided in this paper are based on previous publications, which have been cited to the best of our knowledge.

References

[1] Boer, C. A. Distributed Simulation in Industry. Rotterdam: Erasmus Research Institute of Management, 2005.

[2] Lendermann, P., M. Heinicke, L. McGinnis, C. McLean, S. Strassburger, S. Taylor. Distributed Simulation in Industry - a Real-world Necessity or Ivory Tower Fancy? In: Proceedings of the 2007 Winter Simulation Conference, eds. S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, pp. 1053-1062. December 9-12, 2007. Washington, D.C., USA.

[3] Noseworthy, J.R. The Test and Training Enabling Architecture (TENA) Supporting the Decentralized Development of Distributed Applications and LVC Simulations. In: Proceedings of the 12th IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications, 2008 (DS-RT 2008), pp. 259 – 268.

[4] Banks, J., J. Carson, B. Nelson, D. Nicol. Discrete Event Simulation (Fourth Edition). Prentice Hall, 2005

[5] Sargent, B., R. Nance, C. Overstreet, S. Robinson, J. Talbot. The Simulation Project Life-Cycle: Models and Realities. In: Proceedings of the 2006 Winter Simulation Conference, eds. L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, pp. 863-871.

[6] Taylor, S., S. Strassburger, S. Turner. The SISO CSPI PDG Standard for COTS Simulation Package Interoperability Reference Models. In: Proceedings of the 2008 European Simulation Interoperability Workshop. June 2008. Edinburgh, Scotland.

[7] Taylor, S., S. Turner, and M. Low. 2004. A Proposal for an Entity Transfer Specification Standard for COTS Simulation Package Interoperation. In Proceedings of the 2004 European Simulation Interoperability Workshop. June 28 - July 1, 2004. Edinburgh, Scotland.

[8] Wang, X., S.J. Turner, S.J.E. Taylor. 2006. COTS Simulation Package (CSP) Interoperability – A Solution to Synchronous Entity Passing. In: Proceedings of the 20th Workshop on Principles of Advanced and Distributed Simulation (PADS'06), pp. 201 - 210.

[9] Straßburger, S. The Road to COTS-Interoperability: From Generic HLA-Interfaces Towards Plug-And-Play Capabilities. In: Proceedings of the 2006 Winter Simulation Conference, eds. L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, pp. 1111-1118. December 3-6, 2006. Monterey, USA.

[10] Pedrielli, G., P. Scavardone, T. Tolio, M. Sacco, W. Terkaj. Simulation of Complex Manufacturing Systems via HLA-Based Infrastructure. In: Proceedings of the 2011 ACM/IEEE/SCS Workshop on Principles of Advanced and Distributed Simulation (PADS 2011), pp. 78-86

[11] Strassburger, S., T. Schulze, M. Lemessi. Applying CSPI Reference Models for Factory Planning. In: Proceedings of the 2007 Winter Simulation Conference, eds. S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, pp. 603-609. December 9-12, 2007. Washington, D.C., USA.

[12] Mustafee, N., SJE Taylor, K. Katsaliaki, S. Brailsford. Facilitating the analysis of a UK national blood service supply chain using distributed simulation, Simulation 85 (2) : 113-128

[13] SISO. 2010. "Standard for Commercial-off-the-shelf Simulation Package Interoperability Reference Models". COTS Simulation Package Interoperability Product Development Group, Simulation Interoperability Standards Organization. Accessed January 2, 2012. http://www.sisostds.org/DigitalLibrary.aspx?Command=Core_Download&EntryId=30829.

[14] SISO. 2010. "Standard for: Core Manufacturing Simulation Data – UML Model." Core Manufacturing Simulation Data Product Development Group, Simulation Interoperability Standards Organization. Accessed January 2, 2012. http://www.sisostds.org/DigitalLibrary.aspx?Command=Core_Download&EntryId=31457.

[15] Johansson, M., B. Johansson, S. Leong, F. Riddick, Y. T. Lee. A Real World Pilot Implementation of the Core Manufacturing Simulation Information Model. In: Proceedings of the Simulation Interoperability Standards Organization Spring 2008 SIW Workshop.

[16] Leong, S., Y. T. Lee, F. Riddick. A Core Manufacturing Simulation Data Information Model for Manufacturing Applications. In: Proceedings of the 2006 Fall Simulation Interoperability Workshop. Simulation Interoperability Standards Organization. Orlando, Florida.

[17] Bergmann, S., S. Stelzer, S. Strassburger. Initialization of Simulation Models using CMSD. In: Proceedings of the 2011 Winter Simulation Conference, eds. S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, pp. 2228-2239.

[18] Fournier, J. Model Building With Core Manufacturing Simulation Data Translators. In: Proceedings of the 2011 Winter Simulation Conference, eds. S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, pp. 2219-2227.

[19] Taylor, S., N. Mustafee, S. Turner, K. Pan, S. Strassburger. Commercial-Off-The-Shelf Simulation Package Interoperability: Issues and Futures. In: Proceedings of the 2009 Winter Simulation Conference, eds. M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, pp. 203-215. December 13-16, 2009. Austin, USA.

[20] Bergmann, S., A. Fiedler, S. Straßburger. Generierung und Integration von Simulationsmodellen unter Verwendung des Core Manufacturing Simulation Data (CMSD) Information Model. In: Proceedings der 14. ASIM-Fachtagung Simulation in Produktion und Logistik, eds. G. Zülch, P. Stock. Karlsruhe: KIT Scientific Publishing, 2010, pp. 461-468.

Author Biographies

STEFFEN STRASSBUGER is a professor at the Ilmenau University of Technology in the School of Economic Sciences. In previous positions he was working as head of the "Virtual Development" department at the Fraunhofer Institute in Magdeburg, Germany and as a researcher at the DaimlerChrysler Research Center in Ulm, Germany. His research interests include simulation and distributed simulation as well as general interoperability topics within the digital factory context. He was the Vice Chair of SISO's COTS Simulation Package Interoperability Product Development Group. His email address is steffen.strassburger@tu-ilmenau.de.

SIMON J E TAYLOR is the Founder and Chair of the COTS Simulation Package Interoperability Product Development Group (CSPI-PDG) under the Simulation Interoperability Standards Organization. He is the co-founding Editor-in-Chief of the UK Operational Research Society's (ORS) *Journal of Simulation* and the Simulation Workshop series. He was Chair of ACM's SIGSIM (2005-2008). He is a Reader in the School of Information Systems, Computing and Mathematics at Brunel and leads the ICT Innovation Group. He has published over 100 articles in modeling and simulation. His recent work has focused on the development of standards for distributed simulation in industry. His email address is simon.taylor@brunel.ac.uk.