Volume 1, Issue 1, February 02, 2016

TC-CCPS Newsletter

IEEE Technical Committee on Cybernetic for CPS

Founding Editorial

The continuous scaling of integrated circuit (IC) technologies has been driving the semiconductor industry for several decades. Nowadays, even though the scaling of mainstream CMOS technologies is starting to slow down, the field of VLSI circuits and systems continues its remarkable growth with numerous opportunities along two complementary avenues: (1) development of post-silicon devices, circuits and systems (e.g., carbon nanotube, graphene, memristor, etc.), and (2) discovery of emerging application domains (e.g., biomedical electronics, internet of things, etc.). The VLSI Circuits and Systems Letter, published twice a year, aims to report recent advances in VLSI technology, education and opportunities and, consequently, grow the research and education activities in the area.

This letter is affiliated with the Technical Committee on VLSI (TCVLSI) under the IEEE Computer Society. TCVLSI covers the design methodologies for advanced VLSI circuit and systems, including digital circuits and systems, analog and radio-frequency circuits, as well as mixed-signal circuits and systems. The emphasis of TCVLSI falls on integrating the design, computer-aided design, fabrication, application, and business aspects of VLSI while encompassing both hardware and software.

Xin Li
TC-CCPS Editor
Carnegie Mellon University

Intelligent Machine Condition Monitoring for Cyber-Physical Systems

Arkadeb Ghosal, Kaushik Ravindran, Patricia Derler, Hugo A. Andrade, and Jeannie Falcon
National Instruments Corporation

1 Introduction

Modern day Industrial Internet of Things (IIoT) applications are large heterogeneous distributed systems with over 30,000 sensors and 10,000 nodes. Trends indicate a tremendous growth in the number of connected components over the coming years. Gartner Research predicts over 20 billion interconnected devices by 2020, representing a $3 trillion business and technology opportunity [1]. Lee et al. refer to this emerging global cyber-physical network as the *TerraSwarm*, encompassing trillions of sensors and actuators deployed across the planet [2]. These applications will dynamically assemble sensors and computation nodes, aggregate and process large quantities of data, and transfer decisions to actuators and controllers, while meeting tight performance requirements and cost constraints.

Cyber-physical networked systems can generate gigabytes and potentially terabytes of sensor data about the condition and operation of the system. For example, the condition monitoring solution for the Victoria Line of the London Underground rail system yields 32 TB of data every day [3]. In the midst of this explosion of engineering and measurement data, it has become imperative for systems to incorporate a sound management strategy to aggregate the data, conduct diagnostic analytics about the condition of the system, and facilitate predictive maintenance to reduce downtimes and maximize efficiency. Given the cost and complexity of modern cyber-physical systems, it is important that the monitoring solution be scalable and customizable to meet changing application requirements.

Nevertheless, with the advances in sensing and networking technologies, adding measurements to systems has become easier and cost-effective. Intelligence of data acquisition devices and sensors has drastically increased and become more decentralized, with processing elements moving closer to the sensor. In addition to measurement devices getting smarter, smart sensors have emerged that integrate sensing, signal conditioning, embedded processing, and digital interfacing into the sensor node itself.

As processing moves closer to the sensor, innovation in measurement system software is required to efficiently push analytics to the edge. Future software for edge-based systems will be able to quickly configure and manage thousands of networked measurement devices and push a myriad of analytics and signal processing to those nodes. Going forward, systems must transition to smarter, software-based measurement nodes to keep up with the amount of analog data and derive insights about patterns and trends in the operation of the system. The “smart edge” needs specialized software and platform solutions to perform local control and data acquisition and interconnect with entire networks of intelligent “systems of systems” [3].

In this paper, we discuss advances in intelligent machine condition monitoring for cyber-physical networked systems and recent technologies in this area. Section 2 of this paper reviews key components and techniques in a machine condition monitoring solution. Section 3 then presents an industrial tool called InsightCM from National Instruments and discusses an application case study.

2 Machine Condition Monitoring

Machine condition monitoring (MCM) is the process of monitoring the condition of a machine with the intent to predict mechanical wear and failure. Vibration, noise, and temperature measurements are often used as key indicators of the state of the machine. Trends in the data provide health information about the machine and help detect machine faults early, which prevents unexpected failure and costly repair [4]. The need to eliminate catastrophic downtimes due to unexpected breakdowns and unnecessary maintenance costs has made condition monitoring critical for asset utilization and productivity across diverse industries. The global machine condition monitoring equipment market is expected to grow at a CAGR of 7.6% between 2015 and 2020 from $1.5 billion in 2014 to $2.5 billion by 2020 [5].

MCM provides vital information about the health of a machine. An organization can use this information to detect warning signs early and avoid unscheduled outages, optimize machine performance, and reduce repair time and maintenance costs. Figure 1 shows a typical machine failure example and the warning signs. The user can detect failure signs months before repair is required, allowing for proper maintenance scheduling and shutdown.

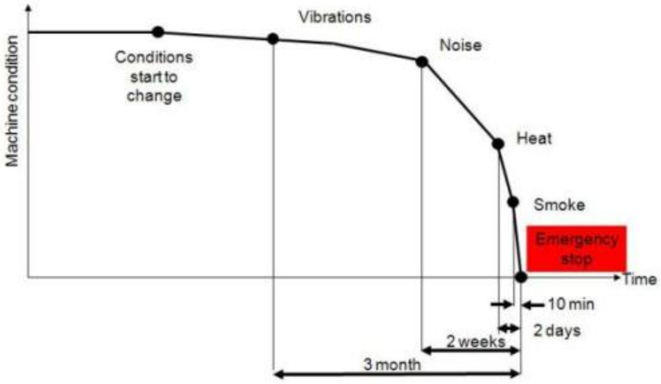

Figure 1: The warning signs of machine failure

As shown in Figure 1, vibrations are the first warning sign that a machine is prone to failure. This warning sign can provide up to three months of lead time before the actual failure date. Monitoring this data with vibration analysis hardware and software helps predict this failure early and schedule proper maintenance [6].

There are five main types of machine condition monitoring. *Route-based monitoring* involves a technician recording data intermittently with a handheld instrument. This data is then used to determine if more advanced analysis is needed. *Portable machine diagnostics* is the process of using portable equipment to monitor the health of machinery. Sensors are typically permanently attached to a machine and portable data acquisition equipment is used to read the data. *Factory assurance test* is used to verify that a finished product meets its design specifications and to determine possible failure modes of the device. *Online machine monitoring* is the process of monitoring equipment as it runs. Data is acquired by an embedded device and transmitted to a main server for data analysis and maintenance scheduling. *Online machine protection* is the process of actively monitoring equipment as it runs. Data is acquired and analyzed by an embedded device. Limit settings can then be used to control turning on and off machinery.

Direct machine condition monitoring is accomplished via sensors, of which the most prevalent types are: accelerometers, tachometers, and proximity probes.

- *Accelerometers* are used to monitor vibrations of a machine. These are transducers for measuring the dynamic acceleration of a physical device, and are important to machine monitoring because they monitor system vibrations, which can be used to predict the life cycles of parts and to detect faults in machinery. Among the most common transducers are piezoelectric accelerometers, unbonded strain gage accelerometers, vibrating element accelerometers, and Hall effect accelerometers.

- *Tachometers* are used to determine the rotational speed of a shaft to provide phase information for the vibration data. These are transducers for measuring the rotational speed of a physical device, and are important to machine monitoring because they provide rotational speed as well as phase information, so that frequency components can be matched to shaft speed and position. Drag torque tachometers are among the most common.

- *Proximity Probes* are used to monitor the movement of a shaft. These are transducers for measuring the displacement of a physical device, and are important to machine monitoring because they monitor the movement of a rotating shaft. Proximity probes are usually found in 90-degree offset pairs to map an X-Y plot of the shaft movement. Then imperfections such as misalignment of the shaft, faulty bearings, or other external factors preventing perfect rotation can be prevented.

Most machine condition monitoring sensors require some form of signal conditioning to optimally function, such as excitation power to an accelerometer. Filtering on the signal is also common, to reduce both line noise and unwanted frequency ranges. Once the signals have been acquired, software-based signal processing is used to analyze and display the data from rotating machinery. The analysis can include calculation of overall vibration level (RMS, peak, crest factor); integration from acceleration to velocity or displacement; operation of online order analysis such as order tracking, order extraction, and order spectra computation; processing of digital and analog tachometer signals; application of limit testing on time data or power spectra; and drawing of a variety of plots ranging from spectral maps to time based plots.

3 Condition Monitoring Tools

There are many commercial tool offerings for machine condition monitoring. Major players in the marketplace include Brüel & Kjær Vibro, ClampOn AS, Data Physics Corporation, DLI Engineering Corp, Emerson Process Management, FLIR Systems Inc., GE Energy, Honeywell Process Solutions, among others [5]. We now introduce one such MCM tool; National Instruments' (NI) InsightCM^{TM} Enterprise [7]. InsightCM is an online MCM tool for monitoring health of critical rotating machinery and auxiliary rotating equipment. The goal is to optimize machine performance, maximize up-time, reduce maintenance costs, and increase safety. This solution allows maintenance specialists to acquire, analyze, visualize, and manage sensor data throughout the life cycle to draw diagnostic conclusions, manage alarms based on calculated features and sensor measurements, remotely configure, monitor, and manage acquisition devices, as well as authenticate users and devices to address network security concerns. By integrating into the IT infrastructure the tool can interact with existing databases and enterprise software.

Figure 2 illustrates key components of the InsightCM solution: monitoring systems, server for data management and analysis, data explorer clients, and management infrastructure.

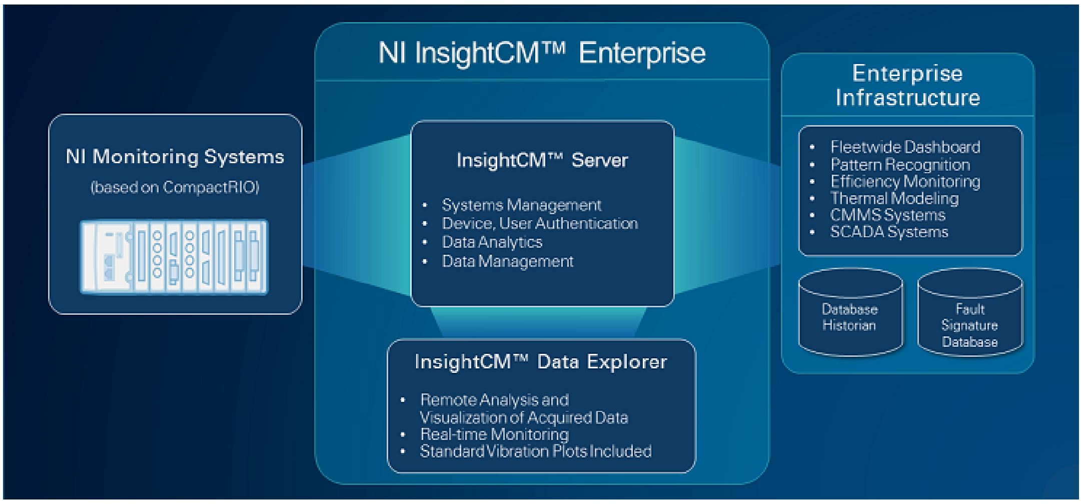

Figure 2: NI Insight^{TM} Architecture

The monitoring devices at the edges of the system are NI CompactRIO platforms [8]. These devices, in addition to sensors and I/O modules, have processing and reconfigurable components for inline data processing, control analytics, network communication, and timing. The CompactRIO devices supports a range of analog and digital sensors, such as proximity probes, accelerometers, pressure sensors, voltage and current sensors, thermocouples, and temperatur detectors. The monitoring system supports periodic monitoring as well as observation of important transient events such as start-ups and coast-downs. The *periodic recorder mode* allows data logging based on configurable time intervals, measurements, and user triggers. Dynamic waveform and static measurement sensors account for 80% of measurements in a predictive maintenance program, suitable for replacing traditional, manual diagnostic rounds. The *transient recorder mode* streams time waveform data during transient events at run-up and coast-down until steady state is maintained, and includes support for accelerometers, velocity meters, and proximity probes.

The server delivers analytics coupled with management of CompactRIO systems, data, and alarms. The server software manages reliable, loss-less communication across the entire architecture and includes capabilities to configure, view, and manage the remote acquisition systems. The software processes dynamic waveform data and analyzes RMS, peak-peak, true-peak, derived-peak, DC gap, crest factor, and spectral bands. It also supports custom measurements like bearing, gear, and other fault frequencies. Additionally, the server provides a security layer to authenticate and protect sensor and server data.

The data explorer provides interactive visualization and analysis of real-time and historical offline data stored in the server. The software package helps in remotely analyzing raw time-series data and results, drawing comparisons and viewing historical trends with support for standard vibration plots. The data explorer provides two modes: one for viewing periodically acquired data and one for viewing previously captured transient events. Users can detect imbalances, bent shafts, misalignment, bearing defects, and other faults in rotating machinery, and determine actions that need to be taken as part of diagnosis and maintenance procedures.

InsightCM has been widely used across multiple industrial domains including traditional power generation, oil and gas, renewable power generation, transportation and aerospace, heavy equipment, and manufacturing [9]. We discuss the use of InsightCM for a power grid monitoring application.

Duke Energy [10] is the largest power generation holding company in the US with a diversified energy portfolio mix and the capability to generate 58GW across 80 plants. Data used to be collected manually in periodic rounds on assets such as turbines, transformers, boilers, radiators, valves, motors, pumps, fans, and generators. The typical measurements include motor current, lube oil level, vibration, pressure, performance, and thermography. In this approach, 80% of the effort was spent on data collection, and 20% on analytics. Besides being labor intensive (about 60000 rounds/month), this approach has limited instrumentation and inconsistent diagnostics, which severely constrains the analysis. By employing InsightCM for condition monitoring, Duke Energy was able to phase out manual collection and spend more resources on the analysis. The system solution consists of one monitoring and diagnostic center for 80+ power plants controlling 30,000+ sensors distributed over 10,000+ assets. The monitoring architecture uses 1200 CompactRIO systems, generating and analyzing over 600 GB of data each week [11].

4 Summary

Machine condition monitoring (MCM) for large scale Industrial Internet of Things deployments will be critical for enterprises owning such systems. The need to eliminate catastrophic downtimes due to unexpected breakdowns and unnecessary maintenance costs has made condition monitoring critical for asset utilization and productivity across diverse industries. It has become imperative for such MCM systems to incorporate a sound management strategy to aggregate the data, conduct diagnostic analytics about the condition of the system, and facilitate predictive maintenance to reduce downtimes and maximize efficiency. According to a September 2015 report from Frost & Sullivan on Global Big Data Analytics Market for Test & Measurement, product development costs can be reduced by almost 25%, operating costs can be reduced by almost 20%, and maintenance costs can be reduced by 50% if big data analytics is applied for testing [3]. In this paper, we discussed the roles of enterprise MCMs and presented a representative tool, NI InsightCM$^{\text{TM}}$, which has been used for controlling and monitoring large scale distributed systems like a modern power generation network. To push the boundaries and maintain a competitive edge, the engineering community must provide MCM tools that find new correlations based on the monitored data to predict key future behaviors and even automatically take preventative actions with little human supervision.

References

[1] Gartner Research Inc., “The Internet of Things”, *Press Release*, November 2015, http://www.gartner.com/newsroom/id/3165317 .

[2] Edward A. Lee, Jan Rabaey, David Blaauw, Kevin Fu, Carlos Guestrin, Bjorn Hartmann, Roozbeh Jafari, Doug Jones, John Kubiatowicz, Vijay Kumar, Rahul Mangharam, Richard Murray, George Pappas, Kris Pister, Anthony Rowe, Alberto Sangiovanni-Vincentelli, Sanjit A. Seshia, Tajana Simunic Rosing, Ben Taskar, John Wawrzynek, David Wessel, “The Swarm at the Edge of the Cloud”, *Design & Test*, IEEE, Volume 31, 2014.

[3] National Instruments, “NI Trend Watch 2016”, http://www.ni.com/pdf/company/en/Trend_Watch_2016_Full.pdf, accessed Jan 12th, 2016.

[4] A. Davies. “Handbook of Condition Monitoring: Techniques and Methodology”, *Springer Science & Business Media*, 1997.

[5] Research and Markets, “Machine Condition Monitoring Market by Monitoring Type, Components, Monitoring Process“, *Applications, and Geography - Global Trend & Forecast to 2020*, November 2015.

[6] Robert Bond Randall, “Vibration-based Condition Monitoring: Industrial, Aerospace and Automotive Applications”, *John Wiley & Sons*, 2011.

[7] National Instruments, “NI InsightCM$^{\text{TM}}$ Enterprise for Condition Monitoring,” ftp://ftp.ni.com/pub/branches/asean/2015_05_29_insightcm.pdf, accessed Jan 12th, 2016.

[8] National Instruments, “NI Compact RIO”, www.ni.com/compactrio .

[9] National Instruments, “Featured Machine Condition Monitoring Case Studies,” http://www.ni.com/white-paper/12398/en, accessed Jan 12th, 2016.

[10] Duke Energy, https://www.duke-energy.com/, accessed Jan 20th, 2016.

[11] National Instruments, “NIWeek 2014 NI InsightCM$^{\text{TM}}$ Enterprise Announcement ,” http://www.ni.com/white-paper/52392/en/, accessed Jan 20th, 2016.

Software Challenges for Automotive Cyber-Physical Systems

Huafeng Yu
Toyota InfoTechnology Center, U.S.A.

Cyber-Physical Systems (CPS) are well adopted in the automotive domain. Due to the mobility nature of vehicles, CPS are also required to be miniaturized in size, but required to provide powerful computing with low power consumption, particularly for connected and autonomous vehicles. In this trend, automotive CPS become increasingly complex, heterogeneous, and decentralized. Meanwhile, they are required to be more safe, reliable, optimized, and adaptive [5], particularly for the control software in CPS [3]. Since most of these systems are considered to be safety-critical, they require higher safety assurance along with the whole development process, from requirement, modeling, implementation, integration, to verification & validation.

However, the development of CPS, including electronics and their control software is generally achieved in an isolated and parallel manner by suppliers, according to the requirements provided by car manufacturers (OEMs) [3]. Suppliers have the responsibility to assure the correctness, safety, and reliability of the components, whereas OEMs are in charge of requirements, final integration, evaluation and testing. This always leads to a gap between OEMs and suppliers.

Design tools, methods, technologies, and processes are well matured in the automotive domain, but new challenges still emerge for the design of next generation connected and autonomous vehicles. In this paper, we briefly discuss current big challenges in the design of safety-critical software for automotive CPS, and then present a formal integration framework to partially address these challenges.

1 Design Challenges

Formal specification. Automotive specification is mainly based on informal requirements. As a consequence, it frequently leads to ambiguity and misunderstanding between OEMs and their suppliers. However, a complete formal specification is also impossible due to the limitations in language expression, time, cost, performance, etc. Therefore, an appropriate formal specification, created in an incremental, composable, and reusable manner is indispensable for the building of reliable model-based integration and formal verification. This specification is expected to be independent from languages, tools, and platforms for wider adaptation and good reusability in industry.

Modeling. In system design, various general or domain-specific modeling languages are used [14] because of different needs, including different application domains, performance, expertise, cost, etc.. For instance, UML is used as a general modeling language, and SysML is adopted in systems engineering related applications. MATLAB/Simulink and Modelica are used as domain-specific modeling languages in a wide span of domains, while SCADE is mostly used for safety-critical systems and requires a strong background in rigorous design. Each language and its associated tool-set provide good support for their own development process from modeling to implementation. However a virtual integration using multiple languages and models turns out to be complicated, ambiguous and unpredictable.

Architecture modeling. Software architecture [18] was not considered essential, thus rarely formalized in conventional automotive design processes. Consequently, it generally leads to a manual, error-prone, time-consuming architecture exploration and validation. To avoid this problem, formalization, formal reasoning, and early-phase exploration are required, along with explicit quality attributes associated with particular architectural entities. Currently, architectural aspects of the system are not well expressed by general modeling languages. Architecture description languages, such as AADL[16], AUTOSAR [2] and EAST-ADL [6] were therefore proposed for embedded systems, especially avionic and automotive systems. A system-level design, considering both architecture and behavior, is becoming a promising solution to promote the virtual integration solution for embedded control systems [8, 20].

Timing specification. Semantics interoperability is one of the main issues in the composition of models, due to semantic dissimilarity between models and their inherent formalism, particularly for timing specification. The timing issue is among the most significant concerns in automotive system design [3, 20]. In general, the timing issue becomes more explicit when architecture is considered and the system is integrated, due to the gap between software and architecture design. To cope with timing-related semantics interoperability, one of the feasible solutions is to have a common formal model as the intermediate semantic model, and translate all other models into this common model. The intermediate model provides the formal semantics, based on which, the expected properties of the original models and their integration are checked. An example can be found in [21]. However, this approach requires a semantics preservation in the model translation, which is not practical in most cases. Another solution is related to unified formalism [13, 10]. However, this approach is more theoretical and not yet well applied in industry.

Integration frameworks. System integration is a big challenge due to isolated development, lack of integration and architecture specification, late phase for the integration, etc. A new trend to reduce integration issue is model-based integration. Research on model-based system integration has been discussed with regard to cyber-physical systems [19]; Service-oriented Architecture [15]; and heterogeneous models integration and simulation [7]. In addition, AUTOSAR[2] aims at component and platform-level integration for automotive systems, and System Architecture Virtual Integration (SAVI) program [8] targets avionic system integration. However due to multiple challenges in the model integration, integration frameworks, as opposed to specific integration solution, are becoming increasingly important. These frameworks are expected to consider formal specification and analysis, multi-view, and orthogonal attributes of the system, as well as *correct by construction* and *separation of concerns*, to reduce design complexity and validation time.

Certification. In addition to testing or verification methods, automotive engineers also need to consider certification. The developed vehicles are required to be certified to be safe by using the artifacts and evidences produced throughout the development cycle. Such a certification process helps to increase the safety confidence of the developed software and reduce OEM's liability. However, software certification in automotive domain is not yet well established, e.g., safety-relevant standards are not yet well defined, and the automotive safety standard ISO26262 is mainly based on process, lacking of support to product-oriented certification; lack of guidance and supervision from regulators or government agencies, unlike other domains of aviation and medical devices.

Security. Recent reports on security-related vehicle hacking involve various systems in many models from different OEM's [17] [9]. A series of successful hacking activities of current car models show the lack of system-level security consideration in vehicle hardware and software design, integration, certification, and production. These concerns also raise to US political level [12]. Integration of security mechanisms in vehicles involves not only computing resources but also other features, like safety, reliability, etc., which finally make it a big challenge.

Tool support. Tool support of MBE is one of the key concerns from an industry adoption viewpoint [4]. Specific model-based tool chains were developed as solutions, such as [1], for safety-relevant automotive embedded systems. However, the lack of serious consideration of formal aspects and integration semantics in these tool chains limits their support for a reliable integration. Tool qualification is another concern related to certification.

In the following, we summarize our current contribution, aiming at addressing previous issues in the context of model-based integration for automotive software control systems. This work is mainly inspired from [20, 11, 8, 5, 19, 4].

2 The VIF Integration Framework

Based on the previous exploration [21, 20], our current research mainly copes with a formal virtual integration framework for next-generation design of automotive control software. Our framework, called VIF (Virtual Integration Framework), promotes *correct by construction* in the early design phase, rather than a posteriori *Verification & Validation*. The main research topics involved in this framework include: formal timing specification, architecture modeling, design by contract, semantics interoperability and system optimization, as well as specification and modeling for different views and properties of the system, such as behavior, architecture, composition, and timing. All these techniques are adopted in the framework as key solutions to the challenges presented.

Formal timing specification and design by contract play a core role in the trustworthy model integration in our approach. High-level, formalized, multi-clock timing specification, considered as a central topic, is to be defined, observed and analyzed, based on software architecture. The formal contracts, used for describing the functional and non-functional specifications of the components, consider the architecture and platform models as well as their associated properties. A dual design methodology, called *Inside-out* and *Outside-in*, is proposed, where the first part addresses decomposition of a contract into sub-contracts, such that the latter can independently be given to automotive suppliers, instead of natural language specifications. The second part deals with a reliable integration of sub-systems to obtain the required system satisfying all contracts.

Figure 1 briefly illustrates the integration framework. High-level automotive requirements are initially analyzed, from which *formalizable* requirements are extracted, according to the technical formalizability and verifiability. These requirements are then used to create formal contracts for properties of timing, safety, performance, etc. In addition to these aspects, multiple modeling languages are applied for different views of the system in the framework, for instance, AADL for architecture modeling and Simulink for behavior modeling. Both timing specification and contracts are defined on these models, to enable a reliable integration. In the final step, behavior models are mapped onto the architecture model, considering pre-defined optimization criteria.

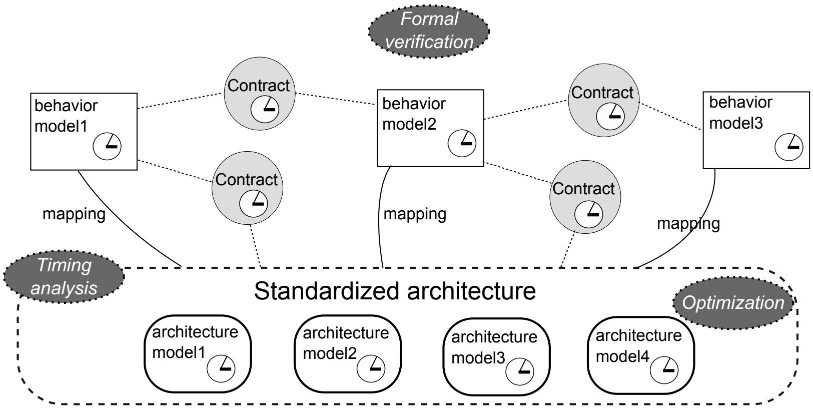

Figure 1: An overview of the integration framework

3 Conclusion

We briefly presented current big challenges in the design of software in automotive cyber-physical systems, including: formal specification, modeling, architecture, timing specification, integration framework, certification, security, and tool support. However, isolated software development, without consideration of system-level requirements and integration, is not enough. More rigorous system-level design methodologies are required to enable and enhance the system-level requirements on safety, reliability, performance, and security, etc. As a potential solution to partially address previous challenges, we exhibited our proposed formal integration work, VIF, with focus on architecture modeling, timing specification, and integration correctness.

References

[1] E. Armengaud, M. Zoier, et al. “Model-based Toolchain for the Efficient Development of Safety-Relevant Automotive Embedded Systems”. In *SAE World Congress & Exhibition*, 2011.

[2] AUTOSAR (AUTomotive Open System ARchitecture). http://www.autosar.org/.

[3] M. Broy. “Challenges in Automotive Software Engineering”. In *International Conference on Software Engineering (ICSE)*, 2006.

[4] M. Broy, M. Feilkas, M. Herrmannsdoerfer, S. Merenda, and D. Ratiu. “Seamless Model-Based Development: From Isolated Tools to Integrated Model Engineering Environments”. *Proceedings of the IEEE*, 98(4):526–545, 2010.

[5] DARPA. “Adaptive Vehicle Make (AVM) Program”. http://www.darpa.mil/Our_Work/TTO/Programs.

[6] EAST-ADL. http://www.east-adl.info.

[7] J. Eker, J.W. Janneck, E.A. Lee, J. Liu, X. Liu, J. Ludvig, S. Neuendorffer, S. Sachs, and Y. Xiong. “Taming Heterogeneity - the Ptolemy Approach”. *Proceedings of the IEEE*, 91(1):127–144, 2003.

[8] P.-H. Feiler, J. Hansson, D. de Niz, and L. Wrage. “System Architecture Virtual Integration: An Industrial Case Study”. Technical report, Software Engineering Institute, 2009. CMU/SEI-2009-TR-017.

[9] M. Harris. “Researcher hacks self-driving car sensors”. *IEEE Spectrum*, 2015.

[10] E. A. Lee and A. Sangiovanni-Vincentelli. “A Framework for Comparing Models of Computation”. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(12):1217–1229, 2006.

[11] Y. Ma, H. Yu, T. Gautier, P. Le Guernic, J.-P. Talpin, L. Besnard, and M. Heitz. “Toward Polychronous Analysis and Validation for Timed Software Architectures in AADL”. In *Design, Automation, and Test in Europe (DATE)*, pages 1173–1178, 2013.

[12] E. Markey. Tracking & hacking: Security & privacy gaps put american drivers at risk, 2015.

[13] D. Mathaikutty, H. Patel, S. Shukla, and A. Jantsch. “Modelling Environment for Heterogeneous Systems based on MoCs”. In *Forum on specification and Design Languages (FDL)*, pages 291–303, 2005.

[14] K.D. Müller-Glaser, G. Frick, E. Sax, and M. Kühl. “Multiparadigm Modeling in Embedded Systems Design”. *IEEE Transactions on Control Systems Technology*, 12(2):279–292, 2004.

[15] A. Rossignol. “The Reference Technology Platform”. In *CESAR - Cost-efficient Methods and Processes for Safety-relevant Embedded Systems*. Springer, 2013.

[16] SAE Aerospace (Society of Automotive Engineers). “Aerospace Standard AS5506A: Architecture Analysis and Design Language (AADL)”. *SAE AS5506A*, 2009.

[17] D. Schneider. Jeep hacking 101. *IEEE Spectrum*, 2015.

[18] M. Shaw and D. Garlan. “Software Architecture: Perspectives on an Emerging Discipline”. *Prentice Hall Englewood Cliffs*, 1996.

[19] J. Sztipanovits, X. D. Koutsoukos, G. Karsai, N. Kottenstette, P.J. Antsaklis, V. Gupta, B. Goodwine, J.S. Baras, and S. Wang. “Toward a Science of Cyber-Physical System Integration”. *Proceedings of the IEEE*, 100(1):29–44, 2012.

[20] H. Yu, Y. Ma, T. Gautier, L. Besnard, J.-P. Talpin, and P. Le Guernic. “Polychronous Modeling, Analysis, Verification and Simulation for Timed Software Architectures”. *Journal of Systems Architecture (JSA)*, 59(10):1157–1170, 2013.

[21] H. Yu, Y. Ma, Y. Glouche, J.-P. Talpin, L. Besnard, T. Gautier, P. Le Guernic, A. Toom, and O. Laurent. “System-level Co-simulation of Integrated Avionics Using Polychrony”. In *ACM Symposium on Applied Computing (SAC)*, 2011.

Cyber-Physical-Social Clouds: Future Insights

Rajiv Ranjan[1], Prem Prakash Jayaraman[2], Ellis Solaiman[1], and Dimitrios Georgakopulos[2]
[1]School of Computing Science, Newcastle University, United Kingdom
[2]School of Computer Science and Information Technology, RMIT University, Australia

Cyber-physical systems (CPS) are a vast interlinked network of things/devices, computing resources, applications/services and humans, that use a range of sensors, actuators and communication topologies, to link the computations systems (platforms) with the physical world. CPS drives the vision of a “smart interconnected cyber social world” where the physical social world is monitored by sensors in real time, and the services in the cyber world use the data to directly influence the decision making in the physical world.

The Internet of Things (IoT) and cloud computing are integral parts of the cyber-physical- ecosystem [4]. While the IoT is seen as the means for connecting disparate sensing and actuation devices (via the Internet) with applications and services, cloud computing offers computation and storage capabilities required by those data processing applications and services. Aided by the low cost and availability of wired and wireless networking technologies as well as cheap sensors and actuator devices, the IoT will transform the Internet into a fully integrated smart environment where large amounts of generated data can be shared across diverse applications and platforms. The social impact of connecting clouds and IoTs to form such smart environments will be a revolution that promises to change people's lives across a variety of domains including; smart homes and smart cities with smart management systems for traffic management and accident prevention, intelligent advanced warning systems to help communities with prediction and preparation for environmental conditions such as storms and floods, and smart healthcare monitoring delivering immediate automated on demand and real time data on the wellbeing of patients.

CPS takes advantage of cloud's pay-as-you go model to support various applications such as emergency health care, evacuation and rescue systems, disaster-management applications, and personal fitness systems. The NIST definition [1] of cyber-physical cloud computing is "a system environment that can rapidly build, modify and provision cyber-physical systems composed of a set of cloud computing based sensor, processing, control, and data services". Based on this definition, we define the cyber-physical-social clouds as "an ecosystem of tools, frameworks and systems that can facilitate rapid development, deployment and management of cyber-physical applications composed of Things (sensors and actuators), Clouds (processing, control, data services and applications) and Humans (Social bounded by a closed loop data flow relationship".

Figure 1 provides an overview of the Cyber-Physical-Social Clouds (CPSC) using an onion model to describe the expanding and extending relationship between various layers. As depicted in the figure, the Physical space is composed of embedded systems that could include a range of sensing and actuation devices (e.g. wireless sensor networks), things (e.g. fitbit), smart phone and smart vehicles (e.g. google car). They are interwoven with various ubiquitous communication capabilities allowing them to be networked (e.g. Internet). The cyber cloud hosts applications, services supported by big data processing components (such as Apache spark, Hadoop etc.), in order to process, analyse and compute actionable outcomes from the sensed data. The actionable responses are propagated back into the physical world via actuators. The cyber-physical ecosystem also encompasses a human aspect wherein humans provide data, act on analysed data and make informed decisions.

The components of CPSC namely the IoT and cloud are very distinct and complimentary. While IoT provides localization, therefore enabling low latency and context awareness, the cloud provides global centralization. Further, Clouds offer virtually unlimited, scalable access to resources (computing, storage) while IoT is generally conceived as a resource constrained environment. Finally, IoT bring with it the sheer scale (estimated to be 50 billion devices by 2020 [2]) and complexity.

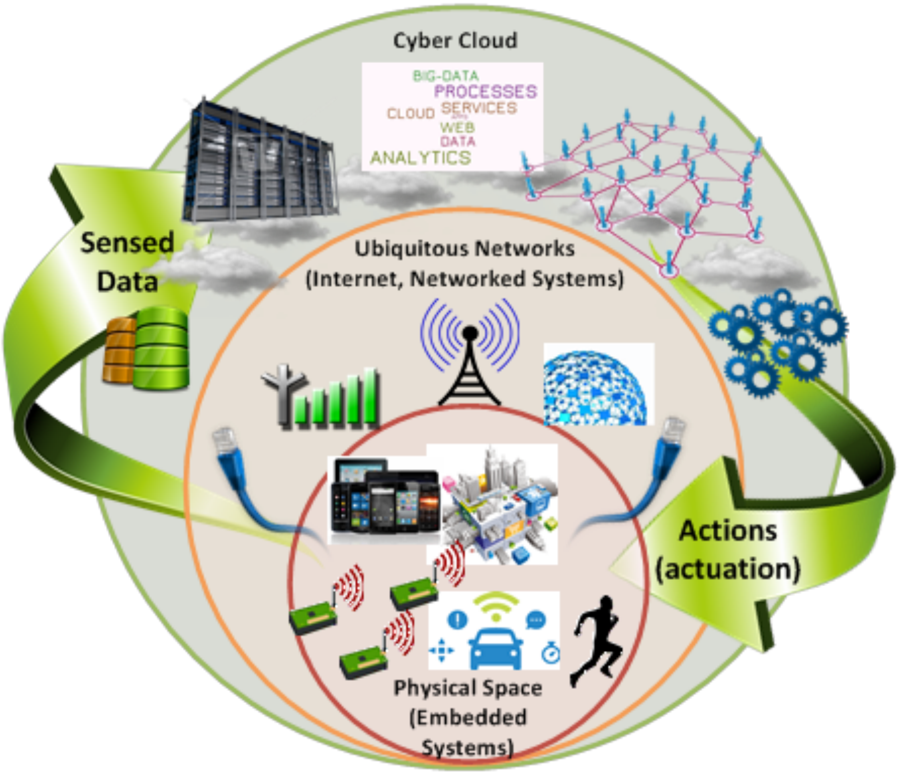

Figure 1: Cyber-Physical Clouds

Rapid advances in IoT (e.g. mobile computing, man-machine, machine-machine communications, and smart phone networks) will cause a paradigm shift in the design of CPSC applications and operations, by bringing improvements not only to the quality of service (QoS) but also to quality of experience (QoE), cost efficiency, reliability, security, and energy efficiency. Moreover, resources in data centers and cloud infrastructures have to be efficiently managed and scheduled to optimize reliability and scalability of CPSC under various constraints of QoS and QoE. Besides, CPSCs are expected to deal with data directly coming from trans-domain applications (e.g. traffic accident detection in smart transportation, energy management in smart grid, health monitoring and evaluation), which could be in various forms such as GPS coordinates, flood level, temperature, rainfall rate, vehicle speed, electricity consumption, etc. How to coordinate various applications of heterogeneous systems and facilitate a deeper integration, interaction and personalization of the physical, cyber, and social domains is an important challenge. In order to build the next generation CPSC applications and services, it is essential to address the challenges introduced by IoT and Clouds while leveraging on their advantages.

Challenges:

In this newsletter, we highlight some of the key challenges that need to be addressed in order to develop scalable, reliable and cost-efficient CPSC applications and services. These include:

Real-time data processing, and actuation: IoT is tightly bound by its real-time data processing and its subsequent analysis that can help with driving decisions making processes. Hence, CPSC applications require the development of models to reduce latency and inefficient communication between Clouds and IoT in order to achieve real-time data processing and actuation needs.

Multi-tenancy Clouds: IoT is a distributed system that produces enormous amounts of data. In fact, IoT is considered a major source of Big-data. Hence, there is a need to support storage and processing of un-structured and semi-structured big data coming from distributed sources to provide real-time/near real-time services. This will require multi tenancy cloud models to meet the scalability and real-time needs of CPSC applications.

Dependable QoS management: CPSCs bring unique challenges in the form of IoT that is very different to traditional cloud-based applications. For example, CPSC system performance is based not only on the Clouds but also the IoT devices. Hence, careful consideration and prediction of QoS and corresponding SLAs is key for CPSC applications. These QoS metrics could include performance targets/constraints of CPSC applications, distributed processing and storage of the massive data as well as run-time migration of cloud servers (VMs).

Discovery and Service Composition (Clouds and IoT): Discovery is a mechanism that will enable users and application likewise to find and access Cloud services and corresponding IoT data without the need to know the actual source of the data, sensor description, or location. An intrinsic requirement of CPSC applications is to manage heterogeneity at various level such as different types of sensors, data and cloud services. Currently the standards that govern the development of these components are at their infancy. Even with appropriate standards, given the growth rate of IoT, it is expected that we will soon face the challenge of integrating a multitude of devices and data stemming from IoT. Hence, there needs to be mechanisms that allows the discovery of these IoT devices and the corresponding services (e.g. storage service) that fit the requirement of the CPSC applications.

CPS identity management: In the CPSCs, a service provider could potentially receive data from hundreds of data sources. The provider needs to be able to determine to which end user the data belongs to [3]. In addition to data and data source identification [3], we need to be able to both specify and identify to which user/s the ‘thing/s’ belongs to, thus determining the context (e.g. relating to which person) in which a particular sensor or actuator is operating. Sensors could be shared and generate data that is relevant to a number of different users. For example a proximity sensor in a hospital should identify when particular staff or patients are near it. When the “things” are actuators, it becomes particularly important that the right actions are triggered for the right actors. Also, conflicts might arise. In a home IoT scenario, different members of a family may have different temperature preferences, so policies over thermostat control could conflict. In such cases and others, policy conflict detection and resolution mechanisms maybe of utmost importance.

Virtualised Cloud IoT: CPSC systems provide a unified view of the physical world to the cyber user and vice-versa. Hence, it is essential to maintain a virtual environment that represents the physical world monitored by the IoT devices. This will require the development of various domain ontologies and corresponding reasoning techniques that can best describe the current physical world consistently in order to facilitate discovery and service composition.

Emerging CPSC business models: CPSC is still in its infancy and so is challenged by the development of relevant and appropriate business models. E.g. the classical cloud model may no longer be applicable as CPSC is not only about the computer resources, it encompasses the IoT and the social aspects (humans). Hence, business models need to be developed that takes into consideration all the 3 factors of CPSC namely, IoT, Clouds and Humans.

Platforms for CPSC development and deployment: The development of CPSC systems is not a trivial task and requires careful consideration of various aspects including the construction and deployment of IoT infrastructure and cloud infrastructure, while giving due diligence to the social aspect. The question here is what could be the essential features of a cloud-based frameworks for CPSCs in order to support the right level of development techniques and methodologies?

References

[1] Eric D. Simmon, Kyoung-sook Kim, Eswaran Subrahmanian, Ryong Lee, Frederic J. de Vaulx, Yohei Murakami, Koji Zettsu, and Ram D. Sriram, "A Vision of Cyber Physical Cloud Computing for Smart Networked Systems", *NIST Interagency/Internal Report (NISTIR) – 7951*, August 26, 2013.

[2] Cisco, "Internet Of Things Will Deliver $1.9 Trillion Boost To Supply Chain And Logistics Operations", April 15, 2015, http://newsroom.cisco.com/press-release-content?articleId=1621819.

[3] P. N. Mahalle, B. Anggorojati, N. R. Prasad, and R. Prasad, "Identity Authentication and Capability Based Access Control (IACAC) for the Internet of Things", *Journal of Cyber Security and Mobility*, vol. 1, no. 4, pp. 309–348, 2013.

[4] L. Wang and R. Ranjan, "Processing Distributed Internet of Things Data in Clouds", *IEEE Cloud Computing*, Vol. 2, No. 1, Feb 2015, BlueSkies Column, IEEE Computer Society.

1 Workshop

- DAC-2016 Workshop on Design Automation for Cyber-Physical Systems (CPSDA-2016)
- INFOCOM-2016 Workshop on Cross-Layer Cyber-Physical Systems (CPSS-2016)

2 Special Issues in Academic Journals

- Integration, The VLSI Journal special session on Hardware Assisted Techniques for IoT and Big Data Applications
- IEEE Transactions on CAD special issue on CAD for Cyber-Physical System
- IEEE Transactions on Computers special issue on Smart City Computing
- IEEE Transactions on Multi-Scale Computing Systems special issue on Hardware Software Crosslayer Technologies for Trustworthy and Secure Computing

3 Special Sessions in Academic Conferences

- ICCAD-2015 special session on The Landscape of Smart Buildings: Modeling, Management and Infrastructure
- ICCD-2015 special session on Cyber-Physical Integration and Design Automation for Microfluidic Biochips
- DAC-2015 special session on Securing Cyber-Physical Systems: From Surveillance to Transportation and Home
- DAC-2015 special session on The Researcher Who Cried Wolf

Call for Contributions

TBD