Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A.A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

SYSTEMS ENGINEERING AND SIMULATION: CONVERGING TOWARD NOBLE CAUSES

Bernard P. Zeigler

ACIMS, University of Arizona
Tucson, AZ, US
and
RTSync Corp,
Chandler, AZ 85286, US

Joseph W. Marvin
John J. Cadigan

Prime Solutions Group, Inc (PSG)
1300 S. Litchfield Road
Goodyear, AZ 95338, US

ABSTRACT

This paper highlights the accomplishments and shared vision between the International Council on Systems Engineering (INCOSE) and the Modeling and Simulation community (represented by the Society for Modeling and Simulation, International (SCS), and Simulation Interoperability Standards Organization (SISO, among others). We describe convergence between the model-based systems engineering initiative of the INCOSE community and the model-based simulation developments of the SCS community. The goal is not only to highlight the outstanding accomplishments of our time, but also to emphasize the parallels and relationships. The paper is intended to enhance communications and facilitate the outreach already in motion. Modeling and Simulation (M&S) represents a core capability and need for addressing today's complex and grand challenges. We suggest a collaboration of INCOSE and SCS, as leaders in the systems and M&S communities, to solve these challenges complicated by multi-dimensional, hierarchical, and uncertain Big Data and propelled by exascale computational platforms.

1 INTRODUCTION

Military and defense, smart energy grids, and additive manufacturing are examples of systems that have received considerable attention across INCOSE and SCS (we use the designation INCOSE and SCS to designate the systems engineering and modeling and simulation (M&S) communities, resp.) Complexity and emergence have become the norm for these domains. In the meantime, decision makers want more capability, resilience, and predictability at the system and System of Systems levels. We introduce the notion of a noble cause to the community where engineering practices are scaled to "next-level" triumphs

- These Systems of Systems (SoS) have software components that must increasingly operate on large, time-varying, heterogeneous data.
- Big Data (defined by Volume, Velocity, Variety, and Veracity attributes) determine 1) how these systems perform across an indeterminate number of scenarios, 2) how their dynamics are controlled in real-time, and 3) their quality of decisions.
- Adequately understanding, engineering, and operating such systems present challenges that are multi-dimensional, hierarchical, and uncertain requiring advanced M&S capabilities.

In this paper, we discuss how engineers must increasingly apply advanced M&S techniques and latest Information Technology (IT) supported by systems engineering research to address such challenges. We give examples of how agile systems engineering combined with system-theory based M&S artifacts can be applied to support discovery of system behavior patterns that can help develop solutions to today's systems engineering problems. We discuss how advancements in systems engineering and M&S have evolved, seemingly independently, but great minds are converging to recognize patterns, agility and formalisms as the overall roadmap for better engineered solutions.

Complex and emerging systems such as ballistic missile defense, smart energy grids, and additive manufacturing have become the norm for today's systems engineering challenges. Moreover, these systems continue to increase in complexity as more capability and new features are asked of these systems by society and decision makers. At the global enterprise level, INCOSE has recognized these needs. As a result, progress has been made on many fronts, but there is still much to do. The authors are practitioners from various backgrounds and association with technical societies such as INCOSE and SCS. While collaborating on opportunities, we have seen potential for integrating significant work across these communities. Complex military, space, energy and manufacturing systems are constantly met with changing technology. Figure 1 is one perspective of the many components that influence the environment we find ourselves in today.

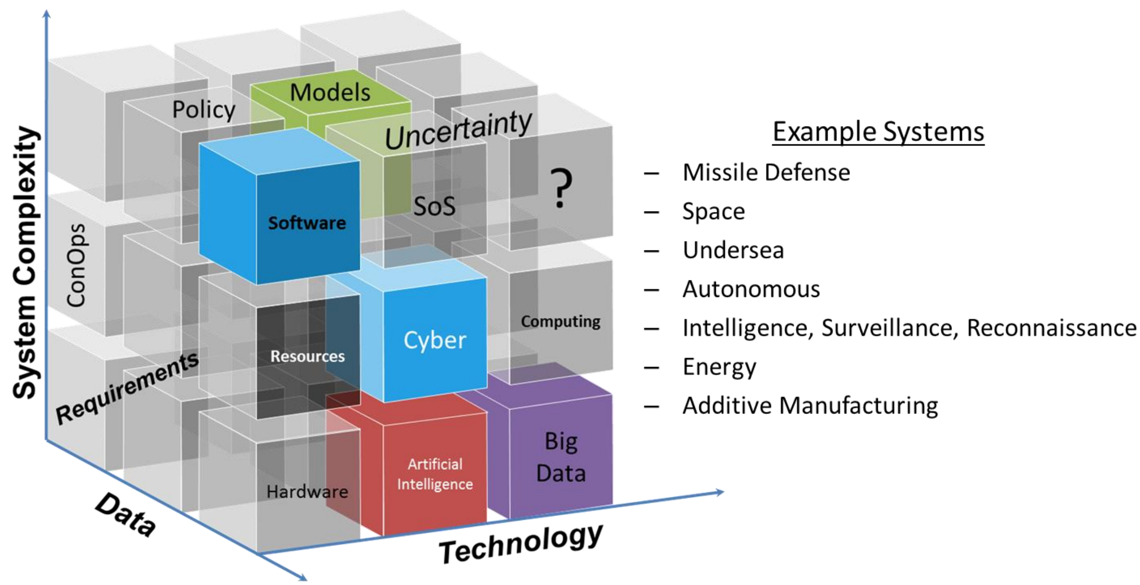

Figure 1: Complexity Drives Need for Advances in Tools and Techniques

A few examples taken from the figure highlight the issues:

Additive Manufacturing (AM) - large scale and micro AM to enable fabrication at the point of need (buildings, equipment, runways). Extreme environment components and parts in aircraft and spaceborne applications.
Big Data - (defined by Volume, Velocity, Veracity, Variety) feature technologies to advance data management and data mining systems within missile defense digital architecture. Digital twins that address Cyber Security, complex wargaming, and reduction of exascale volumes of Intelligence Surveillance Reconnaissance data.
Systems of Systems (SoS) - Space system and satellite technologies that advance state-of-the-art in payloads, subsystems, and mission operations. Naval systems that exploit vast underwater and ocean floor opportunities; healthcare systems that require extensive health information sharing (Traoré et al. 2018).
Artificial Intelligence (AI), Machine Learning, Deep Learning – Transformative data technologies that have dramatic consequences for current and future civilian and military systems.

A common denominator of the above ideas is Information Technology (IT). The observation is that systems, M&S and IT are three essential ingredients needed to advance engineering of complex systems.

2 DEVELOPMENTS IN SYSTEMS ENGINEERING

INCOSE promotes the transformation to *Model Based Systems Engineering* (MBSE) as a strategic objective. MBSE techniques are expanding to *Pattern Based Systems Engineering* (PBSE) and Agile Systems Engineering practices are studied, documented and shared globally.

The above progress is augmented by INCOSE outreach to other communities and associations such as System Science, IEEE, NDIA, and NAFEMS. Authors of this paper met at the 2017 Winter Simulation Conference and committed to do our part in connecting communities to raise awareness of activities which can further progress. This combination of outreach and collaboration across industry, academic, and professional societies is an essential ingredient to development and implementation of new methods, tools, techniques that will drive new technical and business models that do not exist today.

The INCOSE SE Vision 2020 defines MBSE as "the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases. MBSE is part of a long-term trend toward model-centric approaches adopted by other engineering disciplines, including mechanical, electrical and software. In particular, MBSE is expected to replace the document-centric approach that has been practiced by systems engineers in the past and to influence the future practice of systems engineering by being fully integrated into the definition of systems engineering processes." As practitioners, the value in the ideas and concepts behind MBSE allow for a more organized and deliberate engineering process. More importantly, MBSE provides the foundation for incorporating the growth in computing technology and the introduction of modeling standards such as SysML, UPDM, Modelica, HLA, DEVS, and others. MBSE enables integration of diverse models needed to fully specify and analyze systems. Figure 2 provides the INCOSE Roadmap for MBSE found at the INCOSE and Object Management Group MBSE wiki at http://www.omgwiki.org/MBSE/doku.php.

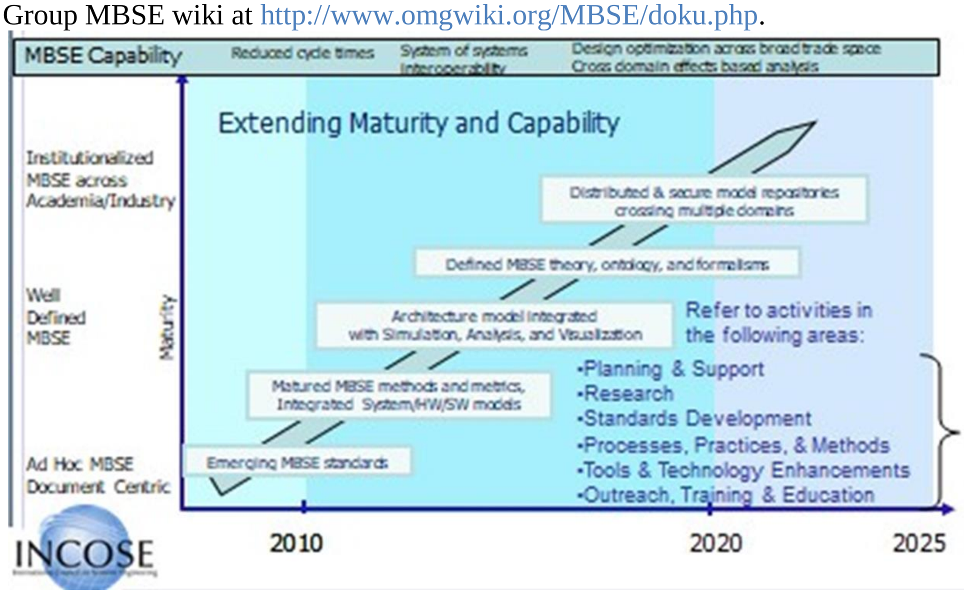

Figure 2: INCOSE MBSE Roadmap – Aggressive Steps to 2025 Goals (http://www.omgwiki.org/MBSE/doku.php)

Of special note from the Roadmap is the indication of architecture models integrated with simulation, analysis and visualization. This is followed closely by defined theory and formalisms. The Discrete Event

Simulation Specification (DEVS) is one of those formalisms described in this paper and of interest to the systems and Modeling & Simulation communities.

Invariably, complex SoS present engineers, developers, and stakeholders with Big Data challenges. Oftentimes, the ability to characterize performance, design, implement or deploy these systems is dependent upon advanced modeling and simulation techniques. This reality is due to limitations in testing and operating these systems in the diverse and dynamic environments of interest. We simply cannot test the real system in order to gain the knowledge needed for decision quality information about the system. Couple this with the desire to include AI into system operations and Big Data becomes an overriding consideration. Big Data plays heavily in complex systems and SoS. Five principal characteristics distinguish systems as SoS and are formed from the interaction of other complex systems. These five characteristics include: operational independence of the components, managerial independence of the components, evolutionary development, geographic distribution and emergent behavior (Maier 1998). When employed with components that can exhibit intelligent, adaptive, autonomous or semi-autonomous behavior resulting in feedback loops across large number of constituent components, they are better characterized as Complex Adaptive Systems (CAS) (Rouse 2008).

2.1 Pattern Based Systems Engineering

Model-based languages, such as SysML, offer a structured way to represent the requirements of SoS engineering. However, they fall short in offering a methodology to specify the hierarchical nature of SoS, ability to generate system behaviors, and to flexibly explore the ever-expanding design space. The systems engineering community is making great strides with MBSE by updating specifications (OMG SysML version 2.4) and promoting MBSE throughout the systems community. Below is a graphic that captures key features of MBSE.

An exciting extension of MBSE, promoted within INCOSE is the idea of Patterns and Pattern Based Systems Engineering (PBSE) (Schindel 2013). In PBSE, the nature of systems and SoS are investigated by using a concept called the S* Metamodel shown in Fig 2. The objective is for understanding the functional roles and interactions associated with features of the system under consideration. The significance of the S* Metamodel is its use in forming S*Models which can then be organized into Patterns. This construct is shown in Figure 3 below.

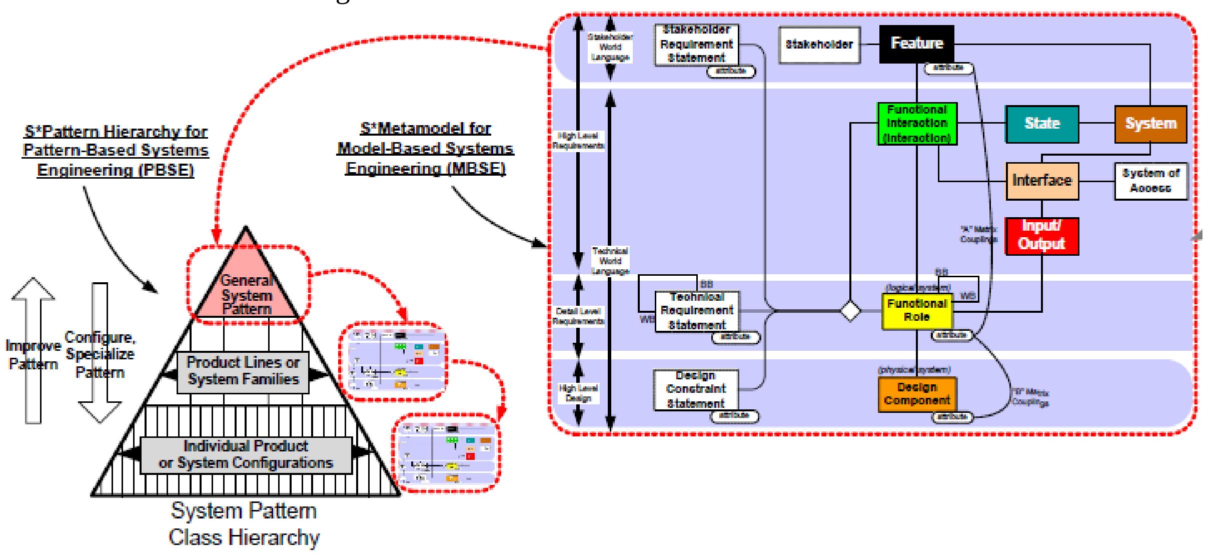

Figure 3: PBSE Captures Patterns to Address Complexity (Schindel 2013)

System Patterns are configurable, re-usable System Models that allow repeatable and structured approaches over a hierarchy of systems and system components. System Patterns can accumulate organizational learning and expertise. Because they are configurable and re-usable models of families or classes of systems, System Patterns extend the ideas of MBSE and provide a framework for progression of the MBSE Roadmap.

2.2 Agile Systems Engineering

Agile systems engineering techniques enable us to gain a better understanding of our operational environment to create more agile systems. Agile systems are better equipped to handle both internal and external environmental forces that are changing (Dove 2017). Agile systems engineering (SE) techniques give us an awareness of the interfaces we must consider during development of critical artifacts of M&S (to be described in Section 4), so that our system will accommodate for interactions with its environment, both internally and externally.

The Agile SE CURVE technique helps characterize the problem-space and build agility in our M&S artifacts. The CURVE acronym represents the following categories of situational responses:

- **Capriciousness** Randomness among unknowable possibilities.
- **Uncertainty** Randomness among known possibilities with unknowable probabilities.
- **Risk** Randomness among known possibilities with knowable probabilities.
- **Variation** Randomness among knowable variables and associated variance ranges.
- **Evolution** Gradual successive developments.

From this type of analysis, we gain an awareness that enables our models to better represent the real-world scenarios they intend to capture. INCOSE and SCS communities seek to show agility in their work. CURVE helps us think about the scope and purpose of M&S in systems development. We can consider each potential risk item listed above and determine if M&S will help address the issue. If we decide to address an issue with M&S, we can consider the combinatorial possibilities of the environment during development. We can add the model to a repository and utilize it for co-simulations. This can enable future use of the model to inform a larger, hierarchical system of systems. In agile systems engineering, this aligns with Reconfigurable-Reusable-Scalable (RRS) design principles (Dove 2017), namely, the Encapsulated Modules, Facilitated Interfacing, and Facilitated Reuse principles.

3 DEVELOPMENTS IN MODELING & SIMULATION AND DEVS

Such challenges result in high complexity that cannot be easily tackled using classic modeling, simulation, and optimization techniques. Recent model-based system engineering has proved inadequate due to lack of a full-strength M&S computational substrate (Mittal, Durak and Oren 2017; Mittal and Martin 2017). Modeling and Simulation methodology has been evolving to provide increasing capability to help systems engineers develop models of SoS (Zeigler and Nutaro 2016). Such simulation models support design and testing of mechanisms with learning capabilities to coordinate the interactions of the operationally and managerially independent components. Design of such systems present challenges to the currently employed independent use of simplified models for formal verification or brute-force simulations which are severely limited in the range of conditions they can test. Modeling and simulation of CAS (and the underlying SoS) must have a usable modeling environment that facilitates model validation from the end-user and a robust simulation infrastructure that can be formally verified to ensure correct model execution. Together, they enable exhaustive parameter evaluation and advanced experimentation. Model-based methods which support traditional systems engineering need to be augmented with simulation-based methodologies to ensure they support complex systems engineering that integrate discrete and continuous systems for complex hybrid systems. CAS engineering will not become possible unless the undesired emergent behaviors are completely removed from a computational environment or are known a priori so that they can be knowledgeably eliminated. A computational simulation-based environment provides experimentation opportunities to validate a CAS model, such that it becomes predictable and eventually useful (Mittal and Martin 2017).

The task of integrating various simulators to perform together as a composite simulation, termed as co-simulation, involves weaving the time series behavior and data exchanges accurately, failure of which, will yield inaccurate simulation results. As elaborated by Mittal and Zeigler (2017), every such hybrid system would require a dedicated effort to build a co-simulation environment. Bringing various simulators together is much more than a typical software engineering integration exercise.

A solution gaining increased acceptance is offered by the DEVS formalism with a holistic construct called the Modeling and Simulation Framework (MSF). In a brief review, we say that the framework defines the entities and their relationships of the enterprise of M&S and includes the relation between detailed models and their abstractions (Zeigler, Muzy and Kofman 2018). The framework is based on mathematical systems theory and recognizes that the complexity of a model can be measured objectively by its resource usage in time and space relative to a particular simulator, or class of simulators. Furthermore, properties intrinsic to the model are often strongly correlated with complexity independently of the underlying simulator. Successful modeling can then be seen as *valid simplification*, i.e., reduction of complexity to enable a model to be executed on resource-limited simulators and at the same time, creating *morphisms that preserve behavior and/or structural properties*, at some level of resolution, and within some experimental frame of interest. Indeed, according to the framework, there is always a pair of models involved call them the *base* and *lumped* models.

The DEVS formalism is formulated within MSF and formally specifies the internal behavior of the system as well as macro behavior of the overall system due to its closure under coupling property. This robustness in both structural and behavioral description ensures that the unwanted holistic behaviors, also known as negative emergent behaviors are explicitly avoided, along with the guaranteed manifestation of the desired (or positive) emergent behaviors (Mittal 2013; Mittal and Martin 2013; Zeigler and Nutaro 2016). The DEVS super-formalism provides a foundation (Mittal and Zeigler 2017) that specifies an abstract simulation protocol between the model and the simulator (Zeigler, Muzy and Kofman 2018). Thus, a requirement for M&S of CAS is to employ the principles of the Parallel DEVS simulation protocol (as illustrated by the hybrid approach of Camus et al. (2018), for example) to support the required robust co-simulation.

3.1 MBSE, DEVS and CAS: Towards Unification

We have seen that MBSE and PBSE call for formalized models to replace documents as the fundamental building blocks of systems engineering and that practicality demands that such models eventually support all the activities typically associated with the simulation discipline. However, as suggested current MBSE formalisms stop well short of this capability. One approach to bridging this gap is to enable mappings to be defined that precisely specify simulation models that realize their behaviors. Taken to practical limits, this approach entails building more capability into such a formalism so that it eventually replicates all capabilities associated with traditional simulation methodology. Although there are attempts to achieve this goal (Bocciarelli et al. 2018; Amissah et al. 2018), there are also fundamental reasons why it is not attainable (Aliyu et al. 2016; Abdurrahman and Sarjoughian 2018). An approach that we hypothesize to work is to tie MBSE models with informal but well documented links. Further, as experience grows with such cross-linkages it might eventually become feasible to formalize these associations.

Going beyond MBSE to PBSE, an emerging paradigm is to look for patterns that can be provided by PBSE that can be invariant to changes as a CAS evolves. Familiar patterns can be reused as new noble causes are encountered. More specifically, PBSE plays into CAS in the following way. If we create the smallest possible model of many system components, these components can be combined into any number of CASs. The result is that we have many components we can draw from over time. This saves modeling time spent on common components allowing us to get at unique CAS model components. We want to reduce time to model by reusing components and patterns where possible. This is an important form of knowledge accumulation and a major consideration for decision makers who don't have the time nor resources to dedicate for modeling activities.

3.2 Architecture and workflow for M&S working within MBSE

Figure 4 outlines an architecture and workflow for M&S working toward production of models for use in MBSE. The process starts with the development (or reuse) of a System Entity Structure (SES) that organizes a family of simulation models for the current application of interest. SES is an ontology, a language with syntax and semantics to represent declarative knowledge (Pawletta et al. 2016). The SES is a knowledge representation scheme that structures the search for a subset of models that are of particular concern under criteria that relate essentially to their behavior and can't be defined in the first instance by their structural properties. Indeed, the behavior generated under simulation is observed within the Experimental Frame that characterizes the criteria defining the subset of interest. Roughly, an experimental frame (EF), as defined within the MSF, is a specification of the conditions under which the system is observed or experimented with. As such, experimental frames are the operational formulation of the criteria that motivate the M&S-based pursuit of the models of interest. In an example of defense interest, types and placements of launch sites that collectively create high damage may be the focus of interest while

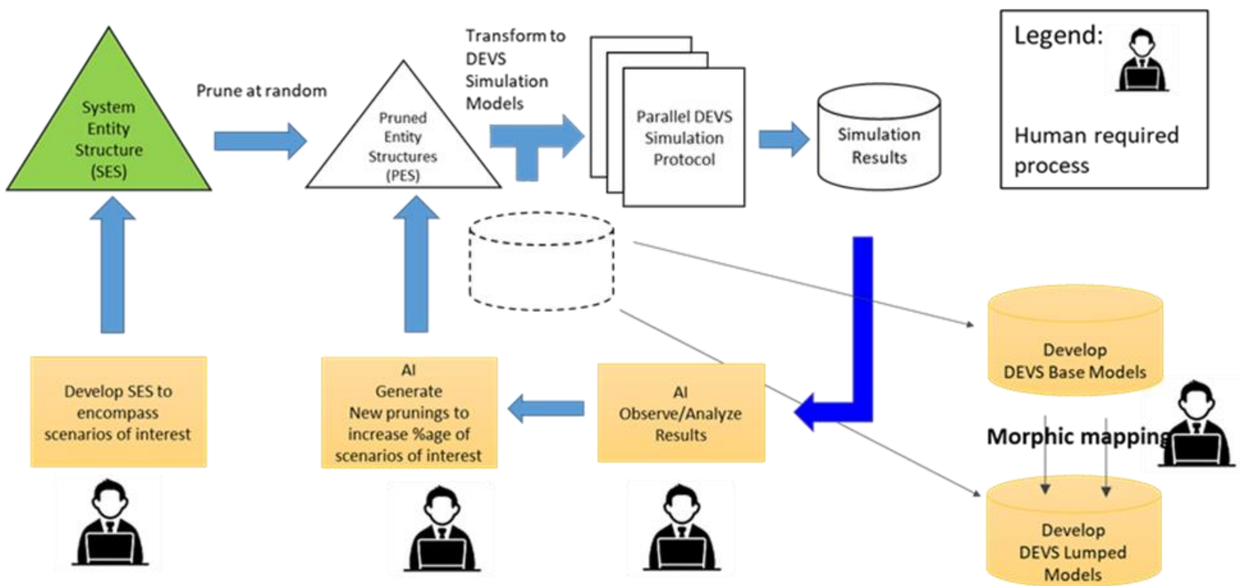

Figure 4: Architecture and Workflow for M&S working within MBSE

employing a model-base of suitable missile defense system models. The SES includes coupling information that directs the compositions of hierarchical models from components in the model-base. The combination of coupling opportunities for options for selection from specializations and aspects, leads to a very high combinatorial search space. Since an SES describes a number of system configurations, the SES tree needs to be pruned to get one particular configuration, which is called Pruned Entity Structure (PES). Pruning operations factor out a particular model specification which can then be transformed automatically into a coupled model with components from the model base. Such components are either DEVS models or have been wrapped in a DEVS interface for DEVS compliance and amenability to the coupling specified by the SES. Simulation of such a model, eventually on a high-performance platform using parallel simulations of multiple models under test for reasonable execution times, generates the behavior of the model and produces results in the experimental frame of interest. These results measure the extent to which the governing criteria are satisfied and are analyzed for guidance to direct the pruning toward a larger percentage of models that fully satisfy the criteria. At this point AI is useful to help analyze the results and predict which new prunings of the SES should performed at the next iteration. Built into the iteration loop is a second cycle of transition between base and lumped models where the lumped model can greatly accelerate the search for high-value models by enabling faster runs that provide useful information for the more detailed base model. Some fundamental distinctions between base and lumped models concern objectives, representation, entity attributes and variables, interaction processes, timing mechanisms, and computational complexity (Zeigler, Muzy and Kofman 2018).

As illustrated in Figure 3, the architecture envisions a collaboration between human and AI agents. The Human modeler develops the SES and the DEVS model base to span configuration space that encompasses the subset of interest. The AI agent, under control of the user, analyses the results and generates new prunings in order to increase percentage of models of interest. Human modeler develops valid simplification morphisms for the DEVS base and lumped models and decides when and how to iterate between the levels of resolution in order to accelerate the overall process (Zeigler and Nutaro 2016).

4 CONCLUSIONS: CONVERGENCE TOWARD NOBLE CAUSES

We have identified Big Challenges where engineering practices are scaled to “next-level” triumphs: software defined systems that must increasingly operate on large, time-varying, heterogeneous data. Big Data enables and requires that these systems perform across an enormous variety of operating conditions presenting engineers with multi-dimensional, hierarchical, uncertain and critical control and decision challenges. Recent work has begun to address these challenges. Kavak et al. (2018) offer a structured modeling approach to produce agents or parts thereof directly from data that focuses on individual-level data to generate agent behavioral rules and parameter values Generalizing from the approach that recently enabled the AlphaGo program to defeat the world’s top ranked human Go player, Wang et al. (2018) envision an AlphaGo-like computation platform to enable artificial systems to model and evaluate complex systems, and through the virtual-real system interaction, realize effective control and management over the complex systems.

Here we have taken a more fundamental perspective on the Big Challenges. Based on Systems thinking, concepts, theory, modeling, and simulation we have identified a convergence in motion between the model-based systems engineering initiative of the INCOSE community and the model -based simulation developments of the SCS community. The trends toward replacing documents with models as the basis for knowledge accumulation, supported by the identification of Patterns for reuse align with similar trends toward DEVS-based simulation. Moreover, the Agile/CURVE framework that enables models to better represent the real-world scenarios offers an approach to M&S to deepen the capabilities of its simulation models that enable more extensive and validated exploration virtual-test. The architecture of Figure 3 offers a generic workflow that supports Wang’s et al. (2018) vision of AlphaGo-like computational strength for future M&S-based systems engineering and management. The Modeling and Simulation Framework (Zeigler, Muzy and Kofman 2018), and in particular its system specification hierarchy for acquiring levels of knowledge about an observed system, provide a solid basis for inference of structures from the volumes of Big Data envisioned by Kovak et al. (2018).

Once the framework has been built, it becomes easier for the integration of specialty engineering all the way out to the users. For future work, we recommend developing tool sets that expand out to include dealing with the “illities” of systems engineering (reliability, availability, etc.). The goal is to make it easier for designers to deal holistically with complex adaptive systems of systems, including maintaining round trip design consistency and keeping patterns/models valid as evolution proceeds. Once such tools become common, it will be possible to include more of the human elements (doctrine, tactics, procedures, governance) and stakeholders (users, partners, regular people, soldiers) directly in the process.

We have mentioned the goal of trying to manage the bad aspects of emergence while preserving its good qualities. A reviewer points out that this is reminiscent of Whitehead and Russell’s attempts to control self-reference in Formal Mathematics in the Principia Mathematica which they eventually proved to paradoxical in nature. Principia's solution of hierarchical set constructions may suggest a way forward in the SoS case and the DEVS formalism offers a ready-made solution for hierarchical model construction justified by closure under coupling (Zeigler, Muzy and Kofman 2018).

It seems clear that we need to get a better handle on the whole SoS life-cycle with a more deliberate combined MBSE/DEVS approach. This will help us focus on the problem and better understand the five attributes of SoS that underlie and interact to induce emergence.

Still we should recognize enormous obstacles that must be overcome to achieve these visions. Progress may require new ways of thinking about systems that truly enable them to be developed with reusable components. We must become able to identify the limitations in dealing with Big Data and limitations in dealing with its multi-dimensional, hierarchical, and uncertain nature. Along these lines, Zeigler et al. (2018) have identified strong requirements that must be satisfied to enable DEVS-based M&S to be practiced at its most productive level. These prerequisites for best practice include: a) developing an effective operational ontology, b) enabling the ontology to support combinatorial model compositions, c) including the major facets to ensure representation of all levels (macro, meso, and micro) of behavior, d) curation of a large spectrum of models for combinatorial composition, and perhaps most critically, e) instrumenting the complex SoS to support acquisition of on-going high quality data (Mittal and Martin 2013). Working to put the infrastructure in place to meet these requirements will move both systems and M&S communities along realistic paths toward the convergence to noble causes that we all envision.

REFERENCES

Abdurrahman A I, Sarjoughian. H., Model -Driven Time-Accurate DEVS-Based Approaches For Design, TMS/DEVS, SpringSim, 2018.

"Achieving Science with CubeSats: Thinking Inside the Box," Division on Engineering and Physical Sciences, National Academies of Sciences, Engineering, and Medicine, National Academies Press, 2016.

Aliyu, Hamzat Olanrewaju, Oumar Maïga, Mamadou Kaba Traoré. 2016. The high level language for system specification: A model-driven approach to systems engineering, Int. J. of Modeling, Simulation, and Scientific Computing,(07) 11, https://doi.org/10.1142/S1793962316410038.

Amissah, Toba, Handley, Seck, "Towards a Framework for Executable Systems Modeling: An Executable Systems Modeling Language (ESysML), SpringSim 2018.

AS-2 Embedded Computing Systems Committee SAE. Architecture Analysis & Design Language (AADL). SAE Standards no. AS5506C (2017).

Bocciarelli, Ambrogio, Giglio, Paglia, "Model Transformation Services for MSaaS Platforms" SpringSim 2018.

Boydston, A. and P. Feiler, S. Vestal, B. Lewis, "Joint Common Architecture (JCA) Demonstration Architecture Centric Virtual Integration Process (ACVIP) Shadow Effort", AHS 71st Annual Forum (2015).

Camus, B. et al. 2018, Co-simulation of cyber-physical systems using a DEVS wrapping strategy in the MECSYCO middleware, Simulation J. Published online, Jan. 2018 doi/10.1177/0037549717749014.

Cox, Matthew, "Army Sees Rapid Prototyping as Key to Rapid Innovation", Defensetech.org, April 2015, defensetech.org/2015/04/01/army-sees-rapid-prototyping-as-key-to-rapid-innovation.

Cox, Matthew, "Mobile Labs Build On-the-Spot Combat Solutions", Military.com, Aug. 2012, military.com/daily-news/2012/08/17/mobile-labs-build-on-the-spot-combat-solutions.html.

Dove, R. 2017. Agile Systems & Processes 106: Risk Management and Mitigation. INCOSE Webinar 104. International Council on Systems Engineering. September 20. Video: www.parshift.com/s/AgileSystems-106.mp4. Slides: www.parshift.com/s/AgileSystems-106.pdf.

International Council on Systems Engineering (INCOSE), 2014, Systems Engineering Vision 2025 July, 2014; Accessed July 6 at http://www.incose.org/docs/default-source/aboutse/se-vision-2025.pdf?sfvrsn=4

Kavak, H, Lynch, C., Padilla, J and Dialio, S., Big Data, Agents, and Machine Learning: Towards A Data-Driven Agent-Based Modeling Approach, SpringSim, 2018.

Maier, M.W. 1998. Architecting Principles for Systems-of-Systems, Systems Engineering, 1(4): 267-284.

Mittal, S. 2014. Model Engineering for Cyber Complex Adaptive Systems, European Modeling and Simulation Symposium, Bordeaux, France.

Mittal, S. and B. P. Zeigler. 2017. Theory and Practice of M&S in Cyber Environments, In A. Tolk, T. Oren, (eds.) The Profession of Modeling and Simulation: Discipline, Ethics, Education, Vocation, Societies and Economics, Wiley & Sons.

Mittal, S. and J. L. R. Martin. 2017. Simulation-Based Complex Adaptive Systems July 2017 In book: Guide to Simulation-Based Disciplines.

Mittal, S. and J. L. R. Martin. 2013. Netcentric System of Systems Engineering with DEVS Unified Process, CRC Press. 1st edition.

Mittal, S. and J. L. R. Martin. 2017. Simulation-based Complex Adaptive Systems, in S. Mittal, U. Durak, and T. Oren (eds). Guide to Simulation-based Disciplines: Advancing Our Computational Future, Springer AG.

Mittal, S. Durak, U., Oren, T. 2017. Guide to Simulation-based Disciplines: Advancing our Computational Future, Springer AG.

Pawletta T., Schmidt A, Zeigler B.P., Durak U. Extended Variability Modeling Using System Entity Structure Ontology Within MATLAB/Simulink. *Proceedings of the 49th Annual Simulation Symposium*, Pasadena, California, 3-6 Apr 2016. Article 22.

Rouse W, 2008, Health Care as a Complex Adaptive System: Implications for Design and Management, The BRIDGE Jounal of the National Academy of Engineering, Spring 2008, pp 17-25.

Schindel, Bill, Troy Peterson, “Introduction to Pattern-Based Systems Engineering (PBSE): Leveraging MBSE Techniques”, in Proc. of INCOSE 2013 Great Lakes Regional Conference on Systems Engineering, Tutorial, October, 2013.

Traoré, Mamadou Kaba, Gregory Zacharewicz, Raphaël Duboz, and Bernard Zeigler, Modeling and Simulation Framework for Value-based Healthcare Systems, Simulation J., 2018.

Wang, Fei-Yue, et al., Where Does AlphaG0 Go: From Church-Turing Thesis to AlphaGo Thesis and Beyond, IEEE/CAA JOURNAL OF AUTOMATICA SINICA, VOL 3, NO 2, APRIL 2018.

Wells, Lee J.; Camelio, Jaime A.; Williams, Christopher B.; White, Jules, "Physical Security Challenges in Manufacturing Systems", Manufacturing Letters, Vol 2(2), April 2014, pp. 74-77.

Zeigler, Bernard P., Mamadou Kaba Traoré, Saurabh Mittal, Fundamental Requirements and DEVS Approach for Modeling And Simulation Of Complex Adaptive System of Systems: Healthcare Reform, SpringSim, 2018.

Zeigler, Bernard P and James J. Nutaro Towards a Framework for More Robust Validation and Verification of Simulation Models for Systems of Systems, JDMS, 2016.

Zeigler, Bernard P., Muzy A., Kofman E. *Theory of Modeling and Simulation,* 3nd Edition. New York: Academic Press; 2018.

AUTHOR BIOGRAPHIES

BERNARD ZEIGLER Bernard P. Zeigler (F'95) is Professor Emeritus of Electrical and Computer Engineering at the University of Arizona, Tucson, AZ, USA and Chief Scientist of RTSync Corp., Phoenix, AZ, USA. He is internationally renowned for his seminal contributions in modeling and simulation theory. He is well known for the Discrete Event System Specification (DEVS) formalism that he invented in 1976 and which is now being used world-wide in advanced information systems. Dr. Zeigler is a Fellow of the SCS and received the INFORMS Lifetime Achievement Award. In 1995, he became Fellow of the IEEE in recognition of his theory of discrete event simulation. He is an Emeritus Professor with the Arizona Center of Integrative Modeling and Simulation, Tempe, AZ, USA, and a Chief Scientist with RTSync Corp, Tempe, AZ, USA, and MD, USA. His e-mail address is zeigler@rtsync.com.

JOSEPH MARVIN is an INCOSE Expert Systems Engineering Professional (ESEP) who has career experience as a research and development systems engineer. His experience includes U.S. Air Force Research and Development Engineer – Space Systems, and industry experience with Lockheed Martin and SAIC where he supported multi-year Major System Acquisitions in military communications and National Security Space programs. He is the founder of Prime Solutions Group, Inc and leads the company research activity with government, industry and academia. These efforts were recognized by the selection of the company as the IEEE Phoenix Sector Small Business of the Year 2017. Joe is the former Chair of the INCOSE Very Small Enterprise Working Group (VSE WG) and currently serves as the INCOSE Assistant Director for Internal Operations. His email address is joemarvin@psg-inc.net .

JOHN CADIGAN is an INCOSE Associate Systems Engineering Professional (ASEP). He currently serves as the Systems Engineering Lead at Prime Solutions Group, Incorporated (PSG) located in Goodyear, AZ. At PSG John conducts research on Big Data elements in energy and defense programs. He leads the PSG Senior Capstone projects and serves as PSG's engineering representative to professional and technical organizations. John graduated Magna Cum Laude from the Ira A. Fulton Schools of Engineering at Arizona State University with a major is Engineering Management. John was chosen for the prestigious INCOSE Leadership Institute in 2018. His email address is johncadigan@psg-inc.net.