CALL TO ACTION

• Self-adaptive systems (SAS) researchers ought to drive the cyber physical systems research agenda
• CPS R&D affords spectacular and transformative opportunities
• As software engineering and SAS researcher be a leader, inject your expertise, and exploit these opportunities
CPS SOCIETAL IMPACT

- Virtually every engineered system is affected by advances in the networked and cloud capabilities of CPS
- Future CPS applications are expected to be more transformative than the IT revolution of the past three decades
- Enormous funding opportunities
- CASCON Workshop on CPS
  Mylopoulos, Litoiu, Müller
NIST REPORTS ON CPS

• Distilled perspectives on CPS from experts from industry, academia and government

• Future CPS will sweeping impacts on how we live, work and do business

http://www.nist.gov/el/isd/cps-020613.cfm
CPS

• Smart systems that encompass computational and physical components, seamlessly integrated and closely interacting to sense the context of the real world
• Physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing, communication and control core
• Tight integration and coordination between computational and physical resources
• Add capabilities to physical systems
• Exceeds today’s systems in adaptability, autonomy, efficiency, functionality, reliability, resiliency, safety and usability
CPS

Tight integration and coordination between computational and physical resources

Physical Resources

CPS

Computational Resources
CPS

Smart systems that encompass computational and physical components, seamlessly integrated and closely interacting to sense the context of the real world
CONFLUENCE OF SENSORS, NETWORKS, CLOUDS, DEVICES, AND APPS

Convergence of analytical and cognitive capabilities, real-time and networked control, pervasive sensing and actuating, as well as compute and storage clouds
EXERCISE

What is the difference between the CPS and IoT? Convince each other that there is a difference.
CPS
CONCEPT
MAP

Properties
- Networked
- Distributed
- Real-time
- Adaptive
- Self-adaptive
- Human in loop
- Smart

Functions
- Sense
- Monitor
- Analyze
- Reason
- Actuate

Systems
- Feedback sys
- Control sys
- Adaptive Sys
- M2M Sys
- II Sys

Foundations
- Models
- V&V
- Model identification
- Requirements
  @runtime
SMART BUILDINGS
CONNECTED CARS
AUTONOMOUS VEHICLES
CAN WE BUILD SUCH AMAZING CPS SYSTEMS?

MINDBOGGLING SITUATION AWARENESS

*HUMANS ARE AMAZINGLY ADAPTIVE*

*Tokyo, Japan*
Who thinks this experiment is possible right now?
GOOGLE DRIVERLESS CAR — 2:07 MINS
LICENSED IN CALIFORNIA, NEVADA AND FLORIDA
CPS NURSE LOG

It is critical to educate engineers and scientists in cyber physical systems to reap the competitive advantages of developing and mastering advanced CPS.
CPS

Applications

Control

Computing

Communications
SELECTED CPS APPLICATIONS

- Energy
- Computing
- Environment
- Vehicles

CPS Foundations
Computing, Communications, Control
SELECTED CPS APPLICATIONS

- Energy
- Computing
- Environment
- Vehicles

CPS Foundations
SELECTED CPS APPLICATIONS

- Smart Grid
- Smart Buildings
- Green Computing
- Smart shopping and elderly web tasking
- Digital Ecosystems
- Wearable Computers
- Smart Ocean
- Smart water
- Tidal Energy
- UAVs
- Connected Cars
- Smart Transport

Energy
Computing
Environment
Vehicles

CPS Foundations
SELECTED CPS FOUNDATIONS

Adaptive Systems and MART
Cloud Computing and Cyber Security
Analytics and Machine Learning
Wireless and Mobile Networks
Wireless Sensor Networks (WSN)
Software Defined Networks (SDN)
Networked and Distributed Control
Model Adaptive and Predictive Control
Autonomic Computing

Computing
Communications
Control

CPS Foundations

University of Victoria
EXERCISE

• Why is it so important to educate engineers and scientists in the foundations of CPS?
CONTROL AND SYSTEMS SCIENCE FOR CPS

• Abstraction infrastructure to bridge digital and physical system components
  • Integrated and networked control
  • Adaptive and predictive control
  • Composition of control
  • Reference models

• Characterizing problems & guaranteeing solution quality
  • Utility function policies and optimization
  • Uncertainty characterization and quantification

• Assurance at runtime
  • Models at runtime
  • V&V at runtime
CONTROLLER AS SOFTWARE
ACRA
AUTONOMIC COMPUTING REFERENCE ARCHITECTURE
AUTONOMIC COMPUTING REFERENCE ARCHITECTURE (ACRA)

HIERARCHY OF CONTROLLERS

Utility function policies
Goal policies
Action policies
POLICY TYPES FOR CPS

• Action policies
  – If-then action rules specify exactly what to do under a condition
  – Rational behaviour is compiled in by the designer
  – Basis for reflex agents

• Goal policies
  – Requires self-model, planning, conceptual knowledge representation

• Utility function policies
  – It chooses the actions to maximize its utility function
  – Finer distinction between desirability of different states than goals
  – Numerical characterization of state
  – Needs methods to carry out actions to optimize utility
UTILITY THEORY

- Utility theory with service level agreements
- One cost objective function
- Optimization

John Wilkes, HP Labs
Utility functions, prices, and negotiation

John Wilkes, Google
formerly HO Labs
ADAPTIVE CONTROL

• Adaptive control is the idea of “redesigning” the controller while online, by
  – looking at its performance and
  – changing its dynamic in an automatic way

• Motivated by aircraft autopilot design
  – Allow the system to account for previously unknown dynamics

• Adaptive control uses feedback to observe the process and the performance of the controller and reshapes the controller closed loop behavior autonomously.
ADAPTIVE CONTROL

- Modify the control law to cope by changing system parameters while the system is running
- Different from Robust Control in the sense that it does not need *a priori* information about the uncertainties
  - Robust Control includes the bounds of uncertainties in the design of the control law.
  - Therefore, if the system changes are within the bounds, the control law needs no modification
CHARACTERISTICS OF THREE-TIER HIERARCHICAL INTELLIGENT CONTROL SYSTEMS

- The three-tier architecture is prevalent
  - service-oriented software systems
  - automation systems
  - decision-support systems
  - many other types of adaptive and self-managing systems

- Three layers
  - separate concerns (e.g., three-tier web architecture where the presentation and data tiers are separated by an application or business logic tier)
  - Impose a hierarchy along a dimension where such a dimension represents an extra-functional requirement or quality criterion as outlined
    - performance, internal state, goals, policies, plan sophistication, “intelligence”, or quality of service
  - The scales depend on the actual requirement or criterion of the dimension
    - from specific goals to general goals
    - from high precision to low precision
    - from fast performance to slow performance
    - from stateless to memory of the past and predictions of the future
    - from hard-wired policies to utility-function policies (i.e., trade-off analysis)
  - Rationale for three tiers is usually not explicitly stated, but frequently a natural fit
AI and robotics communities generated several closely related three-layer reference control architectures:

HIERARCHICAL INTELLIGENT CONTROL SYSTEM (HICS) ARCHITECTURE

HICS ARCHITECTURE

• Hierarchical Intelligent Control System (HICS)
• HICS is probably the most general reference architecture emerging from AI and robotics
• Three HICS layers (from bottom to top)
  – Execution
  – Coordination
  – Organization Level
• The complexity of reasoning (i.e., intelligence) increases from the execution to the organization level
• The flexibility of policies decreases from organization to execution (i.e., the precision of increases).
ROBOTICS INSPIRED THREE-LAYER ARCHITECTURE MODEL

Goal Management

Change Management

Component Control

Create new plans based on high-level objectives

Execute pre-computed plans

Application control: Sensors, actuators

The DYNAMICO Reference Model

- Guides the design of highly dynamic self-adaptation mechanisms
- Manages uncertainty due to changing requirements
- Preserves context-awareness in self-adaptation

ADAPTIVE CONTROL—MRAC
MODEL REFERENCE ADAPTIVE CONTROL

Two layers

Reference Input $r(t)$

Control Error $e(t)$

Reference Model

Model Output $y_m(t)$

Controller Parameters $c(t)$

Controller

Controller Input $u(t)$

Managed System

Measured Output $y(t)$

Adaptive Algorithm

Diagram showing the two-layer structure of Model Reference Adaptive Control (MRAC) with the reference input $r(t)$ and control error $e(t)$ feeding into the reference model, which provides the model output $y_m(t)$. The controller adjusts its parameters $c(t)$ based on the error and produces the control input $u(t)$, which manages the controlled system to achieve the desired output $y(t)$. The diagram highlights the two layers of control methodology.
ADAPTIVE CONTROL—MIAC
MODEL IDENTIFICATION ADAPTIVE CONTROL

Two layers

MODEL PREDICTIVE CONTROL (MPC)

- Two-level controllers like controllers for adaptive control
- Model predictive controllers rely on dynamic models of the managed system
- Most often linear empirical models obtained by system identification
- The main advantage of MPC is the fact that it allows the current timeslot to be optimized, while taking future time slots into account
- Optimize a finite time-horizon, but only realize the current timeslot
- MPC has the ability to anticipate future events and can take control actions accordingly
- Generic PID controllers do not have predictive abilities
CONTROL AND SYSTEMS SCIENCE FOR CPS

• Abstraction infrastructure to bridge digital and physical system components
  • Integrated and networked control
  • Adaptive and predictive control
  • Composition of control
  • Reference models

• Characterizing problems & guaranteeing solution quality
  • Utility function policies and optimization
  • Uncertainty characterization and quantification

• Assurance at runtime
  • Models at runtime
  • V&V at runtime
HOW SHOULD WE TEACH THE CONCEPTS OF HIGHLY DYNAMICAL SOFTWARE SYSTEMS IN THE AGE OF CONTEXT?

How do we integrate these topics into computing science and software engineering curricula?
Contents at a Glance

Preface

Part 1 Introduction to Software Engineering
Chapter 1 Introduction
Chapter 2 Software processes
Chapter 3 Agile software development
Chapter 4 Requirements engineering
Chapter 5 System modeling
Chapter 6 Architectural design
Chapter 7 Design and implementation
Chapter 8 Software testing
Chapter 9 Software evolution

Part 2 Dependability and Security
Chapter 10 Sociotechnical systems
Chapter 11 Dependability and security
Chapter 12 Dependability and security specification
Chapter 13 Dependability engineering
Chapter 14 Security engineering
Chapter 15 Dependability and security assurance

Part 3 Advanced Software Engineering
Chapter 16 Software reuse
Chapter 17 Component-based software engineering
Chapter 18 Distributed software engineering
Chapter 19 Service-oriented architecture
Chapter 20 Embedded software
Chapter 21 Aspect-oriented software engineering

Part 4 Software Management
Chapter 22 Project management
Chapter 23 Project planning
Chapter 24 Quality management
Chapter 25 Configuration management
Chapter 26 Process improvement
We need a new discipline

SOFTWARE ENGINEERING @ RUNTIME
SOFTWARE ENGINEERING @ RUNTIME

- Profound impact on SE and CS
- Rethink software design and evolution for highly adaptive software systems
- Feedback loops and control theory are key

Boundary between development-time and run-time is disappearing

EXERCISE

Convince each other why software engineers and computer scientists should take a course in control theory?
Control science can be defined as a systematic way to study certifiable V&V methods and tools to allow humans to trust decisions made by self-adaptive smart systems.
BOOKS TO PROBE FURTHER
BOOKS THAT PROFOUNDLY INFLUENCED US

  http://dspace.library.uvic.ca/bitstream/handle/1828/4476/Villegas_Norha_PhD_2013.pdf?sequence=6&isAllowed=y
  http://link.springer.com/book/10.1007/978-3-642-35813-5/page/1#
  http://dspace.library.uvic.ca/bitstream/handle/1828/4476/Villegas_Norha_PhD_2013.pdf?sequence=6&isAllowed=y
- Chignell, Cordy, Kealey, Ng, Yesha (Eds.): The Personal Web: A Research Agenda, LNCS 7855, Springer, pp. 151-184 (2013)
  http://www.sei.cmu.edu/uls
  http://books.google.ca/books
  http://shop.oreilly.com/product/0636920028970.do
CPS PAPERS

• NIST Cyber-Physical Systems Program: http://www.nist.gov/cps/


IOT PAPERS

IOT PAPERS


• IBM Corp.: How to compete in the era of “smart” (2014)  
http://www.ibm.com/smarterplanet/

• IBM Corp.: The Internet of Things, YouTube Video (2010)  
http://www.youtube.com/watch?v=sfEbMV295Kk

http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=6512846


• Industrial Internet Consortium (with over 150 member companies)  
http://www.iiconsortium.org

http://research.gold.ac.uk/5641/

• Bradley, et al.: Internet of Everything (IoE)— A $4.6 Trillion Public-Sector Opportunity, CISCO TR (2014)  
IEEE COMPUTER SOCIETY ELECTIONS

IEEE Computer Society Board of Governors Election
• [http://www.computer.org/web/volunteers/bog/](http://www.computer.org/web/volunteers/bog/)
• Hausi Müller is running for 1\textsuperscript{st} Vice President of IEEE Computer Society

IEEE CS TCSE Election
• Marin Litoiu is running for TCSE Chair

ありがとうございます

Thank you
TAKE HOME MESSAGES

• Take on a leadership role in CPS research
• Software engineering at runtime
• Control and systems science for CPS
• Hierarchical and networked control
• Assurance at runtime

• Vote for Marin and Hausi in IEEE CS Elections

ありがとうございます
Thank you