Chapter 1
Cloud-based M&S for Cyber Physical Systems Engineering

José L. Risco-Martín and Saurabh Mittal

Abstract Cyber Physical Systems (CPS) are complex systems that have two essential elements: the cyber element and the physical element. These are analogous to the early hardware/software (HW/SW) systems that predated Internet. CPS are fundamentally HW/SW systems with an additional capability of being remotely controlled, which introduces significant amount of risk in CPS operations. Today, such CPS leverage cloud computing infrastructures to provide scale and wider usage and as such many such systems are remotely deployed as well. The use of Modeling and Simulation (M&S) in HW/SW engineering is standard practice but incorporating M&S in cloud environment to support CPS engineering brings forth a new set of challenges for both the M&S technology and CPS engineering methodologies. This chapter will provide an overview of M&S Cloud computing technology and its impact on CPS engineering, along with various challenges. It will also provide a brief overview of the chapters that follow.

1.1 Introduction

According to a definition provided by the National Science Foundation (NSF), Cyber-Physical Systems (CPS) are hybrid networked cyber and engineered physical elements co-designed to create adaptive and predictive systems for enhanced performance. These systems are built from and depend upon the seamless integration of computation and physical components. NSF early identified CPS as “a key area of research” [1]. Examples of CPS include

---

José L. Risco-Martín
Department of Computer Architecture and Automation, Universidad Complutense de Madrid, C/Prof. José García Santesmases 9, 28040 Madrid, Spain, e-mail: jlrisco@ucm.es

Saurabh Mittal
The MITRE Corporation, Fairborn OH, USA, e-mail: smittal@mitre.org

autonomous automobile systems, automatic pilot avionics, industrial control systems, medical monitoring, smart grid, and robotics systems.

A typical CPS is comprised of the following components [2]:

- Sensors
- Actuators
- Hardware platforms (that host sensors and actuators)
- Software interfaces (that access hardware directly or remotely through a cyber environment)
- Computational software environments (that may act both as controller or service provider)
- Networked environments (that allow communication across geographical distances)
- End user autonomy (that allows CPS to be used as a passive system or an active interactive system)
- Critical infrastructures (water, power, etc., that provide the domain of operation and operational use-case)
- Ensemble behaviors
- Emergent behaviors

Figure 1.1 shows various aspects of CPS divided into Left-Hand Side (LHS) and Right-Hand Side (RHS). LHS consists of a collection of users, systems (both hardware and software) and devices (physical platforms). Traditional systems engineering practices and end-user use-cases can be developed in LHS. The RHS shows aspects related to infrastructures. Fundamentally, they can be characterized into Information Technology (IT) and Operational Technology (OT). Between the LHS and RHS is the network/cyber environment that allows information exchange between the two. With the network spanning large geographical distances, the presence of a large number of entities/agents and their concurrent interactions in the CPS result in ensemble and emergent behaviors. The "infrastructure-in-a-box" is largely unavailable but can be brought to bear with various existing domain simulators in an integrated simulation environment.

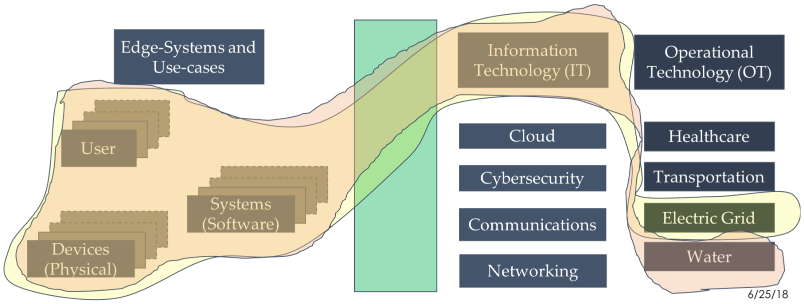


Fig. 1.1 CPS Landscape (reproduced from [2])

Today more and more CPS design problems are reaching insoluble levels of complexity. System complexity continues to grow by leaps and bounds. Multi-level complexity is a fundamental nature of heterogeneous systems today. Current CPS may incorporate large scale systems, which may be under independent operational and managerial control. These are better termed as Industrial CPS. The Internet of Things (IoT), which is also considered as a CPS, is beginning to incorporate all of these characteristics and is becoming a significant contributor to the increase in complexity. However, the IoT phenomenon is still in the formative stages of apparently exponential growth. Designing these systems is equally complex and methodologies available through traditional systems engineering practices fall short of engineering these complex systems [3]. New methods are needed to advance the engineering practices. Research must advance in two directions: (1) methodological, defining standards, languages and protocols to handle such complexity, and (2) technological, adapting new engineering processes to the new available computing infrastructures, so the technology could be applied at multiple levels of CPS specifications.

A challenge in the development of CPS is large differences in the design practices between all the involved engineering disciplines, like software engineering or systems engineering. The emergence and proliferation of CPS has proved that the historical distinction between software engineering and systems engineering is becoming less relevant. Understanding physical, biological, artificial, and social systems requires a well-founded, formal, yet intuitive methodology and language that is capable of modeling the complexities inherent in these systems in a coherent, straightforward manner. Considering the current scenario where rapid innovations are assumed to be essential, engineers must be able to explore and exploit software and systems designs collaboratively, analyzing trade-offs and obtaining rapid conclusions. A solid Modeling and Simulation (M&S) methodology to guide all these processes will allow disciplines to cooperate seamlessly [4]. In this regards, the realization that models should serve as foundational architecting and design artifacts has started to gain momentum among both software engineers and systems engineering professionals, resulting in model-based methodologies. For software engineers, this happened in the early 1990s. After the object-oriented paradigm, the Unified Modeling Language (UML) was adopted in 1997 under the auspices of the Object Management Group with nine different diagram types, which grew to 13 with the transition to UML 2.0 in 2005 [5]. In response for systems engineers, the System Modeling Language (SysML) was developed and adopted in 2007. Like UML 1.x, it had nine diagram types, but not exactly the same set; some were removed from UML 2.0, some modified, and new diagrams were added [6]. Later, the community developed methodologies to integrate software and systems engineering to perform software-systems engineering. However, these practices fall short of doing multi-domain systems modeling and simulation.

UML and SysML based approaches were largely used to develop Information Technology (IT) systems and found limited use with formal systems engineering practices that required heavy engineering analysis. For example, an electrical system, or a control system that relied on scientific disciplines like Electrical Engineering and Control Theory. To date, formal engineering methodologies that have been used to develop hardware-software systems in the past four decades are finding limited used with UML/SysML for their systems engineering endeavors. They still rely on engineering methodologies that are rooted in the scientific discipline, such as Electrical Engineering, to engineer closed systems. A closed system is a system that can be expressed in a closed form using a scientific theory. Formal modeling approaches using formalisms such as Discrete Event System (DEVS) and Colored Petri Nets are extensively used due to their verification and validation rigor. When many such closed systems (physical components in CPS) are required to operate with other systems across the network, along with human interaction as an integral part of functioning, new architectures are needed. The nature of architectures also changed to a distributed one, infusing new set of challenges.

To facilitate architecture development, various architecture frameworks came into existence in the next decade that facilitated interconnection between various component systems (e.g., US Department of Defense Architecture Framework (DoDAF), UK Ministry of Defense Architecture Framework (MODAF), Unified Architecture Framework (UAF) etc.). However, these architecture frameworks were described using UML/SysML and did not mandate any M&S paradigm, theory or a tool to perform simulation-based systems engineering for closed or hybrid systems. Consequently, the gap between a formal model specified through a user-friendly representation, that is implemented and executed correctly by a simulator, still remains. To fill this gap, work by Mittal and Martin [7] on DEVS Unified Process (DUNIP) and the incorporated DEVS Modeling Language (DEVSML) Stack [8, 9] using the DEVS formalism, defined a methodology and an abstract language for conceptual modeling and complex systems architecting that integrated the conceptual, structural and functional aspects of the modeled system.

Contemporary CPS are considered as a new trend in IoT, where the physical systems are the sensors that collect real-world information. This information is transferred to the cyber layer, i.e., the computational modules. The cyber layer analyzes and notifies the findings to the corresponding physical systems through a feedback loop [10]. In the CPS model, the integration of cloud technologies in the CPS cyber layer to ensure scalability, communication and computation limits the model's accuracy, and adequate energy consumption is highly recommended. A cloud-based M&S architecture can facilitate the deployment of cloud-based CPS in two phases. The first one is the conceptual phase, when the system is initially conceived. Here a solid M&S architecture is fundamental to have an initial idea of the structure and behavior of the whole system with a reduced cost. Because of the inherent distributed nature of current CPS, this M&S framework must support distributed system design as well [11]. The second phase is the production system, i.e., when the system is actually deployed in the real world.

This process can be done based on the 5C architecture: Connection, Conversion, Cyber, Cognition, and Configuration [12]. In the *Connection* level, devices are designed to auto-connect and auto-sense its behavior. In the *Conversion* level, meaningful information has to be inferred from the data. In the *Cyber* level, information is being pushed to it from every connected machine to form the machine-to-machine network. An efficient methodology for managing and analyzing information at this level is creating "digital twins" to study performance for further analysis. The *Cognition* level generates knowledge of the monitored system. Proper presentation of the acquired knowledge to expert users supports the correct decision to be taken. Finally, in the *Configuration* level, the production system can be reconfigured based on the priority and risk criteria, i.e., it acts as a resilience control system. Figure 1.2 illustrates the relation of the 5C and IoT layers. A contemporary M&S architecture can facilitate the design of twin models, in which physical operations are coupled with virtual operations using intelligent reasoning agents.

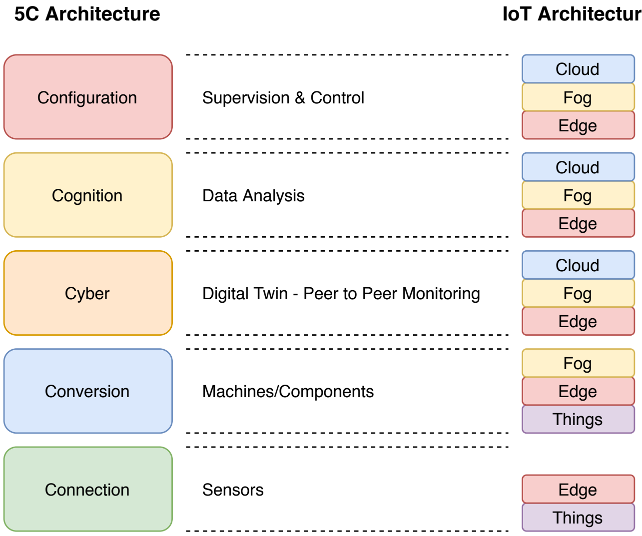


Fig. 1.2 Relation between 5C and IoT architectures

Since cloud infrastructure usage is becoming ubiquitous in our day-to-day life, an integrative Cloud-based M&S architecture for CPS engineering provides a bridge between design practices in different engineering disciplines, physical CPS layers and application CPS layers. As a result, CPS application design, reconfiguration and autonomy becomes features that can now be explored in an efficient manner.

This chapter provides an overview of M&S Cloud computing as a universal methodology to act as a medium to conceptualize current complex CPS and the technology to handle the design and evolution of contemporary IoT/CPS applications. The chapter is organized as follows. Section 1.2 provides current aspects and architectures of Cloud-based M&S and ongoing community efforts. Section 1.3 describes some challenges and cloud implications with Cloud-based M&S CPS engineering. Section 1.4 provides a quick overview of the book chapters. Section 1.5 concludes the chapter.

1.2 Cloud-based M&S

Modeling and simulation are two distinct activities. Likewise, cloud-based modeling and cloud-based simulation require different capabilities. Fundamentally, they require that the digital infrastructure be deployed in cloud environment and is accessible through remote mechanisms. While modeling activity requires editor workbenches that may be accessible through an Internet browser, the simulation activity requires specific simulation architecture to be cloud compliant. Transitioning any existing M&S application in a cloud-based environment requires explicit M&S infrastructure engineering. Additionally, cloud-based M&S has challenges that traditional simulation systems engineering has largely solved [13]. However, as Mittal and Tolk [2] discuss, CPS are multi-modal and multi-domain systems that involve domain-specific concepts and architectures from more than one domain.

Cloud-based M&S foundation is built on offering modeling as a service and simulation as a service. This implies that both the model engineering and simulation engineering activities must be service-oriented. Further, these services must be deployed in a cloud environment for realizing a cloud-based M&S solution. Model Engineering (ME) [14] incorporates full lifecycle management of the model and credibility to the model engineering process, which includes establishing standards, theory, methods and tools for doing modeling, and the management of the model, data, knowledge, activities and organizations/people involved in the model engineering process in a collaborative manner. ME, to be cloud-deployed, requires the availability of a model repository and the execution of the ME process through a browser-based access mechanism. There exist many such tools (e.g. NoMagic Cameo, IBM Rhapsody, Eclipse IDEs, etc.) that facilitate ME in a cloud environment. Simulation engineering incorporates the execution of model in a computational environment by a simulator and various tools and software dependencies that are required by the simulator. Deploying a simulator in a cloud-environment is not straightforward though. Simulators may require high computational resources and the High Performance Computing (HPC) community has been working on bringing large computational resources for simulation execution for quite a long time. Leveraging the HPC community body of work of the past few decades and masking it behind the service interface for cloud-enabled access is indeed the easiest solution when the simulation system is purely a software system. As Mittal and Tolk [15] explore in their recent book, CPS M&S takes the form of a Live, Virtual and Constructive (LVC) system. LVC system incorporating simulators at varying levels of fidelity involve both hardware and software components. While one can make the software (purely constructive) components available in a cloud environment, bringing virtual (may include hardware) and live (hardware and hybrid) components is not practical and requires simulation engineering to rely on specific technologies, standards and methods developed by the Distributed Simulation Engineering community engaged in LVC SoS engineering.

Simulation Interoperability Standards Organization (SISO) stood up the Cloud-Based Modeling and Simulation (CBMS) Study Group in 2016 [16] under the leadership of Col. Robert Kewley, to identify and document the existing M&S in the cloud activities, document best practices, highlight lessons learned and identify various potential standards to facilitate adoption by other practitioners. Their focus was strictly on CBMS, and application to CPS was out-of-scope. The group identified several focus areas or themes into which the efforts and ideas could be organized to answer important questions or to identify best practices. Potential themes included (but are not limited to):

1. Developing composable services
2. Service discovery
3. Security
4. Deployment, management and governance of services
5. DEVS modeling and other alternative modeling frameworks
6. Development of a reference architecture
7. Service oriented architectures
8. Business case analysis and return on investment
9. Application of the Distributed Simulation Engineering and Execution Process (DSEEP)
10. The emerging role of the cloud service provider
11. Impact on Validation, Verification and Accreditation (VV&A) practices
12. Data services (including terrain)

The CMBS study group organized the group in four broad areas:

- Models, simulators and data: Analyze cloud M&S efforts from an interoperability perspective for the models, simulators and data aspects. This includes investigating semantic model interoperability, simulation architectures and handling of structured and unstructured data.
- Architecture: Synthesize concepts and constructs from the current M&S architectures into a coherent vision for the future optimized for modern cloud computing and Big Data environments. One of the goals is to enable interoperability with legacy architectures while providing an unconstrained path to the future.

- Cloud infrastructure: Investigate the impact of cloud computing technologies on various aspects of M&S, including system scalability, advanced visualization, scalability of data systems and high bandwidth or low latency connections to compute and memory. Also investigate ease of integration into the internet-of-things type scenarios, which is similar to embedding M&S into live military hardware.
- Services: Investigate, propose and evaluate standards, agreements, architectures, implementations and cost-benefit analysis of Modeling and Simulation (M&S) as a Service (MSaaS) approaches.

The CBMS literature survey is available in [13]. The CBMS SG produced a report [17], pending approval by SISO Standards Activity Committee. The CMBS SG also synchronized their effort with North Atlantic Treaty Organization (NATO) Modeling and Simulation Group (MSG) effort named NATO MSG-136 [18]. NATO MSG-136 effort developed a Reference Architecture for M&S as a Service (MSaaS) (Fig. 1.3).

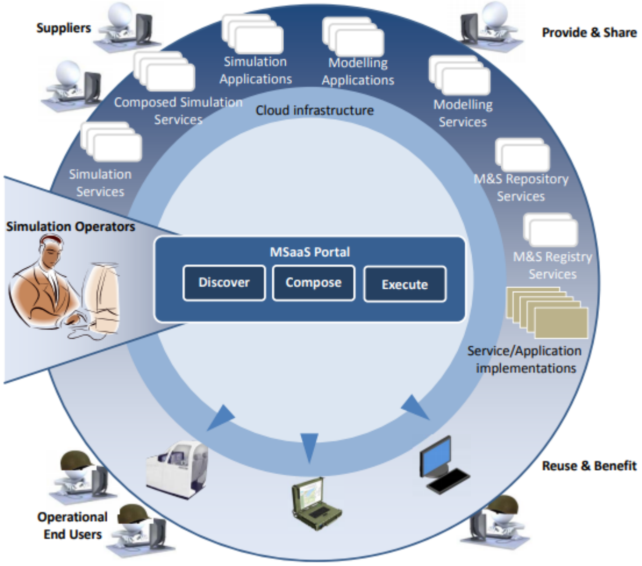


Fig. 1.3 Allied Framework for MSaaS (reproduced from [18])

The NATO MSaaS Technical Reference Architecture is described in the form of architecture building blocks and architecture patterns. An architecture building block defines a capability, capturing amongst other requirements, applicable M&S and data standards, and enabling technology. An architecture pattern suggests ways of combining Architecture Building Blocks. The idea is that the Reference Architecture is not a final product but provides a structure where more content can be added over time. In principle, all capabilities for M&S in the cloud could (eventually) be captured in this Technical Reference Architecture. The NATO MSaaS Engineering Process (EP) is a process description for the development and execution of simulations within an existing MSaaS implementation (or infrastructure). An existing MSaaS implementation is assumed to have M&S Enabling Services in place (such as repository services, composition services, and management and control services) and provides capabilities to create a simulation. The process is described as an overlay to the DSEEP and addresses the MSaaS specific engineering considerations during DSEEP execution. The process description will be updated as the Technical Reference Architecture evolves.

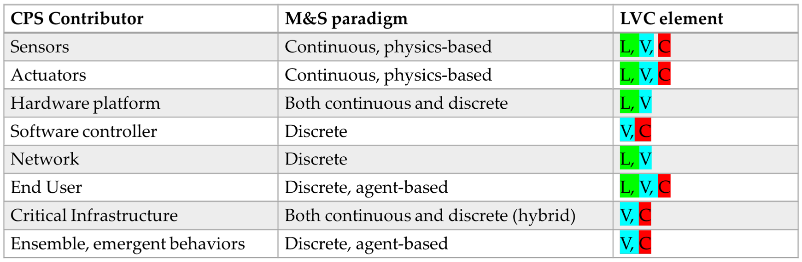
1.3 M&S-based CPS engineering

M&S has been stated as a powerful design technique for CPS [19] but has inherent challenges involved in the design of contemporary CPS. Through the use of M&S, models take the center stage of the entire design process. Every single system specification (and the underlying components) are defined using models. These models are able to show the evolution of the system, and can be used since the very first stages of the design: preparation of concerns, trace generation, impact analysis, verification and validation, simulation, synthesis, etc. Models can be used for early identification of design defects prior to prototyping, significantly reducing costs. Additionally, the use of M&S can facilitate the automation of some key design processes like the synthesis in prototype devices, automatic code generation or the system deployment on complex and heterogeneous platforms [11]. However, current heterogeneity and complexity of CPS can not be easily handled by all the current existing methods and techniques. Currently, it is not possible for a single M&S language or framework to adequately address all the challenges related to CPS [20].

In M&S, we are not only limited to the computational implementations of models. We distinguish between live simulations in which the model involves humans interacting with one another (role playing, play acting, etc.); virtual simulations where the model is simulated by a fusion of humans and computer-generated experiences; and constructive simulations where the model is entirely implemented in a digital computer and may have increased levels of abstraction. Increasingly, we are mixing the three forms of simulation in what is commonly known as live-virtual-constructive (LVC) simulation [21, 22].

From a systems theoretic perspective, a CPS model is a hybrid system made up of both continuous and discrete systems. A continuous system (CS) is one that operates in continuous time and in which input, state and output variables are all real values. A discrete (dynamic) system (DDS) is one that changes its state in a piece-wise constant event-based manner (which also included discrete-time systems as they are a special case of discrete event systems) [23]. A typical example of a hybrid system is a CPS in which the computation subsystem is discrete and a physical system is CS. The LVC environment also qualifies as a CPS, with live systems as CS, constructive systems as DDS and virtual systems as a hybrid (containing both CS and DDS). At the fundamental level, there are various ways to model both timed and untimed discrete event systems, all of which can be transformed to, and studied within, the formal DEVS theory [24, 25, 8, 26, 27].

Figure 1.4 associates each of the CPS constituent elements with the corresponding M&S paradigm and how it can be incorporated in the LVC environment.



CPS Contributor	**M&S paradigm**	**LVC element**
Sensors	Continuous, physics-based	L, V, C
Actuators	Continuous, physics-based	L, V, C
Hardware platform	Both continuous and discrete	L, V
Software controller	Discrete	V, C
Network	Discrete	L, V
End User	Discrete, agent-based	L, V, C
Critical Infrastructure	Both continuous and discrete (hybrid)	V, C
Ensemble, emergent behaviors	Discrete, agent-based	V, C

Fig. 1.4 CPS contributor and the associated M&S paradigm (reproduced from [15])

Using M&S for CPS engineering is not straight forward due to the inherent complexities residing in both the modeling and the simulation activities. A recent panel explored the state of the art of CPS modeling and the complexity associated in engineering intelligence, adaptation and autonomy through M&S. The literature survey conducted in [28] enumerated the following active research areas and the associated technologies for CPS modeling, and concluded that the need for a common formalism that can be applied by practitioners in the field is not yet fulfilled:

- DEVS formalism: Strong mathematical foundation that supports multi-paradigm modeling, multi-perspective modeling and complex adaptive systems modeling to handle emergent behaviors.
- Process algebra: Provides hybrid processes using multi-paradigm modeling. Models combine behavior on a continuous time scale with discrete state transition behavior at given points in time.
- Hybrid automata: Combines finite state machines with Ordinary Differential Equations (ODE) to account for non-deterministic finite states. Bond graphs are used to govern changes.
- Simulation languages: Combines discrete event and continuous system simulation languages. Involves modular design of hybrid languages, multiple abstraction levels combining different formalisms.

- Business processes: Use of standardized notation languages like Business Process Modeling Notation (BPMN) provide value in securing buy-in from the stakeholders in an efficient manner.
- Interface design for co-modeling: Functional Mock-up Interface (FMI) as a means of integration of various CPS components. DEVS can also be used as a common denominator in a vendor neutral manner.
- Model-driven approaches: Model transformation chains to arrive at a single formal model. Governance is required to develop such automation.
- Agent-based modeling: Paradigm to employ component models at scale with individual behaviors, to study ensemble effects.

The above-mentioned approaches and technologies allow the development of CPS models, albeit in a piece-wise manner. These model pieces and their definitions and specifications are dictated by the cross-domain CPS operational use-case. Assuming we now have a validated model (i.e. a model that has been deemed valid by the stakeholders), next comes the task of executing it on a computational platform i.e. simulation. The piece-wise model composition sometime does not directly translate into a monolithic simulation environment due to the confluence of both continuous and discrete elements in the hybrid system.

To design contemporary CPS through cloud-based M&S, two major challenges must be addressed. The first one is the design of a modeling language able to deal with the complexity of current CPS. This chapter has already stated in previous sections that none of the actual modeling languages alone can address all the challenges related to CPS modeling. The second one is to provide modern frameworks to perform CPS engineering, taking into account the distributed nature of CPS that usually integrate cloud computing aspects. In the following, we visit these two main challenges.

1.3.1 The need for a unified M&S process

As stated above, high level languages such as UML, SysML or MARTE have been used for modeling complex CPS. UML has been traditionally used for modeling software systems, and although it defines a syntax of model diagrams, it does not offer semantics for CPS modeling. The OMG SysML standard offers functionalities like requirements management, which is interesting for CPS, but does not provide mechanisms to define aspects of real-time embedded systems such as performance, energy consumption or non-functional constraints. MARTE, another OMG standard for real-time embedded systems, still suffers from not having detailed guidelines and semantics, a problem for its correct utilization. Taking into account Cloud computing modeling aspects, other modeling languages have been used like SoaML [29] or CloudML [30]. However, these approaches are still software-oriented and not able to define the complexity of current CPS.

In the search of a *Universal Modeling Language*, or better said, a *Unified Modeling and Simulation Process*, the initial foundations of the theory of modeling and simulation should be carefully revisited and taken into account. One of the most important theories in the last 50 years was pioneered by Zeigler in [31]. It defines a set of elements and systems specification for model-driven systems engineering. In this regard, DEVS theory was formulated on two orthogonal concepts. It distinguishes between system structure, i.e., how the system is internally constituted, and system behavior, i.e., how the system is externally manifested. This is the foundation of modular systems that have defined input and output interfaces through which all interaction with the environment occurs [8]. This hierarchy of systems specification has 5 levels as shown in Table 1.1, which also related some concepts as defined in the theory of modeling and simulation provided in [25].

Level	Name	System Specification	Elements from the M&S Framework
5	Coupled Systems	Systems built from component systems with a coupling recipe	Model, Simulator, Experimental Frame
4	I/O System Structure	System with state and transitions to generate the behavior	Model, Simulator, Experimental Frame
3	I/O function	Collection of input/output pairs partitioned according to initial state	Model, Source System
2	I/O behavior	Collection of input/output pairs from external black-box view	Model, Source System
1	I/O frame	Input and output variables and ports together with values over a time base	Source System

Table 1.1 Hierarchy of System specifications (adapted from [8]).

Following a standard specification certainly facilitates the definition of a M&S Unified Process. Such process must be based on an *open system* concept. An *open system* is a system that can exchange energy, material and information with the outside world through its reconfigurable interfaces. Current CPS provide a dynamic environment analogous to a variable structure system. This M&S unified process must be able to perform the following aspects:

1. Requirements specification using many Domain Specific Languages (DSLs), like UML, SysML, Business Process Modeling Notation (BPMN), DoDAF, MoDAF, UAF, etc.
2. Platform Independent modeling (PIM) at lower levels of systems specification, using a specific modeling language based on a well-known M&S formalism.
3. Model Structures at higher level of System resolution using the same modeling language.

4. Platform Specific Modeling and execution environment.
5. Automated Test Model generation using PIMs.
6. Cloud-based execution support that allows the deployment of complex M&S scenarios.
7. Interfacing of models with real-time systems where the model becomes the actual systems component.
8. Verification and Validation at every level of system specification and life-cycle development.

The application of M&S in CPS engineering using a unified process must be fully supported by a suite of modeling languages, i.e. domain specific languages that preserve the semantics of the specific domain. This allows engineers to decouple the domain from the model and the model from the simulation framework. This provides many benefits, since models can be constructed independently of the M&S platform, and with DSLs, models can still preserve the domain semantics. A modeling language also facilitates the definition of transformation from DSLs to PIMs.

The DEVS Unified Process (DUNIP) [8] with the underlying DEVS Modeling Language (DEVSML) stack [9] (see Figure 1.5) were designed to fulfill all these requirements, using as a foundation the DEVS application of discrete event dynamical systems theory.

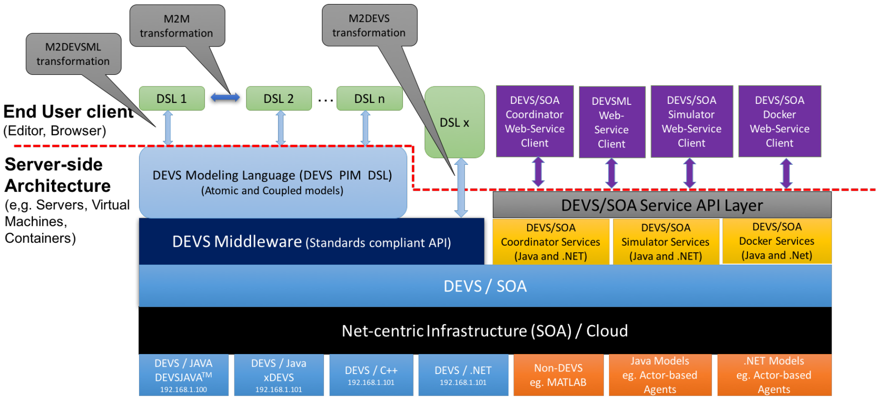


Fig. 1.5 DEVSML Stack (reproduced from [9])

Modeling languages that handle only the software aspects are not positioned to accurately define current CPS. Other languages more oriented to systems engineering have not reached a proper level of M&S maturity to take into account aspects related to the physical processes or distributed nature of contemporary CPS. For CPS M&S, effective semantic alignment is needed to support holistic M&S of complex and heterogeneous CPS. It requires both an M&S Unified Process and an expressive M&S Modeling Language that would yield an unambiguous model. Although there are some approaches at the research level that address these challenges, this is still an open problem that needs to be addressed with the help of an industry standard consortium.

1.3.2 Cloud Implications for CPS engineering

Cloud-based CPS aims to integrate the paradigms of Cloud computing applied to CPS engineering. There are several examples of Cloud-based CPS such as the design of autonomous vehicles, smart homes, automated factories, HW auto-diagnosis and maintenance, smart grid etc. In general, any system that follows the IoT paradigm can be considered a Cloud-based CPS.

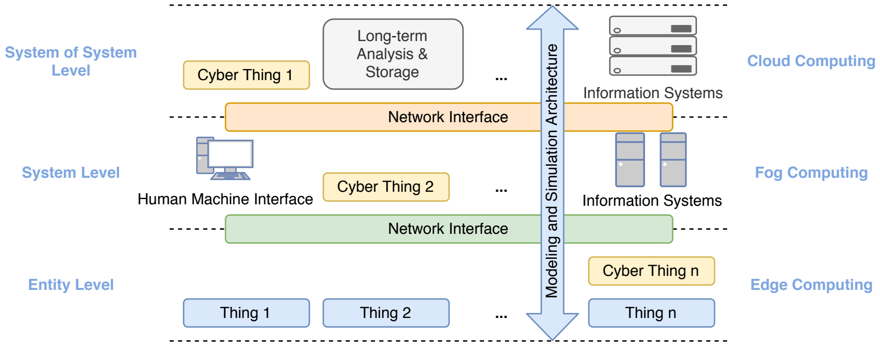


Fig. 1.6 Cloud-based CPS architecture.

As illustrated in Figure 1.6, in a Cloud-based CPS architecture every physical *thing* may have a digital twin representative that can be hosted in the Edge, Cloud or Fog layer. In contemporary Cloud-based CPS, physical and corresponding cyber things do not have to be geographically concentrated. Every thing (physical or cyber) must be identified by a unique ID. Additionally, although not represented in Figure 1.6, physical things are capable of maintaining discrete connections with the cyber things. The edge layer can implement small applications to provide real-time responses. Data are generated by things and consumed by humans through Human Machine Interfaces (HMI), which can also control and provide access to each thing, monitoring and diagnosing the state of the corresponding system. HMI are responsible for maintaining a real-time and collaborative control between the different elements, to keep the coordination of the physical world and the cyber world. Data can be processed in fog or cloud layers using information systems. The allocation of resources (cyber things, HMI, information systems, etc.) will depend on the different requirements of the whole infrastructure in terms of energy consumption, latency, performance and accuracy. A cloud-based M&S

architecture can facilitate the design and deployment of cloud-based CPS, acting as a substrate that is applicable across the entire Cloud-based CPS engineering process. However, this M&S architecture define several technical challenges, enumerated below.

First, it must be a distributed architecture in nature. This aspect brings other key challenges [32]:

1. Application-driven, scalable simulations of large, complex networks.
2. Exploitation of heterogeneous machine architectures.
3. Making parallel and distributed simulation broadly accessible through simpler model development and cloud computing platforms.
4. Online decision making using real-time distributed simulation.
5. Energy- and power-efficient parallel and distributed simulation.
6. Rapid composition of distributed simulations.

Second, the M&S architecture must incorporate co-simulation aspects related to CPS [33]. As in the Unified Modeling Process, here the proposed architecture must be able to combine the strengths of different CPS simulation environments, combining them into a co-simulation framework. These allow system engineers a true multidisciplinary M&S. This demands high levels of interoperability, hardware- and software- in-the-loop, automatic code generation and synthesis, etc.

Third, it must include verification and validation processes, with explicit methods for testing, validation and verification of legacy systems. As a result, these methods will have to be extended to address the scalability needs of Cloud-based CPS. It includes risk, trust, security analysis, etc.

Fourth, the M&S framework must provide support for virtualization, integrating virtualized CPS resources of the whole Cloud-based CPS architecture (both things or management resources). This can provide virtualization as software services so other real or virtual components can use them in co-simulation for data analysis, monitoring and controlling tasks, etc.

Finally, the problem of reliability, scalability and energy efficiency must be addressed by the M&S architecture at several levels [20]: (1) at application-algorithm level; (2) at technology level; (3) at circuit-level, avoiding worst case design; and (4) at system level, using energy-efficient accelerators with build-in trade off Quality-of-Service vs. energy and minimum required subsystems.

1.3.2.1 Current efforts

Under the framework of industry or research projects or software tools, there have been some attempts to develop M&S methodologies to manage Cloud-based CPS engineering. In the MODAClouds project [34] the engineering team can model, develop, deploy, operate, monitor and control cloud applications exploiting the benefits of working with multiple clouds, and fulfilling that the cloud infrastructure and services will always meet some user-defined business requirements. The INTO-CPS project [35] has created an integrated *tool chain* for comprehensive Model-Based Design of CPS. The toolchain supports multidisciplinary, collaborative modeling of CPS from requirements, through design, down to realization in hardware and software. This enables traceability at all stages of the development. Mittal and Risco-Martín [9] integrated Docker with the granular Service Oriented Architecture Microservices paradigm and advanced the state-of-the-art in model and simulation interoperability in Cloud-based CPS (Figure 1.5). They described the architecture incorporating DevOps methodologies using containerization technologies to develop cloud-based distributed simulation farm for Cloud-based CPS specified using the DEVS formalism. Another framework called Simulation, Experimentation, Analytics and Test (SEAT) framework by The MITRE Corporation employs the docker technology and Continuous Integration/Continuous Delivery (CI/CD) pipelines to address integration and interoperability challenges inherent in the CPS M&S engineering [36]. While the DEVSML stack is focused more on transforming various models into a DEVS specification, SEAT focuses on the black box approach of bringing user-apps packaged in docker containers and providing them with a data model to interoperate with other docker containers. The simulation environment (only the constructive in LVC) is deployed as another container app. The DEVSML stack can be subsumed in the SEAT framework as it is docker compliant as well. Figure 1.7 shows the SEAT layered architecture framework.

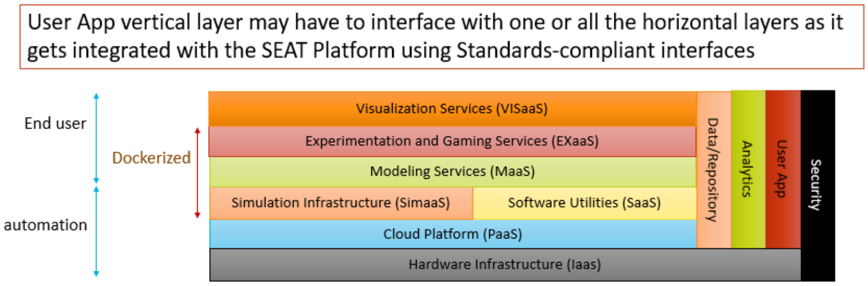


Fig. 1.7 SEAT Layered Architecture Framework (reproduced from [36])

These recent developments bring together cloud technologies, co-simulation methodologies, verification, validation, virtualization and hybrid modeling approaches to deliver an M&S substrate that is applicable across the entire CPS landscape. Performing Cloud-based CPS engineering through the support of an integrative M&S framework, along with a solid M&S process, will facilitate both the conception, design and 5C production of such complex systems.

1.4 Book Overview

This book brings together cloud simulation technologies and their application to CPS engineering. It is divided into four parts: Foundations, Methodology, Applications and Reliability Issues.

The first part of the book, Part I - Foundations, puts together the foundational concepts in composability, distributed simulation, service oriented systems engineering and cloud computing. The second chapter by Andreas Tolk connects various fields of research to support solving composability challenges for effective CPS applications in the domain of cloud, edge and fog computing. The third chapter by Saurabh Mittal and Doug Flournoy details an overview of various mechanisms for maintaining a consistent truth representation in distributed M&S systems. It also presents an architecture to implement mobile propertied agents in a cloud environment for a more robust CPS test and evaluation framework. The fourth chapter by Robert Siegfried provides an overview of the Allied Framework for M&S as a Service (MSaaS) for NATO and allied nations to demonstrate that MSaaS is capable of realizing the vision that M&S products, data and processes are conveniently accessible to a large number of users in a cloud environment. The next chapter by Bo Hu Li et al. introduces a Cyber-Physical System Engineering Oriented Intelligent High Performance Simulation Cloud (CPSEO-IHPSC). CPSEO-IHPSC provides support to access services related to intelligent high performance simulation resources, capabilities and CPS products on demand.

The second part of the book, Part II - Methodology, brings together different methodologies to perform service composition, scheduling and the integration of nature-inspired modeling in Cloud-based CPS engineering. Chapter 6 by Lin Zhang et al. gives a literature review on Cloud-based simulation and proposes a service network-based method to implement service composition and scheduling in simulation. Chapter 7 by Tuncer Oren revises the definition of CPS from the point of view of the evolution of physical tools. It also elaborates on nature-inspired modeling and computing for simulation-based CPS engineering. Chapter 8 by Col. Rob Kewley provides a roadmap for simulation engineers and systems engineering integrators who would like to employ the DEVS Distributed Modeling Framework in their work in order to ease integration and to improve performance in the cloud. Chapter 9 by Daniel Dubois presents methods and an algorithm for simulation of discrete spacetime partial differential equations in classical physics and relativistic quantum mechanics. The development of simulation-based CPS indeed evolves to quantum computing and the chapter presents computing tools that are well adapted to these future requirements of quantum computing.

The third part of the book, Part III - Applications, introduces a few realworld applications of cloud-based CPS engineering. The tenth chapter by Thomas Bitterman shows the design and implementation of a system that implements the simulation as a service model, based on the software as a service model. This system extends the software as a service principle to include high-performance computing hosted applications. Chapter 11 by Kevin Henares et al. presents an automated Cloud-based CPS engineering process for a robust migraine prediction system that allows the generation of alarms before the appearance of new pain episodes. Chapter 12 by Mayank Singh and Hemant Gupta reviews the history of the battery, how a virtual battery works, its application and security considerations in a CPS. The last chapter in this section by Matthew T. McMahon et al. details the development of the MITRE Elastic Goal-Directed simulation framework (MEG), designed to provide modelers and analysts with access to (1) cloud-enabled high-performance computing support, (2) a wide range of design of experiments methods and (3) robust data processing and visualization.

The last part of the book, Part IV - Reliability issues, analyzes different aspects of reliability, truth, resilience and ethical requirements of cloud-based CPS engineering. Chapter 14 by Md Ariful Haque et al. describes a cloud-based simulation platform for deriving cyber resilience metrics for a CPS. Chapter 15 by Sanja Lazarova-Molnar and Nader Mohamed gives a holistic overview of the reliability analysis of CPS. This chapter also identifies the impact that data and new data infrastructures may have on a CPS. Chapter 16 by Margaret Loper defines trust issues, including reliability, with CPS from a multi-dimensional perspective. It describes a set of research projects conducted that span the multiple dimensions of trust. The last chapter by Tuncer Oren describes several dimensions of reliability for CPS: (1) categories or reliability issues, (2) reliability and security aspects of computation, (3) reliability and failure avoidance in simulation and (4) aspects of sources of errors.

1.5 Summary

CPS are complex systems with hardware, software and networking components. The physical hardware components can be accessed and controlled by software over a local network or a geographical network as big as the Internet. Employing M&S for doing hardware-software co-engineering is an already solved problem with established methodologies for engineering closed-loop systems. The Internet era and the distributed networked aspect in CPS have introduced vulnerabilities both at IT and OT levels, which have made CPS engineering a challenging task. To further add to complexity, CPS are multi-modal and multi-domain systems. There do not exist standard practices to perform multi-domain systems engineering. At best, software and systems engineering communities have developed common *lingua franca* and notations such as in UML, SysML or SoAML, but they do not span the entire landscape when it comes to application of these notations to hard engineering disciplines such as Electrical engineering.

M&S for complex systems engineering have developed approaches to build closed loop systems using formal systems modeling concepts founded on mathematical Set theory. This affords validation and verification rigor in M&S solutions. However, the multi-domain nature of CPS requires model transformation chains to be built from domain specific models to a common reference model aligned with the mathematical foundation. While hybrid modeling addresses the model interoperability and semantic alignment, the simulation integration is addressed through the emerging co-simulation approaches.

Cloud-based M&S brings together the latest in Cloud and HPC computing for their use for M&S purposes. Incorporating cloud technologies for M&S solutions requires explicit cloud-systems engineering and the know-how to leverage new technologies for remote management, execution and access. The emerging containter technology, DevOps processes, CI/CD pipelines, remote deployment and management offer automation and ease of use when done correctly. Community involvement at SISO CBMS SG and NATO MSG give evidence of the importance of MSaaS for next generation M&S solutions.

A new generation of architectures are in the making that brings together formal M&S, cloud technologies, hybrid modeling techniques and co-simulation approaches for distributed CPS LVC solutions. This book compiles the state-of-the-art in the next generation cloud-based CPS M&S architectures that aid CPS engineering. We encourage you to continue your journey in the chapters ahead as you explore new vistas in M&S for CPS engineering in the cloud context.

Acknowledgements The author's affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author(s). Approved for Public Release. Distribution Unlimited. Case Number 19-1916-7.

This work has been partially supported by the Education and Research Council of the Community of Madrid (Spain), under research grant P2018/TCS-4423.

References

1. W. Wolf, "News Briefs," *IEEE Computer*, vol. 40, no. 11, pp. 104–105, 2007.
2. S. Mittal and A. Tolk, *Complexity Challenges in Cyber Physical Systems: Using Modeling and Simulation (M&S) to Support Intelligence, Adaptation and Autonomy*, ch. The Complexity in Application of Modeling and Simulation for Cyber Physical Systems Engineering. John Wiley & Sons, 2019.
3. S. Mittal and J. L. Risco-Martín, *Guide to Simulation-Based Disciplines: Advancing Our Computational Future*, ch. Simulation-based Complex Adaptive Systems, pp. 127–151. 2017.
4. J. Fitzgerald and K. Pierce, *Co-modelling and Co-simulation in Embedded Systems Design*, ch. Collaborative Design for Embedded Systems: Co-modelling and Co-simulation, pp. 15–25. Springer, 2014.
5. G. Booch, *The Unified Modeling Language User Guide.* Pearson Education, 2005.

6. S. Friedenthal, A. Moore, and R. Steiner, *A Practical guide to SysML: The Systems Modeling Language.* Morgan Kaufmann, 2014.
7. S. Mittal and J. L. Risco-Martín, *Netcentric System of Systems Engineering with DEVS Unified Process.* CRC Press, 2013.
8. S. Mittal and J. L. Risco-Martín, "Model-driven systems engineering for netcentric system of systems with DEVS Unified Process," in *Proceedings of the 2013 Winter Simulation Conference (WSC 2013)*, pp. 1140–1151, 2013.
9. S. Mittal and J. L. Risco-Martín, "DEVSML 3.0 Stack: Rapid deployment of DEVS farm in distributed cloud environment using microservices and containers," in *Proceedings of the 2017 Spring Simulation Multionference (SpringSim 2017)*, pp. 19:1–19:12, 2017.
10. K. M. Alam and A. El Saddik, "C2PS: A Digital Twin Architecture Reference Model for the Cloud-Based Cyber-Physical Systems," *IEEE Access*, 2017.
11. K. Henares, J. Pagán, J. L. Ayala, M. Zapater, and J. L. Risco-Martín, *Complexity Challenges in Cyber Physical Systems: Using Modeling and Simulation M&S to Support Intelligence, Adaptation and Autonomy*, ch. Cyber-Physical Systems Design Methodology for the Prediction of Symptomatic Events in Chronic Diseases. 2019.
12. J. Lee, B. Bagheri, and H.-A. Kao, "A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, pp. 18 – 23, 2015.
13. C. Sanders, "Research into cloud-based simulation: A literature review (2019-siw-031)," in *Simulation Innovation Workshop*, SISO, 2019.
14. B. P. Zeigler and L. Zhang, *Concepts and Methodologies for Modeling and Simulation*, ch. Service-Oriented Model Engineering and Simulation of System of Systems Engineering. Springer, 2015.
15. S. Mittal and A. Tolk, eds., *Complexity Challenges in Cyber Physical System: Using Modeling and Simulation M&S to Support Intelligence, Adapation and Autonomy.* Wiley & Sons, 2019.
16. SISO, "Cloud based modeling and simulation (cbms) study group (sg)." https://www.sisostds.org/StandardsActivities/StudyGroups/CBMSSG.aspx, 2018.
17. J. Truong, J. Wallace, S. Mittal, and R. Kewley, *Final Report for the Cloud-Based Modeling and Simulation Study Group (CBMS SG) SISO-REF-nnn-DRAFT.* Simulation Interoperability Standards Organization, in review., 2019.
18. J. E. Hannay and T. Berg, "The NATO MSG-136 Reference Architecture for M&S as a Service," in *Proc. NATO Modelling and Simulation Group Symp. on M&S Technologies and Standards for Enabling Alliance Interoperability and Pervasive M&S Applications (STO-MP-MSG-149)*, NATO Science and Technology Organization, 2017.
19. J. C. Jensen, D. H. Chang, and E. A. Lee, "A model-based design methodology for cyber-physical systems," in *2011 7th International Wireless Communications and Mobile Computing Conference*, pp. 1666–1671, 2011.
20. I. Quadri, A. Bagnato, E. Brosse, and A. Sadovykh, "Modeling methodologies for cyber-physical systems: Research field study on inherent and future challenges," *ADA USER*, vol. 36, no. 4, p. 246, 2015.
21. D. Hodson and R. Hill, "The art and science of Live, Virtual, and Constructive simulation for test and analysis," *Journal of Defense Modeling and Simulation*, vol. 11, no. 2, pp. 77–89, 2014.
22. S. Mittal, S. Diallo, and A. Tolk, *Emergent Behavior in Complex Systems Engineering: A Modeling and Simulation Approach.* Wiley & Sons, 4 2018.
23. K. Lee, J. Hong, and T. Kim, "System of Systems Approach to Formal Modeling of CPS for Simulation-based Analysis," *ETRI*, vol. 37, 2015.
24. H. Vangheluwe, "DEVS as a common denominator for multi-formalism hybrid systems modelling," in *IEEE International Symposium on Computer-Aided Control System Design*, pp. 129–134, 2000.

25. B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of Modeling and Simulation. Integrating Discrete Event and Continuous Complex Dynamic Systems.* Academic Press, 2 ed., 2000.
26. S. Mittal and J. L. R. Martin, “DEVSML Studio: A Framework for integrating domain-specific languages for discrete and continuous hybrid systems into DEVS-based M&S environment,” in *Proceedings of Summer Computer Simulation Conference*, 2016.
27. M. K. Traoré, G. Zacharewicz, R. Duboz, and B. Zeigler, “Modeling and simulation framework for value-based healthcare systems,” *SIMULATION*, 2018.
28. A. Tolk, E. Page, and S. Mittal, “Hybrid simulation for cyber physical systems: State of the art and literature review,” in *Proceedings of the Annual Simulation Symposium in Spring Simulation Conference*, 2018.
29. O. M. Group, “Service oriented architecture Modeling Language (SoaML).” http://www.omg.org/spec/SoaML/, 2011.
30. SINTEF, “CloudML.” http://cloudml.org, 2014.
31. B. P. Zeigler, *Theory of Modeling and Simulation.* John Wiley & Sons, 1976.
32. R. M. Fujimoto, “Research challenges in parallel and distributed simulation,” *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 26, no. 4, pp. 1–29, 2016.
33. S. Mittal and B. P. Zeigler, “The practice of modeling and simulation in cyber environments,” in *The Profession of Modeling and Simulation*, Wiley & Sons, 2017.
34. “MODA Clouds Project.” http://multiclouddevops.com, 2020.
35. “Integrated Tool Chain for Model-based Design of Cyber-Physical Systems.” http://into-cps.au.dk, 2020.
36. S. Mittal, N. Kasdaglis, L. Harrell, R. Wittman, J. Gibson, and D. Rocca, “Autonomous and composable m&s system of systems with the Simulation, Experimentation, Analytics and Testing (SEAT) Framework,” in *Proceedings of Spring Simulation Multi-Conference*, 2020.