Challenges in the Operation and Design of Intelligent Cyber-Physical Systems

Sebastian Castro Pieter J. Mosterman Akshay H. Rajhans
Roberto G. Valenti
MathWorks
1 Lakeside Campus Drive, Natick, MA 01760
`{scastro, pmosterm, arajhans, rvalenti}@mathworks.com`

September 15, 2019

Abstract

As technology advances fuel a rapid increase in power and decrease in cost of computation, cyber-physical systems are becoming more ubiquitous in real-world applications such as manufacturing and consumer electronics. With increasingly sophisticated computational functionality implemented as software, these systems are rapidly becoming intelligent and autonomous. Advances in low latency and high throughput reliable communication are enabling such autonomous systems to be networked and interconnected for collaborative operation. The new and evolving capabilities of autonomous and connected engineering systems mean that these intelligent cyber-physical systems are becoming more complex and therefore more challenging to design, test, and operate. This chapter introduces a classification of intelligent cyber-physical systems along two key axes: the *behavior* dimension launched by the advances in *computation*, which corresponds to the degree of autonomy in the systems, and the *configuration* dimension catalyzed by the advances in *communication*, which refers to the scale of the system in terms of the number of connected elements that comprise it. Based on this classification, the chapter explores practical challenges that may arise throughout the design-test-operation life cycle of such systems and outlines the various ways in which computational models can provide value in addressing those challenges.

1 Introduction

Cyber-physical systems (CPS) are computer-controlled physical systems that deploy computational—or *cyber*—elements to sense, control, and operate in a physical environment. Given the rapid strides made by technology advances in computation and communication, these systems are becoming smart and interconnected. Advances in computation have enabled the introduction of artificial intelligence into such systems—sometimes called *intelligent cyber-physical systems* [M¨17] or simply *intelligent physical systems* [KKL15]—that are not just able to sense, understand, and manipulate the physical environment around them, but can also learn and improve over time. Advances in communication have made distributed architectures of intelligent physical systems possible to the extent that individual autonomous systems can operate together in a collaborative fashion.

Modern society is expected to experience a transformational impact of intelligent CPS in a range of sectors. Compelling examples can be found in the utilities, transportation, and manufacturing sectors.

- In the utilities sector, safe and reliable power grids are CPS applications that exemplify *smart energy*, which enables the integration of renewable energy sources that may be intermittent, such as wind energy, by exploiting weather forecasting.

- In the transportation sector, CPS applications are instances of *smart mobility* and include: on land, connected autonomous vehicles that reduce fatalities and optimize for congestion; in water, vehicles that optimize surface shipping and improve underwater exploration; and in the air, unmanned aerial vehicles (often referred to as drones) that improve and enable complex search and rescue operations.

- In the manufacturing sector the *Industry 4.0* paradigm increases safety, reliability, and throughput of industrial production as part of *smart manufacturing*[1]. This also relates to the *digital twin* notion that uses computational simulation of (oftentimes expensive) physical assets on the one hand to predict and detect faults and on the other hand in advanced cases to replace physical functionality with simulated behavior.

Motivated by the global societal-scale relevance of CPS, this chapter intends to outline important design, test, and operation challenges for development and deployment of such systems. Faced with the breadth of a topic of such vast magnitude, systems are classified based on levels of increasingly sophisticated capabilities along dimensions that compare with levels of cognition and communication in humans.

The developed classification provides structure to the chapter content, which is organized as follows. Section 2 introduces an example from the smart mobility domain to motivate the overall discussion. Section 3 draws parallels between the evolution of physical and cognitive faculties in humans and engineered systems as a basis for a feature classification presented in Section 4. With focus on the communicating and collaborating classes, challenges in operating intelligent CPS are outlined in Section 5 followed by challenges for the design and testing of intelligent CPS in Section 6. Section 7 summarizes and concludes the discussion.

2 Connected Autonomous Vehicles

To motivate the discussion in this chapter, let us consider the case of connected autonomous vehicles as an exemplar application of intelligent CPS. Human driving is inefficient and error-prone. By one estimate, traffic congestion cost the U.S. economy \$87 Billion in lost productivity in just the year 2018.[2] As per the National Highway Traffic Safety Administration (NHTSA), more than 37,000 lives were lost

---

[1] http://arminstitute.org/

[2] https://www.cnbc.com/2019/02/11/americas-87-billion-traffic-jam-ranks-boston-and-dc-as-worst-
html

in driving accidents in the U.S. in the year 2017 alone.[3] It is widely believed that the introduction of connectivity and autonomy in automobiles will improve driving safety and efficiency, for example by sensing real-time information about the immediate surroundings to make safer driving decisions, and by connecting to and leveraging aggregated historical traffic congestion information to make smarter routing decisions respectively. A key to achieving these objectives is the aspirational goal of developing engineered systems that are able to navigate traffic situations and driving scenarios like and better than humans.

In order to improve the situational and self awareness of the individual vehicles, automobiles of today and tomorrow must be instrumented by sensors of various modalities, such as cameras, RADAR (radio detection and ranging) and LIDAR (light detection and ranging) elements, global positioning systems (GPS), and inertial measurement units (IMUs). Based on such rich sensory input, advances in software algorithms—such as computer vision algorithms for camera data—are necessary for classifying the raw sensor data and assigning semantic meaning to it, such as road, obstacles, and other agents in the environment. Availability of large amounts of prerecorded data is necessary for offline training of deep neural networks that are then deployed on hardware for real-time online semantic inference. Semantic inferences from various sensory inputs must be combined via sensor fusion algorithms to eliminate shortcomings and potential blind spots of any one sensing mechanism, thereby increasing the reliability.

Processed data must be leveraged in order to build accurate maps of the environment where the vehicle can localize itself and perform planning to safely and effectively navigate traversable areas. Hierarchical planning algorithms must break down an overall goal of reaching a desired destination into a series of steps, such as merges and lane changes, that are further achieved by low-level control tasks such as trajectory optimization and path planning. The extent to which these tasks can be performed by an automobile without the help of a human determine the various *Levels of Driving Automation* identified by the Society of Automotive Engineers (SAE) J3016™ standard.[4]

Connectivity between a network of vehicles and between vehicles and the infrastructure is opening up cooperative and collaborative operation as a new mechanism of operation. Dedicated Short-Range Communication (DSRC) protocols such as those specified in the IEEE 802.11p standard[5] are necessary for real-time sharing of information in vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication modalities [RMB+15]. Raw data, semantic information, and even learning outcomes can now be shared with a fleet of vehicles, so if one car learns a better way to drive, the entire fleet benefits from it.[6]

Smart instrumented intersections such as the variants of Cooperative Intersection Collision Avoidance Systems (CICAS):

- the Stop-Sign Assist (SSA) variant when crossing a high-speed through traffic on rural highways at dangerous stop-sign-controlled intersections [BM13], and

- the Signalized Left-Turn Assist (SLTA) variant when making a left turn in

---

[3] `https://www.nhtsa.gov/`

[4] `https://www.sae.org/standards/content/j3016_201806/`

[5] `https://standards.ieee.org/standard/802_11p-2010.html`

[6] `https://www.recode.net/2016/9/12/12889358/tesla-autopilot-data-fleet-learning`

front of an oncoming vehicle [Mis10],

have been studied as mechanisms to aid human drivers in making better split-second decisions. Even in these infrastructure-based Advanced Driver Assistance Systems (ADAS) incarnations that merely make suggestions to human drivers, there are important heterogeneity, architecture, and verification challenges [RBL+11, RK12, RK13, Raj13, RBR+14].

With increasing autonomy and connectivity, as the SSA functionality transfers from roadside to in-vehicle [BMCD12] and the SLTA functionality is being replaced by connected vehicles that communicate and carry out on-the-fly collision avoidance policies [ZCLM18], the burden of decision making is increasingly being transferred from humans to the connected autonomous vehicles. In process, what once were CICAS suggestions are now morphing into actions with actual consequences. Attaining correct behavior is even more challenging, yet at the same time of paramount importance from a safety point of view.

From an efficiency point of view, meta-level information such as city-level traffic patterns can be used to devise a globally optimal traffic routing strategy to optimize fuel efficiency and traffic throughput, thereby avoiding the so-called *price of anarchy* [ZPCP18] caused by following modern-day routing strategies that are individually optimal but socially suboptimal.

These exemplar challenges and opportunities can be extrapolated to other application domains in CPS. Indeed the need for sensing, perception, decision making, planning, and control and execution to effectively operate in an environment is universal across various intelligent CPS domains. Collaboration between ensembles of smart agents and humans can also be seen across other CPS application domains such as search and rescue [MEB+14, MSB+14, ZMP+15].

3 The Evolution of Physical and Cognitive Faculties in Humans

In order to assess challenges for building intelligent CPS applications that tackle complex challenges such as the ones encountered in connected autonomous driving, it is insightful to consider how humans evolved as intelligent beings who carry out complex tasks in a collaborative society. While a comparison between humans and engineered systems has been made in the literature for specific application domains, such as smart manufacturing [DF16], security and resilience [AE12], and multi-objective Pareto optimization and tradeoff analysis [Kel18], the objective of this chapter is to create a feature classification for intelligent CPS. Specifically, parallels are drawn between the evolution of physical and cognitive faculties of humans with that of engineered systems to build such a feature classification.

3.1 Energy efficiency and physical manipulation

In the early stages of evolution when *Homo erectus* emerged, great advances were made in physical capabilities in order to adapt to the environment. Bipedalism freed up hands and arms while the opposing thumb and padded fingers created far improved manipulation abilities. The ankle and elasticity in tendons enabled energy efficiency that supported exceptional endurance. As additional improvement in the manipulation faculty, *Homo sapiens* developed a shoulder joint that is distinctly different from other species [RVRL13]. This shoulder joint specifically supports the unique ability to throw projectiles with great precision.

3.2 Cognition

Multi-modal sensory processing helped create rich information about the environment. Cognitive abilities to orient in large spaces and plan actions enabled effective use of environment, flora, and fauna. Moreover, the cognitive ability to imitate conspecifics helped share techniques for tool creation and form a culture of tool use that was at the genesis of evolution into modern human [Mac05].

While evidence of tool use shows incremental advances over the course of evolution up until *Homo sapiens*, processing abilities changed significantly [Hol13, Mac05]. Brain size reached a modern volume but perhaps more important are changes in brain architecture to support the cognitive ability that enabled taking the perspective of conspecifics and considering them actors as oneself. Additionally, episodic memory developed that supported the storage of rich sensory information and allowed elaborate planning.

3.3 Language and communication

The next step in cognitive processing abilities enabled the use of symbols, a unique quality of *Homo sapiens* [Hol81]. Symbols as syntactic reference may have developed along with speech, which was enabled first by the anatomical components necessary for rich vocalization. Combined with phonological composition, this provides a practically infinite vocabulary to associate symbols with concepts and notions. The ability to handle elaborate plans supports grammar that requires keeping track of notions expressed in a sentence while keeping track of multiple perspectives not just of different actors but also in future, in past, and at locations other than the current.

With the ability to communicate in a rich language of symbols, larger social structures became possible, which, in turn, supported a much improved culture of knowledge sharing and retention.

3.4 From Natural to Technical

The human faculties that evolved as energy efficiency and manipulation, cognition, and communication in a symbolic language can be grafted onto the three foundational pillars of CPS: control, computation, and communication [KK12]. Control enables efficient energy use in and sophisticated manipulation of the physical world. Computation enables machine intelligence to perform tasks that would have traditionally required human cognition. Communication of data between machines achieves the same tasks as communication in terms of symbols between humans.

Control as manipulation of physical objects and quantities in an energy efficient manner has come to leverage increasing computational power and communication availability. As such, control can be considered functionality layered on top of a compute and communication stack. Hence, the classification of CPS ensembles that follows considers two dimensions. The first dimension considers advances in computation as they relate to advances in cognition from the perspective of control

Type	**Capability**
Reflective	Evaluation and assessment behavior
Reasoned	Deliberate, planned behavior
Reactive	Learned, adaptive behavior
Reflexive	Instinctive behavior, reflexes

Table 1: Behavior at various levels of cognition.

functionality. The second dimension extends the stages of cognition with corresponding communication capability and identifies classes of control functionality that the communication enables.

4 A Classification of Intelligent Cyber-Physical Systems

Given the long history of technology, this chapter presents a structure of technical systems that have emerged as part of the digital revolution. In particular, the cognitive processing faculty of humans is used as a structuring mechanism with communication as a key technology superimposed. The structure enables an outlook as to where technology might move next.

4.1 A Classification of Engineered Systems

A classification of engineered systems is developed based on the increasing cognitive abilities in organisms. Table 1 lists behavior types that involve an increasing level of cognition. The least demanding is behavior of a *reflexive* nature. Examples of this are the dilation of a pupil based on light and the beating of a heart. Behavior that is of a *reactive* nature is learned and adaptable behavior that can be practiced such as reactively kicking a ball to hit a target. Behavior that is *reasoned* involves planning actions to achieve a goal, for example, navigating traffic to enter or exit a highway. Finally, behavior of a *reflective* nature relies on awareness of the organism and allows setting goals based on an assessment of benefit, for example, adopting a healthy lifestyle.

In addition to the behavior of individuals that require different levels of cognitive ability, key in the evolution of *Homo erectus* and later *Homo sapiens* is their ability to communicate with conspecifics [Mac05, Tat14]. While primitive forms of communication are based on direct reference, the ability to assume the perspective of conspecifics was fundamental to more sophisticated communication such as in the form of mimesis (including chant, dance, repeated gesturing, partonomy). This form of communication provided *Homo erectus* with a powerful advantage over contemporary species and allowed spreading all throughout Africa and onto other continents. If the power of communication is of such terrific magnitude, the value of language based on symbolism and endowed with sophisticated concepts such as deixis, as practiced by *Homo sapiens*, is difficult to overstate.

In the technical world, communication technology is quickly becoming the hallmark of engineered systems. Table 2 shows a classification of engineered systems based on the increasingly sophisticated levels of behavior of individual systems with

Configuration	Behavior			
	Reflexive	Reactive	Reasoned	Reflective
Individual	Automatic	Adaptive	Autonomous	Aware
Ensemble	Distributed	Connected	Collaborative	Coallied

Table 2: A classification of behavior in operation.

communication among systems in an ensemble superimposed. The top row characterizes the behavior of individual engineered systems:

- Reflexive behavior is preprogrammed and directly responds to stimuli from the environment. This compares with automatic control architectures where measurements of physical quantities are input to a fixed control law that directly determines an actuation response.
- Reactive behavior is learned, for example by practice or training, and requires an interpretation of the stimuli from the environment. This compares with adaptive systems where different control setpoints or parameters are used based on observations (e.g., an intelligent thermostat may learn preferred temperatures based on historical choices by users).
- Reasoned behavior plans how to reach a goal, which corresponds to a fundamental component of autonomous behavior. Developing plans relies on models of the system itself as well as the observed entities in its surrounding that affect the planning.
- Reflective behavior sets goals while evaluating advantages and drawbacks. This requires an engineered system to be aware of itself, its ambitions, its proclivities, etc., and corresponds to behavior not currently known as generally available in engineered systems.

Turning to ensembles of engineered systems, the corollary of automatic control is *distributed control*, where multiple automatic control loops rely on sharing measurement data among them. In the vein of the intelligent intersection example of Section 2, traffic lights that are synchronized are an example of control that is distributed across a number of individual control loops. Input to such a distributed system may be the arrival of vehicles as detected by an inductive loop in the road surface.

A *centralized* intelligent intersection around infrastructure-to-vehicle (I2V) communication is an example of a *connected control* system in that the coordinator (as part of the infrastructure) creates a situational awareness and then adjusts ensemble setpoints. For example, the coordinator interprets (or it is communicated) whether a vehicle is a luxury car or a tractor trailer combination. Speed setpoints are then computed according to the respective dynamics of the vehicles. These setpoints are communicated to the different vehicles that then adjust in response. Waze[7] is a smartphone app that allows users to share traffic data such as collisions that might have happened at certain geolocations. In addition, Waze analyzes smartphone data, for example, and interprets the data to determine traffic congestion. If vehicles are

---

[7] https://www.waze.com

enabled to automatically respond to such data, an adaptive Waze vehicle is another example of *connected control*. For example, when traffic congestion is detected and shared with vehicles elsewhere, these connected vehicles may adapt the route to their destination.

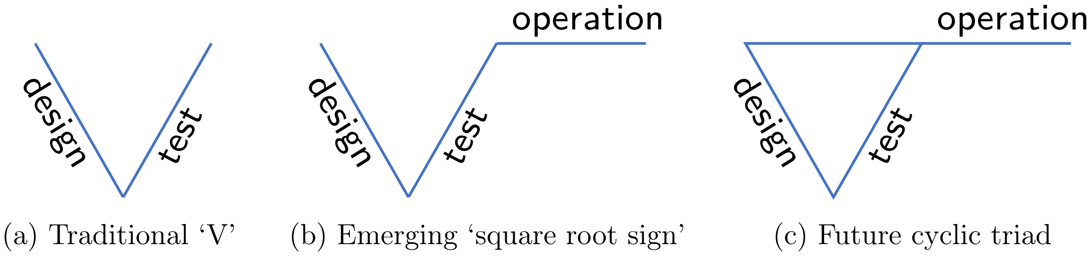
A *decentralized* intelligent intersection (V2V) is an example of *collaborating control* where autonomous systems form an ensemble and share rich information to support overall planning behavior such as individual plans and planning considerations. In contrast with the I2V intelligent intersection, vehicles may not rely on predefined control laws at that intersection with setpoints determined by the coordinator. Instead the different vehicles act autonomously and as they approach an intersection they must reason together about their speed, in which lane to drive, distance to intersection, vehicle dynamics, possible urgency, etc. to come up with an overall plan for the ensemble. This decentralized control orchestrates the movement of each vehicle to best achieve the individual plans of all vehicles using the intersection at that given time period. Returning to Waze, another example of collaborating systems would be a Waze that involves the various vehicles adapting their planning given new information. This could involve changing the order of an overall plan, modifying times of arrival or departure, speed, or deciding on different stops along the way to minimize vehicles contending for the same road segments at certain times. Yet another example of collaborative systems is a smart emergency response system [MEB+14, MSB+14, ZMP+15], where autonomous ground vehicles may plan how to arrive at select locations where they set up a depot. Autonomous rotorcraft may then plan sorties to deliver provisions in the field while minimizing overall time of delivery.

Aware systems in a *coallied control* ensemble would share information about internal states such as intent or assessments, though no examples of such operational ensembles are known.

4.2 Mapping the Classification onto Challenges

Traditional Model-Based Design has been studied following a ‘V’-shaped approach (e.g., [MPE04]) as drawn in Figure 1a, where models are used to design and test the implementation that is deployed. However, intelligent CPS require a paradigm shift in this traditional viewpoint. With new applications such as Industry 4.0 and industrial internet of things (IIoT), modeling and simulation are also being used in operation after systems are deployed. For example, a computational model—a so-called ‘digital twin’—of an expensive physical asset is simulated during operation for prognostics and predictive maintenance purposes. This emerging paradigm can be thought of as a ‘square root sign’ rather than a ‘V’, as drawn in Figure 1b. In the more sophisticated intelligent CPS that constitute dynamically reconfigurable ensembles, the operation facet connects back to design and test for runtime reconfiguration. This reconfigurability morphs the ‘square root’ into a cyclic triad as is depicted in Figure 1c.

It is worth noting that this triad shares some similarities with the CPS trustworthiness framework developed by the U.S. National Institute for Standards and Technology (NIST), which considers conceptualization, realization, and assurance as three sides in a triangular (prism) representation (Figure 8 in [Gro17]). Though the terms conceptualization, realization, and assurance are related to design, operation,



(a) Traditional 'V' (b) Emerging 'square root sign' (c) Future cyclic triad

Figure 1: Paradigm shift from the traditional Model-Based Design 'V' to a Model-Based Design-Test-Operation triad.

and test, the NIST framework considers models in a limited context as the output of the conceptualization phase and an input to the realization phase. In contrast, the view of this chapter holds that modeling and simulation play an equally important role in all three sides of the triad.

The two-dimensional representation from Figure 1c can be projected in three dimensions by adding the various increasingly sophisticated behavior classes from Table 2 as the third dimension. This forms a prism in three dimensions, as is depicted in Figure 2 with the behavior classes in the ensemble configuration—distributed, connected, collaborative, and coallied—annotated next to each triad.

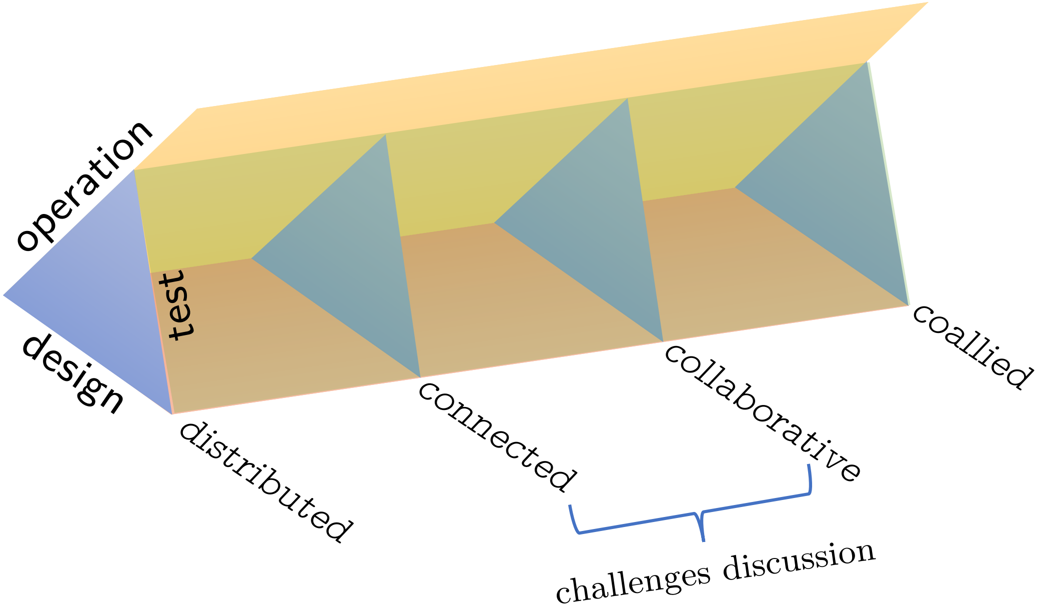


Figure 2: Intelligent CPS Ensemble Design-Test-Operation Prism

There are several challenges associated with the development of intelligent CPS from each of these classes. This chapter focuses the discussion around challenges for systems that are open by communicating with other systems in ensembles. The study of the ensemble dimension is further narrowed by concentrating on systems that are currently under most scrutiny: (i) connected systems and (ii) collaborative systems. The remaining classes in the ensemble dimension are either well understood (distributed control systems) or they require significant research to conceive of and develop technology that is still very much in its infancy (coalitions of systems). Challenges for these systems are discussed in the context of (i) machines being connected, (ii) machines being collaborative, (iii) design of such connected and collaborative ensembles of machines, and (iv) testing such ensembles at runtime as they come online. The challenges are first discussed from the perspective of systems in operation in Section 5 and then the design and test perspective in Section 6. Related work [MZ16a] details a corresponding needs analysis.

5 Challenges in System Operation

During operation, communication among intelligent CPS in an ensemble opens up opportunities for new functionality, even more so when sophisticated collaboration is supported. Future enabling technology that is needed, the current state, and challenges are discussed for connected and collaborative operation.

5.1 Connected Operation

Systems that are part of ensembles where connections with other systems are established during operation have a need for wireless communication, data sharing, and service utilization.

Wireless Communication. High performance wireless communication will allow reliable configuration of flexible system configurations for features with varying quality of service. There are two key challenges.

- The communication protocol stack must be physically aware and configurable while being compatible with the internet protocol (IP), for example, IEEE 802.15.4e [PAV$^+$13]. Such a protocol stack supports real-time services of graded quality with a low energy footprint and enables including (precise) time and location information in communicated data. Useful approaches are the modeling of building blocks that comprise communication protocols as well as modeling of the performance characteristics across electronics hardware targets.

- Precise timing and synchronization (e.g., the precision time protocol IEEE 1588 [CEP07]) must be supported in a distributed and wirelessly connected environment. Two strands of advances that are important are: (i) physical layer based timing and synchronization architectures, which benefit from modeling of the physical radio frequency (RF) layer as well as the antenna and (ii) scheduling of periodic and aperiodic events with reliable execution times (e.g,, [ZGL10]), which builds on advances in scheduler configuration, dynamic scheduling with guarantees, and support for mixed synchronous and asynchronous behavior.

Data Sharing. Advanced data sharing will allow distributed information resources to be effectively exploited and enable system features that were not considered *a priori* at the sharing sources. Where in an offline approach system integration would be responsible for synchronization of data streams [Mul07], in online connectivity scenarios this must be resolved by construction. There are two specific challenges to address.

- The functionality must provide support for multirate architectures where the methodologies concern the synchronization of data from incongruent sources.

Solution aspects to consider include communication modeling, systematic (and automatic) analysis of double buffering schemes, timing properties of software, and clock recovery. An increasing use of models for system integration is imperative [MGF05].

- To derive value for the system features it must be possible to reliably extract corresponding (unambiguous) information from the communicated data. Approaches in support of this represent information as high-level models with well-defined metamodels and ontologies with model import/export under version control, automatic generation of metamodels (e.g., from model libraries), and sharing and comparing of model concepts. The RoboEarth network is an example of concrete advances in that internet connections enable robots to generate, share, and reuse data [WBD+11].

Service Utilization. Systems that are dynamically assembled post deployment will be endowed with the capability to purpose available functionality in service of specific (singular) needs. Three challenges are listed.

- Service-based approaches must operate as real-time embedded services in a physical environment. Advances stem from work on real-time middleware and service-oriented architectures with physical capabilities that must address key technical issues such as service discovery response time (latency, averages, time-out) [HJS07] and request for services in different modalities. Middleware that is real-time capable ranges from a real-time version of the High Level Architecture (HLA), and a real-time Common Object Request Broker Architecture (CORBA), to the Data Distribution Service (DDS) and the Robot Operating System (ROS) [KGS+04, MFF04, PG14, QCG+09, SCH08].

- Service discovery must increase in logic capabilities (be ‘smart’). Possible solutions may be the use of service ontologies for service provider matching that rely on taxonomies for similarity and transformability matching (e.g., [SCM10]). Capabilities for type similarity checking and conversion as well as semantics definition are key.

- Information sharing must be enabled in heterogeneous system ensembles. A language and ontology infrastructure, for example, the Ontology Web Language (OWL) [HPS11] used for describing semantic web services, may serve as an underlying technology to support translation and transformation. Additional technology to build on includes the reliable generation of models and (implementation) code from models.

5.2 Collaborative Operation

Three needs for enabling system ensembles to operate in a collaborative manner include runtime system adaptation, emerging behavior design, and functionality sharing.

Runtime System Adaptation. The ability for safe and reliable system adaptation at runtime enables a system in an ensemble to exploit functionality that is exogenous, implemented by other systems, for efficient, economical, and resilient operation.

- The main challenge is reasoning and planning adaptation of an ensemble of systems, which builds on a number of technologies: (i) introspection of the systems in an ensemble to determine the system state, configuration (possibly using runtime variants), and available services (possibly based on a middleware service description specification); (ii) handling of ensemble (in)consistency with a level of fidelity that is sufficient for runtime needs, which may build on traceability between representations, possibly across transformations (e.g., models@run.time [BFCA14]); and (iii) online (re)calibration of models (e.g., [HSV10]) to continuously ensure accuracy, which may use collected data, along with statistical, optimization, or machine learning tools, to modify the parameters or structure of software artifacts within the system.

Emerging Behavior Design. Robust methods to design emerging behavior allow for the systematic design of systems that are part of an ensemble such that the ensemble as a whole realizes desired behavior in an optimal sense.

- The overall challenge is about collaborative planning, guidance, and control with a number of methods to build on: (i) analysis methods across loosely coupled architectures are key, especially for embedded operation (e.g., globally asynchronous/locally synchronous, GALS, architectures [MWO$^+$05]), which spans event-driven control, discrete-event modeling and analysis, and uncertainty modeling; (ii) planning and synthesis of distributed control functionality on concurrent resources is potentially core and involves concurrency and platform modeling, functionality decomposition, and service composition (e.g., the Towers of Hanoi as a CPS) [MZ15, MZH13]; and (iii) formal methods to ensure conformance and be applicable to collaborative problems with concurrency semantics while enabling property proving with performance models, all by retaining the rigor of formality yet in an accessible manner (e.g., design refinement in Conway's Game of Life [SS12]).

- In addition, behavior can be learned [Hof17]. By tracking the performance of collaborative systems based on the decisions made by the system—for example, allocating tasks, resources, or operating modes—techniques such as online optimization and reinforcement learning [KBP13] can use this historical data to automatically refine the strategy and rules of a complex decision-making system, thus giving rise to novel behavior patterns that may better solve an existing environment or even adapt to an evolving environment.

Functionality Sharing. Sharing functionality among systems in an ensemble, not only by making the functionality available but also by meaningful exchange of information and meta information about the functionality, will allow the creation of novel system features post deployment. There are two challenges to highlight.

- Using functionality for multiple (different) purposes post deployment, which may build on a number of advances: (i) generation of models for a particular task by property identification (e.g., property based model slicing) and model behavior selection (e.g., behavioral analysis and functionality mining such as determining the requirements for a design from its behavior); (ii) hardware resource sharing by creating a dispatch architecture (e.g., a real-time virtual machine [GZ12]) and following a platform-based design approach [BDD$^+$09, Nat12]; (iii) performance characterization (e.g., [LZ05]) via performance models and measures (e.g., critical path analysis and code performance reporting and advise); and (iv) online calibration (e.g., [LT13]) based on objective and performance criteria, which is supported by adaptive filtering, distortion modeling, and automatic groundtruthing (baselining).

- Interaction between features (e.g., [MZ16b]) that leverage the shared functionality to find potential (re)solutions in assumption formalization and dependency effect analysis. Specific technologies include model slicing based on properties or assumptions, tracing between sources and destinations for behavioral anomalies, and mapping formulated assumptions about functionality to behavior.

6 Challenges in System Design and Test

The operational opportunities for ensembles of intelligent CPS build on corresponding advances in their design and test. A key aspect is the cyclic nature of the design, test, and operation stages as illustrated in Fig. 1a. Future technology that is needed is compared against the current state with challenges to arrive at the future state.

6.1 Design

In the design of connected and collaborating ensembles of systems, two main needs are the ability for virtual system integration and for design artifact sharing.

Virtual System Integration. The ability to realistically integrate systems in a virtual sense will enable the confident design of systems as part of a reliable system ensemble that configures during operation. There are three challenges to elaborate on.

- Obtaining proper models in design, which potentially builds on the generation of models with necessary detail given a selected property of interest. This can involve structural model changes, operating point selection and linearization, implementation model generation, and so forth. A number of approaches to highlight include selecting model detail based on properties of interest [FS95, MB00, SL95], counterexample guided refinement [CGJ$^+$00], and requirements guided abstraction selection [JMM14].

- System-level design and analysis of a heterogeneous ensemble by using models requires advances in (i) connecting, combining, and integrating models represented in different formalisms, potentially at the behavior level via cosimulation or a shared simulation API or at a shared semantics level by code generation and (ii) efficient simulation models that can be used across dynamics and execution semantics, which involves an array of potentially interacting solver configurations for continuous-time, discrete-time, and discrete-event behaviors. These challenges are a subject of study in the field of computer automated multiparadigm modeling [MV00] where combining dynamic semantics [MB02] and execution semantics [Mos07] are important advances.

- Connectivity among models, software, and hardware corresponding to different vendors and end manufacturers can build on efforts in creating open tool platforms with trusted interfaces for communication across synchronized and coordinated models, software, and hardware devices. Some underlying technologies include data streaming, target connectivity support, standardized communication protocols (e.g., TCP, UDP), and real-time simulation [PM12].

Design artifact sharing. Being able to securely and reliably share design artifacts across design efforts for a system ensemble will allow convenient, efficient, and consistent collaboration between stakeholders in design and ultimately throughout the system life cycle. Two challenges are presented.

- Given the different organizations (vendors, end manufacturers, and others) that are invariably involved, tool coupling among these disparate organizations may be addressed by building on (e.g., Open Services for Lifecycle Collaboration, OSLC [SS13]): (i) support for traceability across semantic and technology adaptation, for example, based on a service API with change notification to establish relations across abstractions, formalisms, and transformations, all while honoring intellectual property protection, and (ii) information extraction from protected intellectual property (e.g., by obfuscation or encryption) and use of trusted compilers.

- Supporting manifold views and tools that are essential in design, especially for system ensembles can advance based on core technologies such as (i) configurable view projections (e.g., [ASB10]) that are tool specific and support model generation, pattern extraction or slicing, and XML interexchange, and (ii) use of consistent semantics across tools by modeling the execution engines (e.g., [MZ11]) that are combinations of code libraries (e.g., numerical integration, root finding, and algebraic equation solving) with a broad spectrum of optimization so that semantic analysis of an execution engine as a dynamic system is enabled.

6.2 Test

To test behavior of system ensembles that rely on interfaces for runtime configuration, it is key to develop support for runtime system adaption, collaborative functionality testing, and hardware resources sharing.

Testing Runtime Adaptable Systems. Testing of systems in a runtime adaptable configuration will allow confidently exploiting functionality across the overall ensemble, as was mentioned in the context of collaborative operation. A challenge specific to testing is discussed here.

- Testing complex functionality on a deployed system is critical, yet challenging for systems that operate embedded in a physical environment. Often, a full model of the system's interaction with the environment is not available or impractical to implement given the available computational resources. *Surrogate models* [ACVGH10] can provide an approximation of the actual process that is accurate enough and computationally feasible for online testing and design optimization procedures. Specific technologies include variable and dimensionality reduction through methods such as sensitivity analysis and principal component analysis, using data to generate response surfaces, and fitting low-fidelity models ranging from polynomials to artificial neural networks.

Collaborative functionality testing. The ability to test collaborative functionality will enable assurance of collaboration quality on shared resources while being able to identify and automatically mitigate root causes of failure in a distributed environment. Two challenges are expounded on.

- Systematic test suite generation and automated test evaluation is especially challenging for collaborative functionality. Solution approaches possibly build on model-based test generation from requirements while preserving the context of a dynamic ensemble configuration. Specific technologies (e.g., [ASE14, Zan09]) include coverage-based automatic test generation, variants-based testing, and closed-loop testing.

- Reproducible test results under minimum uncertainty could leverage two key advances: (i) setting of initial conditions and injecting fault data, especially when using a service architecture, with specific technologies such as system state restoration, stateless services, and test fixture generation (e.g., [LSD+13]), and (ii) temporal and spatial partitioning to isolate functionality for a specific system architecture under investigation, which includes time partition testing and functionality extraction.

Hardware Resource Sharing. Robust, safe, and secure support for hardware resource sharing will allow contracting out system resources within an ensemble and and will support a balanced use of external resources for resiliency and runtime cost optimization. Two challenges are discussed.

- Determination of key test cases for different implementations builds on characterization of computational architectures, for example, by use of static analysis methods (such as abstract interpretation), automatic test generation, and detailed models of hardware architecture behavior implementations (e.g., to know of 'corner cases').

- Safety of heterogeneous system ensembles is critical and requires advances in: (i) modeling the semantics of time (e.g., [ZMHD11]) to support safety monitoring components (e.g., watchdogs and mitigators [KLM03, KKS+10]), which includes time represented as harmonic periods (e.g., integer time), synchronous behavior (e.g., a single clock or discrete time), simultaneous behavior (iterations of change), dense (e.g., a rational time base), or continuous (e.g., variable-step numerical solutions) and (ii) dynamically mixing safety integrity levels (SiL) and the use of certification kits for components of mixed SiL, which supports matching software with hardware.

7 Conclusions

Cyber-physical systems (CPS) are rapidly increasing in complexity owing to the progress in computation and communication technology. As these systems become connected and autonomous, the complexity of their design, test, and operation presents several key challenges. To outline these challenges in a structured manner, a classification scheme is developed along the behavior and configuration dimensions drawing a parallel between human faculties and engineered systems capabilities. Challenges throughout the life cycles of these intelligent systems during design, test, and operation are discussed, with a focus on connected and collaborative behavior in an ensemble configuration. The class of ensembles with distributed control has been subject of much research and is relatively well understood. In contrast, at the opposite end of the classification spectrum, there is no known scholarly content on the class of ensembles that comprise self-aware systems and so create self-aware ensembles.

The use of computational models is key in addressing the various design, test, and operation challenges. While the traditional role of models during the design phase is well understood, models are also needed for successful real-time operation, for example in predictive maintenance and prognostics and health monitoring applications. For connected and collaborative ensembles that may include several instances of dynamic reconfigurations at runtime, several online iterations of design and test would be necessary along with each instance of reconfiguration (along the 'development and operations' or DevOps paradigm, albeit with differences [JAPT16]). Here, models would provide an effective—and perhaps the only—way to carry out these run-time evaluations as the systems themselves are already deployed and in operation.

The goal of the chapter is to provide a broad overview, yet the discussion is by no means exhaustive. Despite the number and scale of the challenges, the potential of intelligent CPS to fundamentally transform human lives is enormous. Continuing to push the frontiers of CPS will require close collaboration between the research and industry communities, hardware and software vendors, and multiple disciplines in science, engineering, technology, and more.

References

[ACVGH10] Felipe A. C. Viana, Christian Gogu, and Raphael Haftka. Making the most out of surrogate models: Tricks of the trade. *Proceedings of the ASME Design Engineering Technical Conference*, 1, 01 2010.

[AE12] M. Azab and M. Eltoweissy. Bio-inspired evolutionary sensory system for cyber-physical system defense. In *2012 IEEE Conference on Technologies for Homeland Security (HST)*, pages 79–86, Nov 2012.

[ASB10] Colin Atkinson, Dietmar Stoll, and Philipp Bostan. Orthographic software modeling: A practical approach to view-based development. In Leszek A. Maciaszek, César González-Pérez, and Stefan Jablonski, editors, *Evaluation of Novel Approaches to Software Engineering*, volume 69 of *Communications in Computer and Information Science*, pages 206–219. Springer Berlin Heidelberg, 2010.

[ASE14] Aitor Arrieta, Goiuria Sagardui, and Leire Etxeberria. A model-based testing methodology for the systematic validation of highly configurable cyber-physical systems. In *The Sixth International Conference on Advances in System Testing and Validation Lifecycle*, pages 66–72, 2014.

[BDD$^+$09] Felice Balarin, Massimiliano DAngelo, Abhijit Davare, Douglas Densmore, Trevor Meyerowitz, Roberto Passerone, Alberto Sangiovanni-Vincentelli Alessandro Pinto, Alena Simalatsar, Yosinori Watanabe, Guang Yang, and Qi Zhu. Platform-based design and frameworks: Metropolis and metro ii. In Gabriela Nicolescu and Pieter J. Mosterman, editors, *Model-Based Design for Embedded Systems*, Computational Analysis, Synthesis, and Design of Dynamic Systems, chapter 10, pages 259–322. CRC Press, Boca Raton, FL, 2009. ISBN: 9781420067842.

[BFCA14] Nelly Bencomo, Robert B. France, Betty H. Cheng, and Uwe Aßmann, editors. *Models@run.time*, volume 8378 of *Lecture Notes in Computer Science (LNCS)*. Springer, Berlin, Germany, 2014.

[BM13] Ensar Becic and Michael Manser. Cooperative intersection collision avoidance system – stop sign assist traffic-based FOT: MNDOT contract number 98691 deliverable for task 4.1 final report. Technical report, Minnesota Department of Transportation, 2013.

[BMCD12] Ensar Becic, Michael P. Manser, Janet I. Creaser, and Max Donath. Intersection crossing assist system: Transition from a road-side to an in-vehicle system. *Transportation Research Part F: Traffic Psychology and Behaviour*, 15(5):544–555, 1 2012.

[CEP07] Todor Cooklev, John C. Eidson, and Afshaneh Pakdaman. An implementation of ieee 1588 over ieee 802.11b for synchronization of wireless local area network nodes. *Instrumentation and Measurement, IEEE Transactions on*, 56(5):1632–1639, Oct 2007.

[CGJ$^+$00] Edmund M. Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith. Counterexample-guided abstraction refinement. In *Computer Aided Verification, 12th International Conference, CAV 2000, Chicago, IL, USA, July 15-19, 2000, Proceedings*, pages 154–169, 2000.

[DF16] João Dias-Ferreira. *Bio-Inspired Self-Organising Architecture for Cyber-Physical Manufacturing Systems.* PhD thesis, The Royal Institute of Technology (KTH), Stockholm, Sweden, 2016.

[FS95] John B. Ferris and Jeffrey L. Stein. Development of proper models of hybrid systems: A bond graph formulation. In François E. Cellier and José J. Granda, editors, *1995 International Conference on Bond Graph Modeling and Simulation (ICBGM '95)*, number 1 in Simulation, pages 43–48, Las Vegas, January 1995. Society for Computer Simulation, Simulation Councils, Inc. Volume 27.

[Gro17] CPS Public Working Group. Framework for cyber-physical systems: Volume 1, overview. Technical report, National Institute of Standards and Technology, jun 2017. NIST Special Publication 1500-201.

[GZ12] Zonghua Gu and Qingling Zhao. A state-of-the-art survey on real-time issues in embedded systems virtualization. *Journal of Software Engineering and Applications*, 5(4):277–290, 2012.

[HJS07] Joe Hoffert, Shanshan Jiang, and Douglas C. Schmidt. A taxonomy of discovery services and gap analysis for ultra-large scale systems. In *Proceedings of the 45th Annual Southeast Regional Conference*, ACM-SE 45, pages 355–361, New York, NY, USA, 2007. ACM.

[Hof17] Ludovic Hofer. *Decision-making algorithms for autonomous robots.* PhD thesis, Université de Bordeaux, 11 2017.

[Hol81] Ralph L. Holloway. Culture, symbols, and human brain evolution: A synthesis. *Dialectical Anthropology*, 5:287–303, 1981.

[Hol13] Ralph L. Holloway. The evolution of the hominid brain. In Winfried Henke and Ian Tattersall, editors, *Handbook of Paleoanthropology*, pages 1–23. Springer, Berlin, Heidelberg, 2013.

[HPS11] Ian Horrocks and Peter F. Patel-Schneider. Knowledge representation and reasoning on the semantic web: Owl. In John Domingue, Dieter Fensel, and James A. Hendler, editors, *Handbook of Semantic Web Technologies*, chapter 9, pages 365–398. Springer, Berlin, Germany, 2011.

[HSV10] Yilin Huang, Mamadou D. Seck, and Alexander Verbraeck. Towards automated model calibration and validation in rail transit simulation. In Peter M. A. Sloot, G. Dick van Albada, and Jack Dongarra, editors, *Proceedings of the International Conference on Computational Science, ICCS*, pages 1253–1259, 2010.

[JAPT16] Ramtin Jabbari, Nauman Ali, Kai Petersen, and Binish Tanveer. What is devops?: A systematic mapping study on definitions and practices. In *Scientific Workshop of XP 2016. Proceedings*, pages 1–11, 05 2016.

[JMM14] Zhihao Jiang, Pieter J. Mosterman, and Rahul Mangharam. Requirement-guided model refinement. Technical Report MLAB-70, University of Pennsylvania, December 2014.

[KBP13] Jens Kober, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[Kel18] K. L. Keller. Leveraging biologically inspired models for cyber–physical systems analysis. *IEEE Systems Journal*, 12(4):3597–3607, Dec 2018.

[KGS$^+$04] Yamuna Krishnamurthy, Chris Gill, Douglas C. Schmidt, Irfan Pyarali, Louis Mgeta, Yuanfang Zhang, and Stephen Torri. The design and implementation of real-time CORBA 2.0: dynamic scheduling in TAO. In *Proceedings of the 10th IEEE Real-time Technology and Application Symposium (RTAS '04)*, pages 121–129, 2004.

[KK12] K. Kim and P. R. Kumar. Cyber–physical systems: A perspective at the centennial. *Proceedings of the IEEE*, 100(Special Centennial Issue):1287–1308, May 2012.

[KKL15] Daniel E. Koditschek, Vijay Kumar, and Daniel D. Lee. Future directions of intelligent physical systems: A workshop on the foundations of intelligent sensing, action and learning (FISAL), 2015. Available at: `https://basicresearch.defense.gov/Portals/61/Documents/future-directions/4_FISAL.pdf`.

[KKS$^+$10] Bernhard Kaiser, Vanessa Klaas, Stefan Schulz, Christian Herbst, and Peter Lascych. Integrating system modelling with safety activities. In *Computer Safety, Reliability, and Security, 29th International Conference, SAFECOMP 2010, Vienna, Austria, September 14-17, 2010. Proceedings*, pages 452–465, 2010.

[KLM03] Bernhard Kaiser, Peter Liggesmeyer, and Oliver Mäckel. A new component concept for fault trees. In *Proceedings of the 8th Australian Workshop on Safety Critical Systems and Software - Volume 33*, SCS '03, pages 37–46, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[LSD$^+$13] Philipp Leitner, Stefan Schulte, Schahram Dustdar, Ingo Pill, Marco Schulz, and Franz Wotawa. The dark side of SOA testing: Towards testing contemporary soas based on criticality metrics. In *5th International ICSE Workshop on Principles of Engineering Service-Oriented Systems, PESOS 2013, May 26, 2013, San Francisco, CA, USA*, pages 45–53, 2013.

[LT13] Jesse Levinson and Sebastian Thrun. Automatic online calibration of cameras and lasers. In *Proceedings of Robotics: Science and Systems*, Berlin, Germany, June 2013.

[LZ05] Jie Liu and Feng Zhao. Towards service-oriented networked embedded computing. Technical Report MSR-TR-2005-28, Microsoft Research, February 2005.

[M1̈7] Hausi A. Müller. The rise of intelligent cyber-physical systems. *Computer*, 50(12):7–9, December 2017.

[Mac05] Brian MacWhinney. Language evolution and human development. In Bruce J. Ellis and David F. Bjorklund, editors, *Origins of the Social*

Mind: Evolutionary Psychology and Child Development, pages 383–410. Guilford Press, New York, 2005.

[MB00] Pieter J. Mosterman and Gautam Biswas. Towards procedures for systematically deriving hybrid models of complex systems. In *Hybrid Systems: Computation and Control, Third International Workshop, HSCC 2000, Pittsburgh, PA, USA, March 23-25, 2000, Proceedings*, pages 324–337, 2000.

[MB02] Pieter J. Mosterman and Gautam Biswas. A hybrid modeling and simulation methodology for dynamic physical systems. *SIMULATION*, 78(1):5–17, 2002.

[MEB$^+$14] Pieter J. Mosterman, David Escobar Sanabria, Enes Bilgin, Kun Zhang, and Justyna Zander. A heterogeneous fleet of vehicles for automated humanitarian missions. *Computing in Science & Engineering*, 12:90–95, August 2014.

[MFF04] Thom McLean, Richard M. Fujimoto, and J. Brad Fitzgibbons. Middleware for real-time distributed simulations. *Concurrency - Practice and Experience*, 16(15):1483–1501, 2004.

[MGF05] Pieter J. Mosterman, Jason Ghidella, and Jon Friedman. Model-based design for system integration. In *Proceedings of The Second CDEN International Conference on Design Education, Innovation, and Practice*, pages CD–ROM: TB–3–1 through TB–3–10, Kananaskis, Alberta, Canada, July 2005.

[Mis10] James A Misener. Cooperative intersection collision avoidance system (cicas): Signalized left turn assist and traffic signal adaptation. Technical report, University of California, Berkeley, 2010.

[Mos07] Pieter J. Mosterman. Hybrid dynamic systems: Modeling and execution. In Paul A. Fishwick, editor, *Handbook of Dynamic System Modeling*, chapter 15, pages 15–1–15–26. CRC Press, Boca Raton, FL, 2007.

[MPE04] Pieter J. Mosterman, Sameer Prabhu, and Tom Erkkinen. An industrial embedded control system design process. In *Proceedings of The Inaugural CDEN Design Conference (CDEN'04)*, Montreal, Canada, July 2004. CD-ROM: 02B6.

[MSB$^+$14] Pieter J. Mosterman, David Escobar Sanabria, Enes Bilgin, Kun Zhang, and Justyna Zander. Automating humanitarian missions with a heterogeneous fleet of vehicles. *Annual Reviews in Control*, 38(2):259–270, 2014.

[Mul07] Gerrit Muller. Coping with system integration challenges in large complex environments. In *The Seventeenth International Symposium of the International Council on Systems Engineering INCOSE*, page paper ID: 7.1.4, Jun 2007.

[MV00] Pieter J. Mosterman and Hans Vangheluwe. Computer automated multi-paradigm modeling in control system design. In *Proceedings of the IEEE International Symposium on Computer-Aided Control System Design*, pages 65–70, Anchorage, Alaska, September 2000.

[MWO$^+$05] Steven P. Miller, Mike W. Whalen, Dan O'Brien, Mats P. Heimdahl, and Anjali Joshi. A methodology for the design and verification of globally asynchronous/locally synchronous architectures. Technical Report NASA/CR-2005-213912, National Aeronautics and Space Administration (NASA), Langley Research Center, September 2005.

[MZ11] Pieter J. Mosterman and Justyna Zander. Advancing model-based design by modeling approximations of computational semantics. In *Proceedings of the 4th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools*, pages 3–7, Zürich, Switzerland, September 2011. keynote paper.

[MZ15] Pieter J. Mosterman and Justyna Zander. GitHub Repository: Towers of Hanoi in MATLAB/Simulink for Industry 4.0. Jan 2015. doi: 10.5281/zenodo.13977.

[MZ16a] Pieter J. Mosterman and Justyna Zander. Cyber-physical systems challenges—a needs analysis for collaborating embedded software systems. *Software and Systems Modeling*, 15(1):5–16, 2016.

[MZ16b] Pieter J. Mosterman and Justyna Zander. Industry 4.0 as a cyber-physical system study. *Software and Systems Modeling*, 15(1):17–29, 2016.

[MZH13] Pieter J. Mosterman, Justyna Zander, and Zhi Han. The towers of hanoi as a cyber-physical system education case study. In *Proceedings of the First Workshop on Cyber-Physical Systems Education*, Philadelphia, PA, April 2013.

[Nat12] Marco Di Natale. Specification and simulation of automotive functionality using AUTOSAR. In Katalin Popovici and Pieter J. Mosterman, editors, *Real-time Simulation Technologies: Principles, Methodologies, and Applications*, Computational Analysis, Synthesis, and Design of Dynamic Systems, chapter 21, pages 523–548. CRC Press, Boca Raton, FL, 2012. ISBN: 9781439846650.

[PAV$^+$13] Maria Rita Palattella, Nicola Accettura, Xavier Vilajosana, Thomas Watteyne, Luigi Alfredo Grieco, Gennaro Boggia, and Mischa Dohler. Standardized protocol stack for the internet of (important) things. *IEEE Communications Surveys & Tutorials*, 15(3):1389–1406, July 2013.

[PG14] Héctor Pérez and J. Javier Gutiérrez. A survey on standards for real-time distribution middleware. *ACM Computing Surveys*, 46(4):49:1–49:39, March 2014.

[PM12] Katalin Popovici and Pieter J. Mosterman, editors. *Real-time Simulation Technologies: Principles, Methodologies, and Applications*. Computational Analysis, Synthesis, and Design of Dynamic Systems. CRC Press, Boca Raton, FL, 2012.

[QCG$^+$09] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

[Raj13] Akshay H. Rajhans. *Multi-model heterogeneous verification of cyber-physical systems*. PhD thesis, Carnegie Mellon University, 2013.

[RBL$^+$11] Akshay Rajhans, Ajinkya Bhave, Sarah Loos, Bruce H. Krogh, André Platzer, and David Garlan. Using parameters in architectural views to support heterogeneous design and verification. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 2705–2710, Dec 2011.

[RBR$^+$14] Akshay Rajhans, Ajinkya Bhave, Ivan Ruchkin, Bruce H. Krogh, David Garlan, André Platzer, and Bradley Schmerl. Supporting heterogeneity in cyber-physical systems architectures. *IEEE Transactions on Automatic Control*, 59(12):3178–3193, Dec 2014.

[RK12] Akshay Rajhans and Bruce H. Krogh. Heterogeneous verification of cyber-physical systems using behavior relations. In *Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '12, pages 35–44, New York, NY, USA, 2012. ACM.

[RK13] Akshay Rajhans and Bruce H. Krogh. Compositional heterogeneous abstraction. In *Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control*, HSCC '13, pages 253–262, New York, NY, USA, 2013. ACM.

[RMB$^+$15] Christian M. Richard, Justin F. Morgan, L. Paige Bacon, Justin S. Graving, Gautam Divekar, and Monica G. Lichty. Multiple sources of safety information from v2v and v2i: Redundancy, decision making, and trust—safety message design report. Technical report, Office of Safety Research and Development, Federal Highway Administration, 2015. Available at `https://www.fhwa.dot.gov/publications/research/safety/15007/15007.pdf`.

[RVRL13] Neil T. Roach, Madhusudhan Venkadesan, Michael J. Rainbow, and Daniel E. Lieberman. Elastic energy storage in the shoulder and the evolution of highspeed throwing in *Homo*. *Nature*, 498(7455):483–486, 2013.

[SCH08] Douglas C. Schmidt, Angelo Corsaro, and Hans Hag. Addressing the challenges of tactical information management in net-centric systems with DDS. CrossTalk special issue on Distributed Software Development, May 2008.

[SCM10] Zhexuan Song, Alvaro A. Cárdenas, and Ryusuke Masuoka. Semantic middleware for the internet of things. In Florian Michahelles and Jin Mitsugi, editors, *IOT*. IEEE, 2010.

[SL95] Jeffrey L. Stein and Loucas S. Louca. A component-based modeling approach for system design: Theory and implementation. In François E. Cellier and José J. Granda, editors, *1995 International Conference on Bond Graph Modeling and Simula tion (ICBGM '95)*, number 1 in Simulation, pages 109–115, Las Vegas, January 1995. Society for Computer Simulation, Simulation Councils, Inc. Volume 27.

[SS12] Jeff W. Sanders and Graeme Smith. Emergence and refinement. *Formal Aspects of Computing*, 24(1):45–65, 2012.

[SS13] Tiberiu Seceleanu and Gaetana Sapienza. A tool integration framework for sustainable embedded systems development. *IEEE Computer*, 46(11):68–71, 2013.

[Tat14] Ian Tattersall. An evolutionary context for the emergence of language. *Language Science*, 46(Part B):199–206, 2014.

[WBD$^+$11] Markus Waibel, Michael Beetz, Raffaello D'Andrea, Rob Janssen, Moritz Tenorth, Javier Civera, Jos Elfring, Dorian Gálvez-López, Kai Häussermann, J.M.M. Montiel, Alexander Perzylo, Björn Schießle, Oliver Zweigle, and René van de Molengraft. RoboEarth - A World Wide Web for Robots. *Robotics & Automation Magazine*, 18(2):69–82, 2011.

[Zan09] Justyna Zander. Model-based testing of real-time embedded systems in the automotive domain. *PhD Thesis, Technical University Berlin*, March 2009.

[ZCLM18] Yue Zhang, Christos G. Cassandras, Wei Li, and Pieter J. Mosterman. A discrete-event and hybrid simulation framework based on simevents for intelligent transportation system analysis. In *14th IFAC Workshop on Discrete Event Systems (WODES)*, pages 323–328, 2018.

[ZGL10] Yuanfang Zhang, Christopher D. Gill, and Chenyang Lu. Configurable middleware for distributed real-time systems with aperiodic and periodic tasks. *IEEE Transactions on Parallel Distributed Systems*, 21(3):393–404, 2010.

[ZMHD11] Justyna Zander, Pieter J. Mosterman, Gréegoire Hamon, and Ben Denckla. On the structure of time in computational semantics of a variable-step solver for hybrid behavior analysis. In *Proceedings of the 18th IFAC World Congress*, Milan, Italy, September 2011.

[ZMP$^+$15] Justyna Zander, Pieter J. Mosterman, Taskin Padir, Yan Wan, and Shengli Fu. Cyber-physical systems can make emergency response smart. *Procedia Engineering*, 107:312–318, 2015.

[ZPCP18] Jing Zhang, Sepideh Pourazarm, Christos G Cassandras, and Ioannis Ch Paschalidis. The price of anarchy in transportation networks: Data-driven evaluation and reduction strategies. *Proceedings of the IEEE*, 106(4):538–553, 2018.