Building ‘Systems That You Can Bet Your Life On’ Again and Again: Challenges and Opportunities for Cyber-Physical System Design Automation

Pierluigi Nuzzo, Department of Electrical Engineering, University of Southern California, Los Angeles, CA
nuzzo@usc.edu

Cyber-physical systems (CPSs) result from the integration of ‘cyber’ components (computation and/or communication as well as control) with physical processes of different nature to perform functions that cannot be achieved by the cyber and physical parts in isolation. CPSs are an integral part of our lives; examples range from cars, aircraft, and robots, to energy-efficient buildings, computer-monitored and controlled factory lines, and wearable medical devices. By gathering information from a multitude of sources, processing it, and applying it while interacting with the physical environment, CPSs are changing the way entire industries operate, and have the potential of radically influencing how we deal with crucial societal problems, including national security and safety, energy management, civil infrastructure, transportation, manufacturing, and healthcare. However, the complexity and heterogeneity of these systems pose significant design challenges. We refer to the literature for extensive analyses on CPS design and its foundations (e.g., see [1-10].) This article offers some reflections on some of the CPS design challenges, what makes CPS design different from large-scale integrated circuit design, and what could be the opportunities for the design automation community.

Cyber-Physical System Design Challenges. CPS designers are expected to simultaneously reason about hybrid, discrete/continuous design spaces, which are often too large to be efficiently explored while providing strong guarantees of correctness, dependability, and compliance with regulations. Requirement-management tools are still predominantly centered on text-based languages, which creates opportunities for ambiguities and potential conflicts [23]. While virtual prototyping and model-based engineering tools are the *de facto* standard for system development, the *concept design* phase largely remains a manual process, the domain of experienced system architects, often relying on their accrued knowledge and a set of heuristic evaluations to take risky decisions. Poorly inter-operable domain-specific languages and tools make it hard to combine the results of different analysis or synthesis methods. Assessing system correctness is then left to lengthy simulations and prototype tests later in the design process, which may yield implementations that are inefficient and sometimes do not even satisfy the requirements. The realization of CPSs as well as the deployment of autonomous systems would dramatically benefit from principled design methodologies and algorithmic techniques to enhance design quality and productivity under strong guarantees of correctness, dependability, and security. In this respect, the contribution of design automation is crucial. It is about building ‘systems that people can bet their lives on’ [11]. It is also about distilling the design principles that allow doing it again and again, in a systematic and reproducible way, to achieve what we look for today, and what we dream for tomorrow.

Electronic design automation (EDA) has indeed been successful in taming the complexity of billions of devices on a chip, turning it into elegant designs that can be modified and evolved, and allow predictably building complex artifacts [12]. How can CPS design automation mirror the success of EDA? In this article, we focus on three challenges that touch upon aspects that are unique to CPSs, and are being tackled by the research community, with the potential of fostering a new generation of methodologies, algorithms, and tools for system design. The challenges concern reasoning about the interaction between discrete and continuous models, devising compositional and hierarchical design methodologies, and dealing with uncertainty.

From EDA to Cyber-Physical System Design Automation. Analysis and design of CPSs increasingly require *methods and tools that can efficiently reason about the interaction between discrete models, e.g., used to describe embedded software components, and continuous models used to describe physical processes.* In this respect, a central difficulty is the very different nature of the tools used to analyze continuous dynamics (e.g., real analysis) and discrete dynamics (e.g., combinatorics.) Advances in formal verification and optimization over the years have led to techniques that can address large-scale problems in either of the two domains. Boolean methods such as satisfiability (SAT) solving have been successful in tackling large combinatorial search problems for the design and verification of hardware and software systems [15]. On the other hand, problems in control, communications, signal processing, data analysis and modeling, and machine learning often rely on mathematical programming (e.g., see [13],) and specifically convex programming [14] as a powerful solution engine. However, despite their strengths, neither SAT solving, nor convex programming would be effective in CPS design, if used in isolation. We need methods and tools that blend concepts from both of them. An approach to formal reasoning about continuous and discrete dynamics is to model the cyber-physical system as a hybrid automaton, including discrete states as well as continuous states governed by differential equations, and use model checking techniques [15] to prove properties about the system or find bugs. Another approach, based on theorem proving, aims at proving properties of hybrid systems by leveraging formulas in appropriate logic languages capturing their behaviors [16]. The scalability of these verification techniques is an active research area, aiming to extend the number of supported state variables, which is currently limited to only a few tens [4]. Progress has also been made toward correct-by-construction synthesis approaches that algorithmically generate design artifacts from high-level logic specifications, despite the theoretically higher computational complexity of synthesis versus verification [4]. An active research effort aims to leverage the efficiency and formal guarantees of state-of-the-art constraint solving algorithms in both the Boolean and convex analysis domains toward a novel framework, termed satisfiability modulo convex programming (SMC) [17,18]. Inspired by the satisfiability modulo theory [15] approach, SMC rethinks the connection between logic-based methods and numerical methods for reasoning about the combination of discrete and continuous dynamics that can address the complexity of CPS applications.

In addition to verification and synthesis methods, devising design methodologies – *well-defined abstraction layers, hierarchical decomposition mechanisms, regular and reproducible architectures* – has been crucial to the success of EDA. Many of these concepts have been formalized within the platform-based design (PBD) methodology, which has been applied to a variety of domains, from automotive to system-on-chip, from building automation to synthetic biology [19]. *Design methodologies are also on the 'critical path' to operational CPSs.* Designers usually decompose a system into different domains, by adopting the most effective mathematical formalisms to represent different portions of the design or different viewpoints (e.g., function, safety, timing, energy) at different abstraction levels. They then leverage the most suitable tools to analyze and synthesize these models and architectures compositionally. CPS design would then benefit from unifying frameworks that can support different analysis techniques, based on formal verification, simulation, and testing, in a consistent way [1,4,7,26]. Further, design methodologies should consider extra-functional viewpoints, including energy and cost, as well as reliability, security, and usability, most of which are difficult to formally quantify today. Active research efforts in the areas of component-based [20] and contract-based [21-23] design address these challenges by relying on mathematical models of the interfaces between the components and their environments, together with rigorous ways of composing and refining these interfaces. A contract captures the assumptions that a component makes on its environment and the behaviors it guarantees in the context of the assumptions. In a contract-based environment, it is then possible to verify system-level properties in a modular and hierarchical fashion, based on the satisfaction of component-level properties [24]. It is possible to support rigorous stepwise refinement to reason about system decompositions, even if the component implementations are not yet available [25,26]. It is possible to facilitate component reuse, as any components satisfying a contract directly inherit its guarantees and can be used in the design. Contracts can provide a formal foundation and complement the PBD methodology to enable design space exploration and provably-correct concurrent development of system architectures and control algorithms in a modular and scalable way [4,26,27].

As for integrated circuits, *CPSs must operate under the presence of uncertainty* that may result from variability (e.g., due to manufacturing tolerances, usage, failures,) noise, or model approximations. However, in CPSs, uncertainty may also result from external conditions that are not under system control: unknown or unpredictable environment conditions, multi-agent dynamics, and even adversarial behaviors. A classic example are CPSs interacting with humans [10]. Researchers are exploring many approaches for modeling and reasoning about uncertainty, including the use of probabilistic models, capable of expressing the likelihood for a certain event to happen. While probabilistic models and probabilistic logics are well-studied for discrete systems, their application to CPSs are active areas of research [28-31].

Fast Forward. Overall, it is expected that CPSs will rely less and less on humans, and replace humans, for example, in jobs that require precision, or for tedious or dangerous tasks. The intelligence, situational awareness, and decision-making capabilities of these systems will increasingly rely on machine learning components. Similarly, the design process itself will increasingly use data-driven techniques [10]. Learning-enabled CPSs will, in turn, introduce additional sources of approximation, and require probabilistic and statistical reasoning [32]. Further, they will raise additional concerns about the security implications of their tasks as well as the privacy of the data they process. Security and privacy will add new dimensions to the design problem and impact all the abstraction layers. Tackling these fundamental and practical challenges will spur exciting research for design automation at the boundary with formal methods, control theory, real-time systems, machine learning, optimization, and physical platforms, toward advancing the scientific foundations that are necessary to build the planetary-scale, inference-and-decision-making systems we dream of. Design embodies humankind's attempt at creating new artifacts, shaping the surrounding world, and augmenting its capabilities. It is, however, on us to ensure that these artifacts operate safely and enhance our lives.

References

[1] J. Sztipanovits et al., "Toward a science of cyber–physical system integration," Proc. IEEE, vol. 100, no. 1, pp. 29–44, Jan. 2012.

[2] E. A. Lee, "Cyber physical systems: Design challenges," in Proc. IEEE Int. Symposium on Object Oriented Real-Time Distributed Computing, pp. 363–369, 2008, doi:10.1109/ISORC.2008.25.

[3] E. A. Lee and S. A. Seshia, Introduction to Embedded Systems: A Cyber-Physical Systems Approach. Cambridge, MA, USA: MIT Press, 2016.

[4] P. Nuzzo, A. L. Sangiovanni-Vincentelli, D. Bresolin, L. Geretti, and T. Villa, "A platform-based design methodology with contracts and related tools for the design of cyber-physical systems," in Proc. IEEE, vol. 103, no. 11, 2015.

[5] R. Alur, Principles of cyber-physical systems. Cambridge, MA, USA: MIT Press, 2015.

[6] A. Sangiovanni-Vincentelli, W. Damm, and R. Passerone, "Taming Dr. Frankenstein: Contract-Based Design for Cyber-Physical Systems," European Journal of Control, vol. 18, no. 3, pp. 217–238, 2012.

[7] J. Sifakis, "System design automation: Challenges and limitations," in Proc. IEEE, vol. 103, no. 11, pp. 2093–2103, Nov. 2015.

[8] Q. Zhu and A. L. Sangiovanni-Vincentelli, "Codesign Methodologies and Tools for Cyber–Physical Systems," in Proc. of the IEEE, vol. 106, no. 9, pp. 1484-1500, Sept. 2018, doi:10.1109/JPROC.2018.2864271

[9] Proceedings of the IEEE, Special Issue on Design Automation of Cyber-Physical Systems, vol. 106, no. 9, Sept. 2018.

[10] S. A. Seshia, S. Hu, W. Li, and Q. Zhu, “Design automation of cyber-physical systems: Challenges, advances, and opportunities,” in IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 36, no. 9, pp. 1421–1434, Sep. 2017.

[11] J. Wing, “Cyber-Physical Systems,” Computing Research News, vol. 20, no. 1, January 2008.

[12] A. Sangiovanni-Vincentelli, “Corsi e ricorsi: The EDA story,” IEEE Solid State Circuits Magazine, vol. 2, no. 3, pp. 6–26, 2010.

[13] A. Bemporad and M. Morari. “Control of systems integrating logic, dynamics, and constraints,” Automatica, vol. 35, no. 3, pp. 407–427, 1999.

[14] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[15] E. M. Clarke, T. A. Henzinger, H. Veith, and R. P. Bloem, Handbook of model checking. Springer, 2018.

[16] A. Platzer, Logical Foundations of Cyber-Physical Systems. Springer, 2018.

[17] Y. Shoukry, P. Nuzzo, A. L. Sangiovanni-Vincentelli, S. A. Seshia, G. J. Pappas and P. Tabuada, "SMC: Satisfiability Modulo Convex Programming," in Proceedings of the IEEE, vol. 106, no. 9, pp. 1655-1679, Sept. 2018. doi: 10.1109/JPROC.2018.2849003

[18] P. Nuzzo, A. Puggelli, S. A. Seshia, and A. L. Sangiovanni-Vincentelli, “CalCS: SMT solving for non-linear convex constraints,” Proc. Formal Methods Computer-Aided Design, Oct. 2010, pp. 71–79.

[19] A. Sangiovanni-Vincentelli, “Quo vadis, SLD? Reasoning about the trends and challenges of system level design,” Proc. IEEE, vol. 95, no. 3, pp. 467–506, Mar. 2007.

[20] L. de Alfaro, T. A. Henzinger, “Interface theories for component-based design.” In: Henzinger, T.A., Kirsch, C.M. (eds.) EMSOFT 2001. LNCS, vol. 2211, pp. 148– 165. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45449-7 11

[21] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, P. Reinkemeier, A. Sangiovanni-Vincentelli, W. Damm, T. A. Henzinger, and K. G. Larsen. “Contracts for system design.” Foundations and Trends in Electronic Design Automation 12, no. 2-3 (2018): 124-400.

[22] P. Nuzzo, A. L. Sangiovanni-Vincentelli (2018) Hierarchical System Design with Vertical Contracts. In: Lohstroh M., Derler P., Sirjani M. (eds) Principles of Modeling. Lecture Notes in Computer Science, vol. 10760. Springer, Cham.

[23] P. Nuzzo, M. Lora, Y. Feldman, and A. Sangiovanni-Vincentelli, “CHASE: Contract-Based Requirement Engineering for Cyber-Physical System Design,” Proc. Design, Automation and Test in Europe, Dresden, Germany, pp. 839–844, Mar. 2018.

[24] A. Cimatti and S. Tonetta, “Contracts-refinement proof system for component-based embedded systems,” Science of Computer Programming 97, Part 3 (2015), pp. 333 – 348.

[25] I. Filippidis and R. M. Murray, "Layering Assume-Guarantee Contracts for Hierarchical System Design," in Proceedings of the IEEE, vol. 106, no. 9, pp. 1616-1654, Sept. 2018. doi: 10.1109/JPROC.2018.2834926

[26] P. Nuzzo, H. Xu, N. Ozay, J. B. Finn, A. L. Sangiovanni-Vincentelli, R. M. Murray, A. Donzé, S. A. Seshia, "A Contract-Based Methodology for Aircraft Electric Power System Design," IEEE Access, vol. 2, pp. 1-25, Jan. 2014.

[27] D. Kirov, P. Nuzzo, R. Passerone, A. L. Sangiovanni-Vincentelli, “ArchEx: An Extensible Framework for the Exploration of Cyber-Physical System Architectures,” Proc. Design Automation Conference (DAC), June 2017.

[28] M. Kwiatkowska, G. Norman, and D. Parker. “Stochastic Model Checking,” In Formal Methods for the Design of Computer, Communication and Software Systems: Performance Evaluation (Lecture Notes in Computer Science), M. Bernardo and J. Hillston (Eds.), Vol. 4486. Springer, 220–270, 2017.

[29] J. Li, P. Nuzzo, A. Sangiovanni-Vincentelli, Y. Xi, and D. Li, “Stochastic Contracts for Cyber-Physical System Design Under Probabilistic Requirements,” Proc. Int. Conf. Formal Methods and Models for Co-Design, pp. 5–14, Oct. 2017.

[30] A. Platzer, “Stochastic differential dynamic logic for stochastic hybrid programs” In Int. Conf. Automated Deduction, pp. 446–460, 2011.

[31] D. Sadigh and A. Kapoor, “Safe Control under Uncertainty with Probabilistic Signal Temporal Logic,” Proc. of Robotics: Science and Systems, 2016.

[32] D. Amodei et al., “Concrete problems in AI safety,” ArXiv e-prints, Jun. 2016. [Online]. Available: https://arxiv.org/abs/1606.06565