
Proceedings Paper:
Clement, SJ orcid.org/0000-0003-2918-5881, McKee, DW orcid.org/0000-0002-9047-7990, Romano, R orcid.org/0000-0002-2132-4077 et al. (3 more authors) (2017) The Internet of Simulation: Enabling Agile Model Based Systems Engineering for Cyber-Physical Systems. In: IEEE 12th System of Systems Engineering Conference (SoSE 2017). SoSE 2017, 18-21 Jun 2017, Waikoloa, Hawaii, USA. IEEE . ISBN 978-1-5090-5945-4

https://doi.org/10.1109/SYSOSE.2017.7994948

The Internet of Simulation: Enabling Agile Model Based Systems Engineering for Cyber-Physical Systems

S. J. Clement*, D. W. McKee*, Richard Romano†, Jie Xu*
*School of Computing, †Institute for Transport Studies
University of Leeds
Leeds, UK
{S.J.Clement, D.W.McKee, R.Romano, J.Xu}@leeds.ac.uk

J.M. Lopez
SimWare Solutions
Madrid
Spain
JMLopez@simware.es

D. Battersby
Jaguar Land Rover
Gaydon
UK
dbatters@jaguarlandrover.com

***Abstract*—The expansion of the Internet of Things (IoT) has resulted in a complex cyber-physical system of systems that is continually evolving. With ever more complex systems being developed and changed there has been an increasing reliance on simulation as a vital part of the design process. There is also a growing need for simulation integration and co-simulation in order to analyse the complex interactions between system components. To this end, we propose that the Internet of Simulation (IoS), as an extension of IoT, can be used to meet these needs. The IoS allows for multiple heterogeneous simulations to be integrated together for co-simulation. It's effect on the engineer process is to facilitate agile practices without sacrificing rigour. An Industry 4.0 example case study is provided showing how IoS could be utilised.**

***Index Terms*—IoT, IoE, IoS, Simulation, Cloud, SOA, Real-Time, Services, Workflow, Modelling, M&S, SIMaaS, WFaaS**

I. Introduction

The Internet of Things (IoT) is becoming one of the central paradigms of 21^{st} century IT with evermore mobile and interconnected systems [1]. As a general paradigm IoT endeavours to facilitate the connection of Everything (IoE) and Anything (IoA) via the Internet as a set of hugely complex System of Systems (SoS) [2]. Specifically IoT focusses on the integration of *smart* cyber-physical devices where each element can be regarded as a relatively small component or system. As the complexity of these systems increases, engineers are relying more heavily on simulation both for design and Verification and Validation (V&V). There is an identified need for methods and standards to allow rapid prototyping of simulation compositions and simulation interaction with real-world systems. To address these needs we previously proposed the concept of Internet of Simulation (IoS) [3].

One of the most integral elements of the design and development of complex SoS's is simulation throughout the engineering life-cycle including both the integration and validation phases. In order to upgrade a given individual component, for example, the required validation against the existing systems may only be possible through the use of simulation. Either by simulating: the entire SoS; just the component of interest; or the existing system. This depends on the type of system that is being developed or upgraded as well as the stage of the engineering lifecycle the system or component at.

The changes to the engineering lifecycle enabled by IoS bring key benefits to IoT and industry in general. Particularly in manufacturing industries from automotive, aerospace, through to even furniture design, simulation forms an integral part of the design and verification process. At any stage of the design process there can be multiple individual component simulations; complex system simulations using co-simulation; and SoS integration testing [4].

The complexity of emerging cyber-physical systems and SoSs requires increasingly more complex simulations and co-simulations, often produced by multiple stakeholders to fully analyse designs. In order to facilitate this at each and across all lifecycle stages we propose in this paper the Internet of Simulation (IoS) as an engineering tools and suggest how this might effect the engineering process.

The remainder of this paper will discuss the background of simulation integration, current technologies, and standards. In Section III IoS will be presented both architecturally and with respect to its impact on the engineering lifecycle. In Section IV an Industry 4.0 case study is presented before conclusions and future research are discussed in Section V.

II. Background of Simulation Integration

The integration of heterogeneous simulations within a common simulation domain using network protocols has been heavily adopted within the military domain since the late 1980's. This type of simulation has become known as Live-Virtual-Constructive (LVC) simulation.

Several organisations have worked on the standardization of simulation protocols and architectures; including SISO on connectivity and interoperability, and the IEEE with several standards including: DIS (IEEE1278), HLA (IEEE 1516.2000), HLA Evolved (IEEE 1516.2010), MSDL, CBML, CIGI, and DSEEP/DMAO (IEEE 1730.2010 / IEEE 1730.1-2013). These standards have however been mostly developed in parallel with many connectivity and interoperability issues. In fact, only

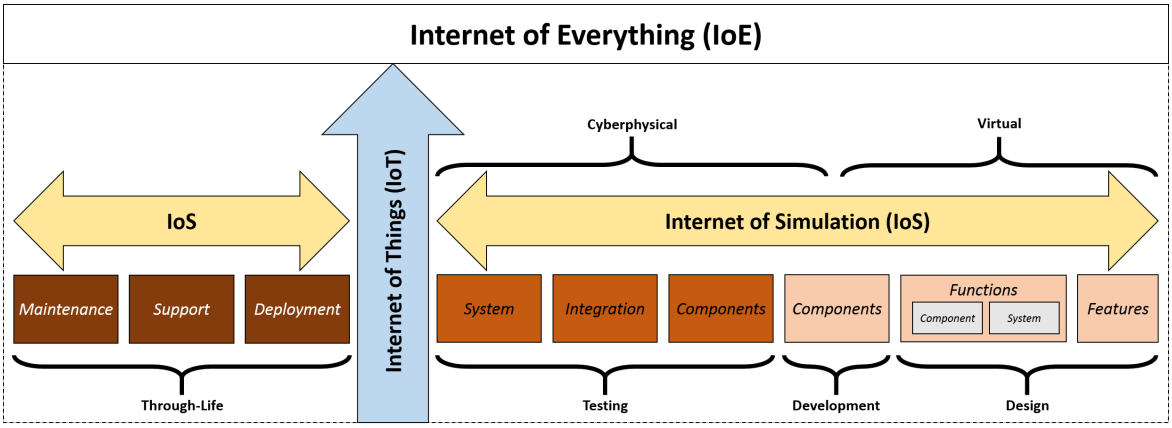

Fig. 1. IoS aims to support all simulations through the engineering life cycle.

low levels of interoperability (level 2 at most) as defined in simulation theory [5], can be achieved now.

Furthermore, SISO and IEEE standards adopted in the military and aeronautic domains have not been universally adopted across other industrial domains such as automotive manufacturing. Instead, alternative standards with the same overall objectives have been designed to support integration and co-simulation, e.g. FMI [6] and pseudo-standards like Simulink S-Functions. Additionally other simulator developers make use of a range of standards including: DDS, CORBA, and web services (WS). A further discussion about the main differences between the main standards in use now to interoperate simulators in the military and aerospace domains is provided by Martinez et al. [7]

Even with these standards, the effective integration and interoperability of heterogeneous simulators remains challenging, requiring complex, expensive, unmaintainable and unscalable ad-hoc solutions. This is primarily due to the lack of a SoSs, and specifically a distributed systems, approach to simulator development. In the computing domain there are a few such simulators, such as SEED [8] and PlanetLab [9], however these are only loosely standards compliant with WS, and across other domains there remains a need for the development of Simulation as a Service (SIMaaS) tools and technologies [10].

One approach under current development is the Layered Simulation Architecture (LSA) approach, a nominated standard at SISO [11]. LSA was developed to address protocol interoperability between standards by using a common data domain to exchange data in the same way Service Oriented Architectures (SOAs) share workflow variables. An implementation of LSA is provided by the Simware simulation platform [12].

III. The Internet of Simulation

The Internet of Simulation (IoS) [3] describes an emerging extension of Internet of Things (IoT) into the domain of simulation (see figure 1). This trend is a convergence of an identified need for increased co-simulation and the increasing application of Internet technologies in simulation. Gubbi et al. [13] define IoT as the use of digital technologies to facilitate the interconnection of components, devices and services at a large-scale across a network. IoS describes a specialisation of this to utilise digital technologies and standards to facilitate the large-scale integration of virtual, simulated systems. This integration can cross into the domain of IoT where simulated systems interact with real world cyber-physical systems. Therefore, IoS describes a number of technologies and methodologies to develop cyber-physical systems where elements are substituted for virtual representations, an internet of *virtual* things. Additionally, with increased simulation capacity and the potential for real-time communication IoS could be used for decision support in traditional IoT applications. In particular, IoS represents three distinct possible usecases for simulation in cyber-physical systems:

Sim $\rightarrow$ **Sim** Co-simulation for virtual engineering, virtual components being used to construct a virtual system for design analysis.

Sim $\rightarrow$ **IoT** Simulated data being fed into IoT systems to verify their operation.

IoT $\leftrightarrow$ **Sim** Data from IoT systems being fed into simulations to predict possible outcomes and provide information for actuation decisions

IoS as described by McKee et al. [3] can be broadly separated into three layers shown in figure 2: integration (WFaaS), execution (SIMaaS) and infrastructure (Cloud). IoS is envisaged to utilise existing cloud technologies and SOAs [14] that already enable large-scale cyber-physical systems. The infrastructure layer of IoS encapsulates the traditional cloud architecture layers of Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and Software as a Service (SaaS) allowing for simulations to be scaled up and executed as needed, taking advantage of the elasticity cloud platforms provide. Simulation as a Service (SIMaaS) provides an execution layer for individual simulations. It allows for the deployment of many different types and fidelities of simulation to be executed on an *as-a-Service* basis. These simulations can be custom simulations designed to run on specific hardware or generic simulations

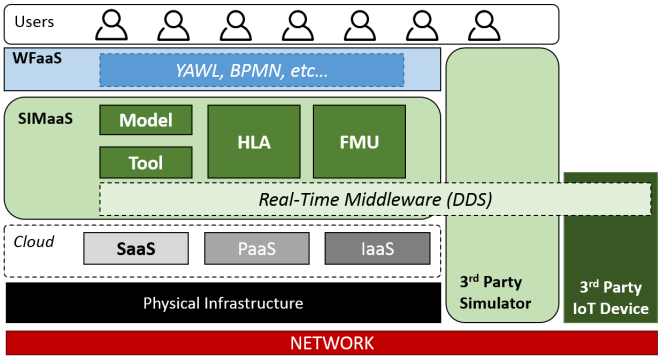

Fig. 2. IoS layered architecture showing some available technologies and standards and how they fit into IoS

created in 3rd-Party tools and hosted for execution in their respective environments. Additionally, 3rd party simulators and devices are integrated into workflows and communicate with IoS via the real-time middleware.

The primary benefit of IoS is large-scale integration and co-simulation of multiple simulations. Simulations can be integrated together and executed as a single virtual system in the workflow layer: Workflow as a Service (WFaaS). Here a user can describe a system simulation using virtual components. The behaviour of the system is captured by the interactions and relationships between components and also with the wider virtual environment. Ultimately IoS must remove the tight coupling of simulations and their respective execution environments [15] for large scale adoption. Currently, most approaches to this problem take a data-centric view, representing the simulations as a black boxes with a collection of inputs and outputs [7], [10], [16].

If wide scale support for simulation integration and co-simulation is achieved along with integration across different simulation domains and fidelities then IoS can easily enable the transition between the different layers of abstraction that are present in system design. At the architectural level there exist the modelling languages such as UML and SysML which statically define the architecture of systems. The introduction of fUML has allowed for the behaviour of systems to be defined using an action language. This allows for the transition from a purely static viewpoint to a dynamic one. In the engineering lifecycle many simulations at differing fidelity are employed ranging from 1D behaviour models to full 3D complex analysis. The primary concerns for the introduction of IoS are the lack of a dedicated standard that allows full simulation integration with unlimited co-simulation and a notation allowing for a formal mapping between architectural and behavioural models in UML/fUML to multi-disciplinary, high fidelity simulations. The definition of such a notation is beyond the scope of this paper but we will examine the available standards for their suitability for IoS.

A. Standards for simulation in IoS

IoS requires technologies and standards from three areas: simulation (integration and co-simulation), communication (middleware and services) and specification languages. A number of technologies and standards already exist that make some progress towards an implementation of IoS:

*1) **FMI**:* Functional Mockup Interface (FMI) is a standard which aims for tool independence enabling model exchange and co-simulation [6]. A component of a larger simulation can be shared and executed by sharing a Functional Mockup Unit (FMU). FMI operates on a single machine with a tool acting as a master, executing slave FMUs. Despite the growing adoption of FMI there are compatibility issues between supporting tools, resulting in execution errors or inconsistent simulation results and the support for multiple FMU's is often untested [17]. Co-simulation also occurs in a single memory environment via function calls so multiple FMUs running in parallel can slow down the simulation. A distributed simulation standard which is very similar to FMI is Functional Digital Mock-Up (FDMU). It provides distributed co-simulation using Web Services based on a SOA architecture and allows a loose coupling of individual simulation components. A comparison of the two standards is provided by Enge-Rosenblatt et al. [18]

*2) **HLA**:* High Level Architecture (HLA) [19] originated as a military standard for LVC, specifies an architecture for simulation reuse and integration. Using HLA simulations are constructed by composing individual distributed simulations into federations sharing a common object model. Simulations interact via a Runtime Infrastructure (RTI), which must be consistent across the federations, but is not defined as part of the standard. Although HLA facilitates distributed simulation as required by IoS, and an RTI coul dbe defined for IoS. It does not lend itself to large-scale, elastic, cloud computing infrastructure that will be required by IoS and IoT.

*3) **Data Distribution Service (DDS)**:* The Object Management Group (OMG) Data Distribution Service (DDS) [20] is a real-time middleware designed around the publisher-subscriber model in a global data space. DDS facilitates data distribution in large scale, distributed systems providing 21 Quality of Service (QoS) parameters. Additionally, DDS has previously been applied to distributed co-simulation and has an overlap with HLA [7]. The one drawback of DDS is the limited support for cloud applications [21].

*4) **Web Services (WS-*)**:* Large scale co-simulation and integration like that proposed by IoS will require large amounts of distributed computing power to operate. Additionally, the need for users to dynamically start and stop multiple simulation as needed requires this power to be elastic. These two elements are already supplied by existing cloud technologies [22].

*5) **Workflows**:* A method for specifying the configuration of simulations in IoS is required. Workflows [23] are already a common method of defining processes and can produce an orchestration of web services. Workflow languages such a Business Process Model and Notation (BPMN) [24], Business Process Execution Language (BPEL) [25] and Yet Another

Workflow Language (YAWL) [26] are mature and able to represent a majority of the workflow patterns identified by Russell et al. [27], [28]. These also have the benefit of being able to be represented graphically [29], greatly reducing the barrier to entry for engineers already familiar with other graphical specification languages.

However, current workflow languages focus on processes, with sequential execution along a predefined path. In contrast, co-simulation requires each simulation service to execute in parallel with the defined dependencies between simulations taking precedence. This is far more similar to a declarative data-flow language than an imperative process workflow.

Declarative data-flow languages are already common in the simulation community and used in tools such as Simulink, LabView or Smalltalk [30]. These languages can treat components as black-boxes and encode the relationships between them as a directional graph of variable dependencies. However, these existing execution tools tightly couple simulations together and are focussed on local execution rather than remote. Additionally, most workflow langauges create an orchestration with a central execution engine calling the required services. An alternative to service orchestration is service choreography [31] which reduces the need for centralised control and therefore removes the communication overhead.

It is apparent that no single solution to the need of IoS for a specification language currently exists. In particular, a critical requirement for this IoS language in systems engineering would be a formal transformation or validation between system specification languages such as UML or SysML and simulation integrations. If system simulations are constructed from a several complex interactions of black-box simulations providing different viewpoints on the system behaviour then this mapping is crucial to ensure that the simulation integration accurately reflects the system design.

B. Systems Engineering with IoS

With an ability to integrate simulations together and execute large-scale co-simulations the engineering process can be dramatically transformed. First and foremost, with wide-spread support for integration the prospect of creating high-fidelity full system simulations becomes feasible. This leads to a much easier assessment of system level behaviour and, depending on the accuracy of the simulations, could lead to emergent system behaviour being identified earlier in the design process.

Additionally, by constructing system simulations using an available library of online component simulations IoS could facilitate the rapid prototyping of many different candidate system designs if the required component simulations already exist. These individual component simulations could even be made public by 3rd-party vendors without releasing internal source code. These product simulations could potentially be made available to be executed by an existing or prospective customer. Enabling the engineer to trial various potential solutions rapidly before fully engaging any particular supplier. Additionally, the simulation behaviour could form the basis of specifications for supplied components.

Furthermore, WFaaS allows SoSs simulation workflows to be constructed and executed as services, and removes the distinction between workflows and individual simulations. This leads to a hierarchical system model representing both system-level behaviour and more granular component behaviour and can allow the designer to traverse the various abstraction layers of the system. Further, within individual workflows there may be different layers of simulation abstraction, iteratively increasing the simulation fidelity in an Agile development manner with continuously increasing complex simulations. Successful integration of these workflows can therefore feedback to the earlier stages and inform the more abstract behavioural models and simulations. Traversing these abstraction layers allow a holistic view of system to be maintained while designing individual components.

If the above is achieved in IoS then this allows for a radical change to the engineering process where Agile methodologies from the software community can be adopted into Model Based Systems Engineering. Test driven development can be utilised where continual testing can be performed on the virtual system as it is developed and rapid changes to requirements can be met by altering models and re-testing.

IV. Industry 4.0 IoT Case Study

As described earlier in Section II and depicted in Figure 1 Industry 4.0 is one of the key domains of both IoT and IoS. Previously we presented an automotive case study integrating simulations of an engine, transmission, transmission control unit (TCU) as hardware-in-the-loop (HIL), and vehicle dynamics in-the-loop with a human driver [16]. However IoS allows further simulation integration across the manufacturing process bringing together the industrial simulations, including vehicular components, with elements of cities, particularly smart cities [32].

In Figure 3 the automotive and manufacturing case studies from [16] & [3] are extended to demonstrate what Industry 4.0 should look like in practice from the perspective of designing and manufacturing autonomous vehicles (AVs). As AVs are introduced there will be a hybrid situation with non-AVs, traffic control systems that may or may not be smart, and pedestrians. Therefore to facilitate safe introduction as well as V&V, simulations must also be integrated.

During AV design there will be many simulations and co-simulations performed of the vehicle itself. IoS facilitates an immediate benefit by providing a net-centric platform in which to connect and interoperate the digital models and simulations across the whole virtual supply chain, from Original Equipment Manufacturers (OEMs) to the automotive companies. In Figure 3 the Engine and Transmission Control Units (ECU & TCU) are integrated via the Cloud and web services, with the former being hosted as a HIL system and the latter as a virtual model representing the control system. These are integrated with other simulations of the engine and transmission system to provide a complete 1D Power Train simulation using DDS. The 1D Power Train along with a body-in-white (B-in-W) 3D simulation are fed into the Leeds Driving Simulator [33]

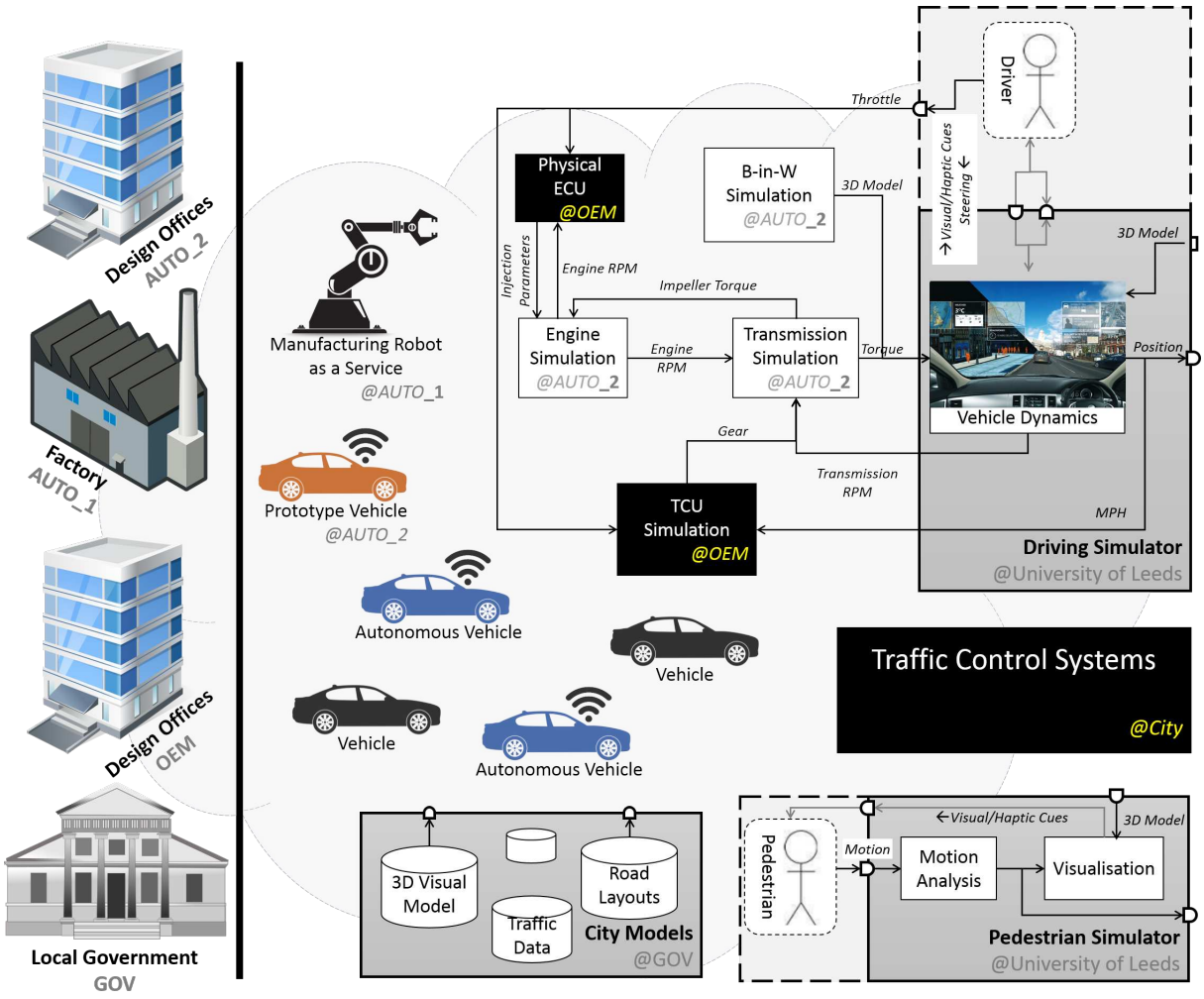

Fig. 3. Industry 4.0 IoT Case Study for autonomous vehicle manufacturing and smart city modelling, enhanced by IoS through Cloud integration across design offices, factories, local government, and academic institutions as part of the VirtuoCITY project [32]

which along with a human driver feeds back into the OEM control units. Changes in the body-in-white models are used to reconfigure the manufacturing robots on the assembly line.

In order to use the driving simulator to evaluate AVs within the city context, 3D visual models of the city, road layouts, and historical traffic data are fed in from local government repositories provided as web services. This data is also fed into a Pedestrian Simulator allowing a human to interact in the same virtual world as the AV that is being designed.

Subsequently a prototype or production AV deployed within the city should also be integrated to provide data services such as accurate traffic data form the local government [34] it can also feedback data the design teams at the relevant manufacturers for the purposes of a *digital twin*. That data is also of use to academic communities and others, such as city planners, for analysis and further simulation. In addition to the design and integration tasks in IoS, simulations can be used for prediction and decision support in live systems [35].

This is a typical case in which many heterogeneous simulations and cyberphysical systems needs to work integrated in a common virtual design-space located in the cloud. Typical solution, used for example in LVC military simulation, would be designed with a gateway based architecture, using gateways and ad-hoc adaptors to translate between the different data-models and protocols used by each component in the integrated solution. This solution is full of restrictions and limitations as it is explained at [7], [10]. As an alternative, an IoS compliant solution can be integrated using an open and modular architecture as the SISO nominated standard LSA, which provides a common simulation platform where all the simulations and cyberphysical systems will exchange data and perform integrated in a common virtual space. All components will be working as black-boxes, interoperating with the other federates by exchanging live and simulation data through the common simulation platform.

V. Conclusion and Further Work

The Internet of Things (IoT) is becoming an increasingly popular and integral aspect of industrial and national infrastructure, facilitating the integration of devices and systems

into ever larger and more complex cyber-physical System of Systemss (SoSs). As manufacturing moves towards Industry 4.0 there is a need to extend IoT to support engineering processes including modelling & simulation for the purposes of design, verification, and validation.

In this paper we extend the concept of the Internet of Simulation (IoS) to support engineering methods for integration of simulations and prototypes across product lifecycle stages. IoS is an extension of IoT bringing simulation in-the-loop with prototype and production systems connected as *things* via the Cloud in order to facilitate greater analysis of those systems through data collection and simulation. The use of IoS in manufacturing results in improved agility of the engineering process with continuous system and component verification.

In Section IV a case study is presented showing the integration of models and simulations across several organisations including an automotive manufacturer, OEM, local government, and academic facilities for the purposes of prototyping and evaluating autonomous vehicles. However, in order to realise IoS at a significant scale there needs to be a concerted effort to normalise and facilitate integration between the various standards that currently exist for simulation integration, including DDS, HLA, FMI, and WS. We propose to leverage the results obtained by the SISO study group for LSA, to work on the development of an IoS standard that can enable the model based system engineering of IoT systems. LSA can provide the foundation to build the SIMaaS and WFaaS layers of a future IoS standard.

ACKNOWLEDGMENT

This work was supported by Jaguar Land Rover, UK-EPSRC grant EP/K014226/1 as part of the Programme for Simulation Innovation.

REFERENCES

[1] V. Woods and R. van der Meulen, "Iot adoption is driving the use of platform as a service," *Gartner report: Predicts 2016: PaaS Innovation Continues Unabated*, 2016.

[2] I. Bojanova, G. Hurlburt, and J. Voas, "Imagineering an internet of anything," *Computer*, vol. 47, no. 6, pp. 72–77, jun 2014.

[3] D. Mckee, S. Clement, X. Ouyang *et al.*, "The internet of simulation, a specialisation of the internet of things with simulation and workflow as a service (sim/wfaas)," in *11th IEEE International Symposium on Service-Oriented System Engineering (SOSE 2017)*. IEEE, January 2017.

[4] D. McKee, D. Webster, P. Townend *et al.*, "Towards a virtual integration design and analysis enviroment for automotive engineering," in *2014 IEEE 17th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing*. IEEE, 2014.

[5] A. Tolk and J. A. Muguira, "The levels of conceptual interoperability model," in *Proceedings of the 2003 fall simulation interoperability workshop*, vol. 7. Citeseer, 2003, pp. 1–11.

[6] T. Blochwitz, M. Otter, J. Akesson *et al.*, "Functional mockup interface 2.0: The standard for tool independent exchange of simulation models," in *Proc. 9th International MODELICA Conference*, 2012, pp. 173–184.

[7] J.-R. Martinez, J.-M. Lopez, D. Gregory *et al.*, "A comparison of simulation and operational architectures," in *SISO Fall SIW*, 2012.

[8] P. Garraghan, D. McKee, X. Ouyang *et al.*, "SEED: A scalable approach for cyber-physical system simulation," *IEEE Transactions on Services Computing*, vol. 9, no. 2, pp. 199–212, mar 2016.

[9] B. Chun, D. Culler, T. Roscoe *et al.*, "PlanetLab," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 3, p. 3, jul 2003.

[10] J. R. Martinez and J. M. Lopez, "Future of lvc simulation: Evolving towards the msaas concept," in *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*, 2014.

[11] D. Gregory, J.-M. Lopez, J.-R. Martinez *et al.*, "A new approach for converging lvc simulation architectures," in *SISO Fall SIW*, 2012.

[12] "Simware." [Online]. Available: http://www.simware.es

[13] J. Gubbi, R. Buyya, S. Marusic *et al.*, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645 – 1660, 2013.

[14] M. P. Papazoglou, P. Traverso, S. Dustdar *et al.*, "Service-Oriented Computing: State of the Art and Research Challenges," *Computer*, vol. 40, no. 11, pp. 38–45, 2007.

[15] C. E. Dickerson, S. Ji, S. J. Clement *et al.*, "A demonstration of a service oriented virtual environment for complex system analysis," *International Journal of Complex Systems - Computing, Sensing and Control*, vol. 3, no. 1, pp. 49–65, 2015.

[16] D. W. McKee, D. Webster, J. Xu *et al.*, "DIVIDER: Modelling and Evaluating Real-Time Service-Oriented Cyberphysical Co-Simulations," in *2015 IEEE 18th International Symposium on Real-Time Distributed Computing*. IEEE, 2015, pp. 272–275.

[17] C. Bertsch, E. Ahle, and U. Schulmeister, "The functional mockup interface - seen from an industrial perspective," in *Proc. 10th International Modelica Conference*. Linkoping University Electronic Press, 2014.

[18] O. Enge-Rosenblatt, C. Clauß, A. Schneider *et al.*, "Functional digital mock-up and the functional mock-up interface-two complementary approaches for a comprehensive investigation of heterogeneous systems," in *Proc. 8th International Modelica Conference*, no. 063. Linköping University Electronic Press, 2011, pp. 748–755.

[19] F. Kuhl, R. Weatherly, and J. Dahmann, *Creating computer simulation systems: an introduction to the high level architecture*. Prentice Hall PTR, 1999.

[20] G. Pardo-castellote, "OMG Data-Distribution Service (DDS): Architectural Overview," Real-Time Innovations, Inc. (RTI), Tech. Rep., 2005.

[21] K. An, S. Pradhan, F. Caglar *et al.*, "A publish/subscribe middleware for dependable and real-time resource monitoring in the cloud," in *Proceedings of the Workshop on Secure and Dependable Middleware for Cloud Monitoring and Management*. ACM, 2012, p. 3.

[22] R. Yang and J. Xu, "Computing at massive scale: Scalability and dependability challenges," in *2016 IEEE Symposium on Service-Oriented System Engineering (SOSE)*. IEEE, 2016.

[23] D. Hollingsworth and U. Hampshire, "Workflow management coalition: The workflow reference model," *Document Number TC00-1003*, vol. 19, 1995.

[24] S. A. White, "Introduction to bpmn," *IBM Cooperation*, 2004.

[25] D. Jordan, J. Evdemon, A. Alves *et al.*, "Web services business process execution language version 2.0," *OASIS standard*, vol. 11, no. 120, p. 5, 2007.

[26] W. van der Aalst and A. ter Hofstede, "Yawl: yet another workflow language," *Information Systems*, vol. 30, no. 4, pp. 245 – 275, 2005.

[27] N. Russell, A. H. M. ter Hofstede, D. Edmond *et al.*, *Workflow Data Patterns: Identification, Representation and Tool Support*. Springer Berlin Heidelberg, 2005, pp. 353–368.

[28] N. Russell, A. H. Ter Hofstede, and N. Mulyar, "Workflow controlflow patterns: A revised view," 2006.

[29] D. Schumm, D. Karastoyanova, F. Leymann *et al.*, "On visualizing and modelling bpel with bpmn," in *Grid and Pervasive Computing Conference, 2009. GPC'09. Workshops at the*. IEEE, 2009, pp. 80–87.

[30] A. Buss and L. Jackson, "Distributed simulation modeling: a comparison of hla, corba, and rmi," in *Proceedings of the 30th conference on Winter simulation*. IEEE Computer Society Press, 1998, pp. 819–826.

[31] A. Barker, C. D. Walton, and D. Robertson, "Choreographing web services," *IEEE Transactions on Services Computing*, vol. 2, no. 2, pp. 152–166, April 2009.

[32] "Virtuocity." [Online]. Available: http://www.its.leeds.ac.uk/research/featured-projects/virtuocity/

[33] "Leeds driving simulator." [Online]. Available: http://www.uolds.leeds.ac.uk/

[34] S. Clement, D. Mckee, and J. Xu, "Service-oriented reference architecture for smart cities," February 2017.

[35] D. Mckee, S. Clement, J. Almutairi *et al.*, "Massive-scale automation in cyber-physical systems: Vision & challenges," February 2017.