Software Engineering for Smart Cyber Physical Systems

ACM SIGSOFT Webinar Series
September 8, 2015

Danny Weyns and Tomas Bures
Aims of the Webinar

• Introduce *smart Cyber Physical Systems (sCPS)* and explain the challenges they pose to software engineers

• Outline key themes of software engineering for sCPS

• Highlight open research topics in the area of sCPS
Outline

• Motivation
• Software Engineering for Smart Cyber Physical Systems Workshop (SEsCPS)
• Key themes of SEsCPS
• Open research topics
• Wrap-up
• Q&A
Motivation

• What are Cyber Physical Systems?

• What are Smart Cyber Physical Systems?
• Why are they so important?

• Why do traditional Software Engineering approaches not suffice?
What are Cyber Physical Systems?

“Engineered systems that are built from, and depend upon, the seamless integration of computational and physical components” [NSF]

“Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa.” [Lee 2008]
What are Cyber Physical Systems?

• CPS – systems of collaborating computational elements controlling physical entities
• Control usually involves feedback loops

• Combine multiple disciplines
  – Physical (continuous/discrete dynamics)
  – Electrical/electronic
  – Communication
  – Software
  • Real-time, Sensing and signal processing, Control, Cognition, Distributed decision making, Self-awareness, Self-adaptability ...
What are **Smart** Cyber Physical Systems?

- “The next generation embedded ICT systems that are **interconnected and collaborating**, providing citizens and businesses with a wide range of innovative applications and services” [H2020]
- Large-scale software intensive and pervasive systems, combine various data sources to efficiently **control** real-world ecosystems
- Have to operate under **uncertainty**
- Equipped with **intelligence** to deal with environment dynamics, control their emergent behavior, and tolerant to threats
- Deliver the “best possible” service in a given situation.
What are Smart Cyber Physical Systems?

• “The next generation embedded ICT systems that are interconnected and collaborating, providing citizens and businesses with a wide range of innovative applications and services” [H2020]

• Large-scale software intensive and pervasive systems, combine various data sources to efficiently control real-world ecosystems

• Have to operate under uncertainty

• Equipped with intelligence to deal with environment dynamics, control their emergent behavior, and tolerant to threats

• Deliver the “best possible” service in a given situation.

Constitute systems that are supposed to address many environmental and societal challenges – smart energy, smart mobility, ambient assisted living ...
Perspectives on being “Smart”

CPS 5C level architecture

[Lee, Bagheri, Kao, 2014]
Perspectives on being “Smart”

CPS 5C level architecture
[Lee, Bagheri, Kao, 2014]

Cyber-physical convergence
[Conti et al., 2012]
Perspectives on being “Smart”

CPS 5C level architecture
[Lee, Bagheri, Kao, 2014]

Cyber-physical convergence
[Conti et al., 2012]

Self-adaptability
(e.g. Rainbow framework)
[Cheng et al, 2004]
Perspectives on being “Smart”

CPS 5C level architecture
[Lee, Bagheri, Kao, 2014]

Cyber-physical convergence
[Conti et al., 2012]

Self-adaptability
(e.g., Rainbow framework) [Cheng et al, 2004]

Ad-hoc self-organization
E.g. autonomic component ensembles
[EU ASCENS project]
Why Do Traditional SE Approaches Not Suffice?

• sCPS are software-intensive systems ... but with many specific challenges:
  – Blurring boundaries between hardware and software
  – Inherent uncertainty
  – Large scale and complexity
  – Role of end-users
  – Open-endedness
  – Strong notion of locality, etc.
Why Do Traditional SE Approaches Not Suffice?

- Engineering sCPS requires expertise from combination of disciplines
  - Physical, electrical, software
  - SW: embedded & real-time systems, distributed systems, pervasive computing, self-adaptive systems, systems of systems, AI, agents, ...

- How to combine the expertise in a reliable way is an open question
Software Engineering for Smart Cyber Physical Systems Workshop (SEsCPS)

• Goals
  – Increase the understanding of problems of SE for sCPS
  – Study the underlying foundational principles for engineering sCPS
    • E.g., Reasoning about uncertainty, dealing with emergent behavior, distributed control
  – Identify promising SE solutions for sCPS
    • E.g., Models, engineering processes, integrated V&V
Software Engineering for Smart Cyber Physical Systems Workshop (SEsCPS)

• First edition
  – One day event organized at ICSE 2015
  – 19 submissions; 10 papers accepted
  – Morning
    • Keynote by David Garlan (CMU)
    • 10 short presentations by participants
  – Afternoon
    • Four discussion groups
    • Plenary discussions
Key themes of SEsCPS

• Modeling challenges

• Faults and conflicts

• Modeling, testing and verification

• Collaboration
Modeling Challenges

• Today’s model-based CPS approaches have various problems:
  – Difficult to integrate the different modeling approaches
  – Difficult to make trade-offs and ensure consistency/completeness
  – Difficult to integrate humans “in the loop”

• Proposed approach:
  – Unified representation through extensions of software architecture and use of architectural views
  – Tools for dependency analysis and coordination
  – Stochastic multi-player games
Faults and Conflicts

• Localization of faults in models
  – E.g., Use statistical models for normal behavior; identify and trace variables that cause adverse and anomalous events

• Conflicts in cooperation of CPS
  – E.g., Employ and check metadata to handle conflicts of actuating the same physical entity in different ways

• Conflicts in specifications when CPS evolve
  – E.g., Versioning and linking of engineering artifacts to cope with multiple heterogeneous engineering domains

• Fault Localization in Embedded Control System Software, K. Liang, Z. Bai, M. C. Cavosoglu, A. Podgurski, S. Ray, Case Western Reserve University, USA
• An Application Conflict Detection and Resolution System for Smart Homes, M. Yagita, F. Ishikawa, S. Honiden, National Institute of Informatics and the University of Tokyo, Japan
• Versioning in Cyber-Physical Production System Engineering? Best-Practice and Research Agenda, R. Mordinyi, S. Biffl, Vienna University of Technology, Austria
Modeling, Testing and Verification

• Uncertainty in CPS is multi-faceted
  – E.g., Framework for uncertainty based on fuzzy logic taking into account component input, output, behavior and composition

• High-level application requirements vs resource-limited microcontrollers
  – E.g., Expose “components” as services in resource-limited microcontrollers

• A Model-based Approach for the Specification of a Virtual Power Plant Operating in Open Context, V. Koutsoumpas, TUM, Germany
• Towards Cyber-Physical Systems as Services: the ASIP Protocol, M. Bordoni, M. Bottone, B. Fields, N. Gorogiannis, M. Margolis, G. Primiero, F. Raimondi, Ocado Group and Middlesex University, United Kingdom
• Accelerating Regression Testing for Scaled Self-Driving Cars with Lightweight Virtualization – A Case Study, C. Berger, University of Gothenburg, Sweden
• Verification and Validation in Cyber Physical Systems: Research Challenges and a Way Forward, X. Zheng, C. Julien, University of Texas, USA
Modeling, Testing and Verification

• Demands on tools and techniques for verification and validation of CPS
  – Trial-and-error testing (state of the practice) does not provide sufficient rigor in error detection
  – Formal methods provide a desired level of expressiveness but are neither intuitive nor efficient
  – Existing simulation tools are limited in their capabilities to jointly model physical and cyber components
Collaboration

• Additional functionality in the interplay of constituents of CPS
  – E.g., Document and analyze systems on an instance level to support automated verification at design time and adaptation at runtime

• Inherent heterogeneity and distribution of CPS
  – Scalability and robustness calls for decentralized approaches
  – E.g., Domain specific modeling language with code generation and runtime environment that supports integration logic

Image Credit: PALUNO, The Ruhr Institute for Software Technology
Open research challenges for SEsCPS

- Aligning different disciplines
- Humans in the loop
- Pragmatic vs/and systematic engineering
- Uncertainty

Questions rather than answers...
Aligning Different Disciplines

• Disciplines can learn from each other
• Necessity to be open-minded and respect disciplines
  – Every discipline has its own (proven) expertise
• Cross-world connections
  – What is the vocabulary to talk to each other?
  – How to identify the mismatches in the vocabulary
  – How to identify assumptions from one world for the other
Aligning Different Disciplines

• Empirical research to understand the world of CPS
  – To what extent is there awareness of the lack of alignment in CPS?
  – How big is the gap between disciplines?
  – Is there a need for alignment? When/Where?

• What is the role of education?
Human in the Loop

- Different “types of humans” in CPS
  - Engineers
    - Create and evolve CPS
  - End users
    - State and change requirements
    - Use the system
  - System-external humans
    - Are influenced by the CPS (e.g. pedestrians in context of autonomous driving)

- How to handle awareness of changes?
  - How should a notification model look like?
  - How to involve humans in decision making and high-quality feedback to CPS?
Human in the Loop

• What are the effects of changes end users introduce to CPS?
  – Can the CPS „calculate“ the costs of adaptation to new requirements?
  – What are the dependencies of the capabilities of CPS on human input?
  – How are CPS capable of handling „bad“ user input to assure most effective/efficient operation?
Pragmatic and Systematic Engineering

• Key mismatch problem in CPS
  – Software engineers and the domain experts speak different languages

• Common practice
  – The domain experts build a system
  – When they’re desperate, they call in a software engineer to try to fix the problem
  – The software engineer has to become immersed in the domain

• Recent anecdotal efforts hint at interdisciplinary teams that *start* with domain experts and software engineers
Pragmatic and Systematic Engineering

• Can we leverage “smartness” to obtain high quality systems?
  – The system may have some flaws, as long as it has smartness allowing the system to recover or repair itself
• Can we derive models that are inherently compose-able?
  – To support open-endedness and evolution of CPS
  – Based on component “premises,” the “connectors” could account for consistency, timing, security, etc.
• Need for seamless integration of design and runtime
  – Improve design-time techniques and make them accessible to domain experts
  – Support run-time monitoring and adaptation
Uncertainty

• sCPS are subject to a variety of uncertainties
  – Changing or new requirements
  – Environment / context dynamics
  – Modeling uncertainty
  – Uncertainty in code
  – Behavior uncertainty
    • Uncertainty due to humans in the loop
    • Emergent behavior of composite system
    • Uncertainty about the adaptation of the system
  – Infrastructure uncertainty
  – Among other types...
Uncertainty

• Mastering uncertainty via runtime adaptation
  – Need for methods to systematic monitor CPS
    • Some work in the area of requirements monitoring
    • Uncertainty of sensing, data representation, etc.
  – Methods for analyzing and planning
    • Decentralization aspect; multi-ownership
  – Perpetuability perspective
    • Flexibility by design
    • Adapt when needed
    • Return to engineer when goals cannot be achieved
    • Evolve online
Wrap-up

• sCPS are central to many environmental and societal challenges
• Challenging domain for software engineers

• Key observations of the workshop
  – Engineering sCPS requires a truly multi-disciplinary approach
  – Humans is a first class player
  – Uncertainty rules
Wrap-up

• Long term challenges include:
  – Understand the alignment of disciplines and identify languages, models, methods, etc. to handle this
  – Understand and realize the perpetuability perspective on engineering sCPS
  – Define prototypical applications to evaluate and compare new methods, techniques and tools

• SEsCPS aims at creating a forum which will make this possible
Bibliography


• Proceedings of SEsCPS: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7173534

• Links to resources: http://d3s.mff.cuni.cz/conferences/sescps
Q&A