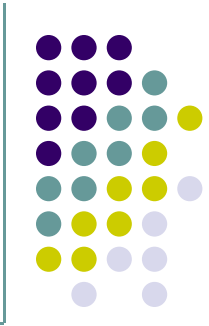
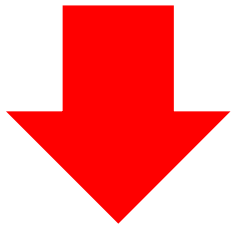


Welcome to SENG 480B / CSC 485A / CSC 586A Self-Adaptive and Self-Managing Systems

Dr. Hausi A. Müller
Professor and Associate Dean Research
Department of Computer Science
University of Victoria



http://courses.seng.uvic.ca/courses/2015/summer/seng/480a
http://courses.seng.uvic.ca/courses/2015/summer/csc/485a
http://courses.seng.uvic.ca/courses/2015/summer/csc/586a

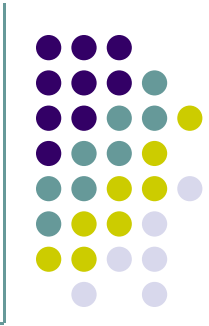
Outstanding TAs

Lorena

Nina

Ron

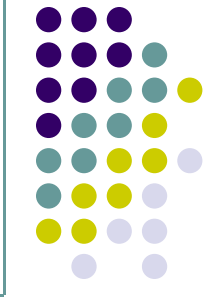
Deadlines and Course Requirements



Unit	Undergrads Weight	Grads Weight	Remarks
A1	12%	9%	Due Fri, May 29, 2015
A2	12%	9%	Due Fri, June 19, 2015
A3	12%	9%	Due Fri, July 10, 2015
A4	12%	9%	Due Fri, July 31, 2015
Grad Project		12%	Due Sat, July 25, 2015
Participation and presentation	7%	7%	Only graduate students are required to give a presentation towards the end of the course.
Midterm 1	20%	20%	June 4, 2015 in class. Closed books, closed notes, no phones, no computers, no calculators, no gadgets.
Midterm 2	25%	25%	July 16, 2015 in class. Closed books, closed notes, no phones, no computers, no calculators, no gadgets.
Total	100%	100%	Have a great course!

- All materials discussed in class are required for the midterm examinations
- Completing all midterms and assignments is required to pass the course
- Passing the midterms is not absolutely required to pass the course, but of course highly recommended

Assignment 1 Part I



Instructions

This assignment consists of three parts. In Part I you are to characterize four feedback systems. In Part II you are to deepen your understanding of ULS systems. In Part III is a group assignment on sensor APIs.

Part I

Identify three (3) feedback systems from different application areas that you encounter in your everyday life. For each system, identify the type of feedback (e.g., positive, negative, or bipolar), identify the sensing and actuation mechanisms as well as the algorithm used in the controller. Describe in detail the underlying model and its assumptions. Describe the uncertainty that the feedback system provides. Describe the dynamics that are controlled through the use of feedback. At least two of the three examples should be software-intensive systems. Graduate students are strongly encouraged to pick at least one system from their research area.

Recommended reading materials

- Murray: Control in an Information Rich World: Report of the Panel on Future Directions in Control, Dynamics, and Systems. SIAM 2003. http://www.cds.caltech.edu/~murray/cdspanel/report/cdspanel-15aug02.pdf
- Chapters 1 & 2

The answer for each feedback system should fit onto approximately one typeset page.

Maximum 3 pages for this part

Do not copy verbatim from any source. Write your own prose.
Cite your sources.

Assignment 1 Part II

Part II

Study the following book on Ultra-Large-Scale Systems (ULS)

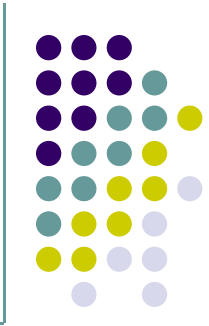
- Northrop, L., Feiler, P., Gabriel, R., Goodenough, J., Linger, R., Longstaff, T., Kazman, R., Klein, M., Schmidt, D., Sullivan, K., Wallnau, K.: Ultra-Large-Scale Systems. The Software Challenge of the Future. Technical Report, Software Engineering Institute, Carnegie Mellon University, 134 pages ISBN 0-9786956-0-7 (2006)
- http://www.sei.cmu.edu/uls

1. What are the main characteristics of a ULS system?
2. Contrast centralized and decentralized control.
3. Describe two selected challenges for the design and evolution of ULS systems in detail.

Maximum 2 pages for this part

Do not copy verbatim from any source.
Cite your sources.

Assignment 1 Part III



Part III - Group Project (4-5 people per group)

1. Identify and describe sensor APIs for different platforms (e.g., different operating systems). Pick an interesting category of sensors or sensor network and describe its API in detail.
2. Design, implement and document a simple application using this API.
3. Describe how this API and your application can be transitioned to a cloud computing environment.

All group members have to work on all three parts together. Learn from each other!
Articulate how the individual group members contributed to Part III.

Submission details:

- **Maximum** 3 pages for this part
- Submit a short video of no more than 5 minutes explaining your implementation and showing a demo of your application. **NOTE**: It is recommended that you upload the video in some external repository (e.g., Dropbox or Google Drive) and submit the access link.

You only need to submit one document and video per group.
Do not copy verbatim from any source.
Cite your sources.

See SensorCloud video later in today's lecture

Groups

G1	
G2	

Groups for A1 Part III

G1	Ernest Aaron, Adel Adil, Luuk Veenis
G2	Maryi Arciniegas Mendez, Navdeep Bahia, Richard Wagner
G3	Simar Arora, Miles Barr, Evan Wilde
G4	Fei Chen, David Bayly, Elliot Wyman
G5	Shu Chen, Jonathan Bowen, Gueorgui Zahariev
G6	Harsh Dawar, Joshua Braidek, Zhenyu Zhang
G7	Erkan Ersan, Po An Chen, Pufan Zheng
G8	Maria Ferman Guerra, Jason Cho, Zhanxue Zhu
G9	Khushboo Gandhi, Sebastian Craig, Mohammad Bin Abdulsalam
G10	Harneet Kaur, Yongjun Xu, Timothy Dalton,
G11	Francis Harrison, Meric Demiriz, Jorge Conde Gomez Llanos
G12	Stephan Heinemann, Sean Debroni, Tory Borsboom-Hanson
G13	Harshit Jain, Rodney Gelera, John Cox
G14	Sumit Kadyan, Alice Gibbons, Fraser DeLisle
G15	Navpreet Kaur, Zhuoli Xiao, Adnan Duale
G16	Parminder Kaur, Abhinandan Jagdev, Dustin Faulkner
G17	Waseem Khan, Anthony Kohan, Xiaotian Li
G18	Nishant Khanna, Patrick Lavoie, Connor McConkey
G19	Akshay Khot, Junru Yang, Ian Leslie
G20	Carlene LeBeuf, Siqi Li, Darren Prince
G21	Ye Liang, Ran Wei, Bowen Liu, Simon Taft
G22	Junnan Lu, Colum McClay, Jiashu Xiong
G23	Adithya Rathakrishnan, MacKay McGillivray, Shuo Yen Yu
G24	Arturo Reyes Lopez, Brenda McPhail, Mohammed Nader Zuhri Yafi
G25	Babak Tootoonchi, Daniel Oon, Muhammad Zubair
G26	Sam Wang, Hernan Rossi, Samuel Navarrete
G27	Stephan Wasylishen, Victoria Sahle, Sahibdeep Sran

Email will be disseminated shortly

If you are uncomfortable about disseminating your email to the class list send me a note

Terms to study on the web

- Internet of things
- Industrial Internet (GE)
- Cyber-physical systems
- Ultra-large-scale systems
- Digital ecosystem
- Wearable computers
- The age of context
- Context awareness
- Situational awareness
- Big data
- Big data analytics
- Ubiquitous computing
- Pervasive computing
- Cloud computing
- Green computing
- Sensors and actuators
- Smart systems
- Google driverless car
- Google glass
- Microsoft Hololens
- iRobot
- Quadcopters

Self-Adaptive Systems (SAS)

- A SAS can alter its behaviour at runtime (on the fly) in response to its perception of
  - its environment
  - its own state

  by adapting itself

- SAS abilities
  - Assess its own behaviour
  - Observe its context or environment
  - Adapt without shut down

- Oreizy, et al.: An Architecture-Based Approach to Self-Adaptive Software, *IEEE Intelligent Systems,* pp. 54-62 (1999)
- MacManus: Why Software is More Important Than Sensors in the Internet of Things, ReadWriteWeb (2010)

Situational Awareness (SA)

- SA is the perception of environmental and personal context with respect to time and space
- Comprehension of its meaning and its projection into the future
- Critical to decision-making in complex, dynamic situations

- Applications
  - Mars Curiosity
  - Aviation—UAV, drones
  - Military command and control
  - Emergency services

- Applications
  - Driving a car
  - Crossing a street
  - Playing soccer
  - Playing basketball
  - Shopping

Lionel Messi

Situation Awareness in Sports

Jordan who? McCabe

Adapting on the Fly— at Runtime

Intuitively we know how critical and valuable context is.
But context is complicated.

"Context is the new battleground between
Android, iOS, Windows, Symbian and
Apple, Google, IBM, Microsoft, Nokia, Samsung."

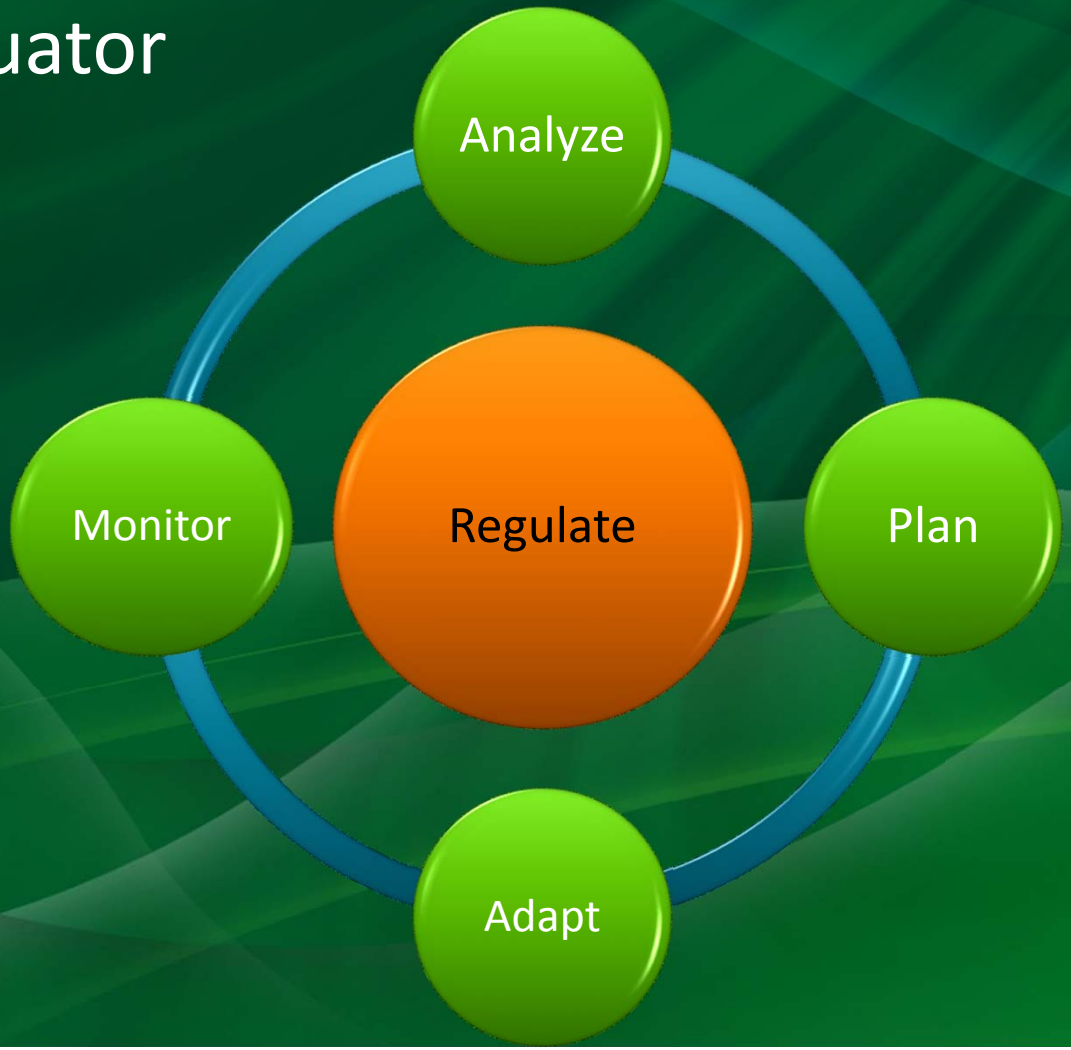
The Age of Context

Simple can be harder than complex. You have to work hard to get your thinking clean to make it simple.

Steve Jobs, BusinessWeek, 1998

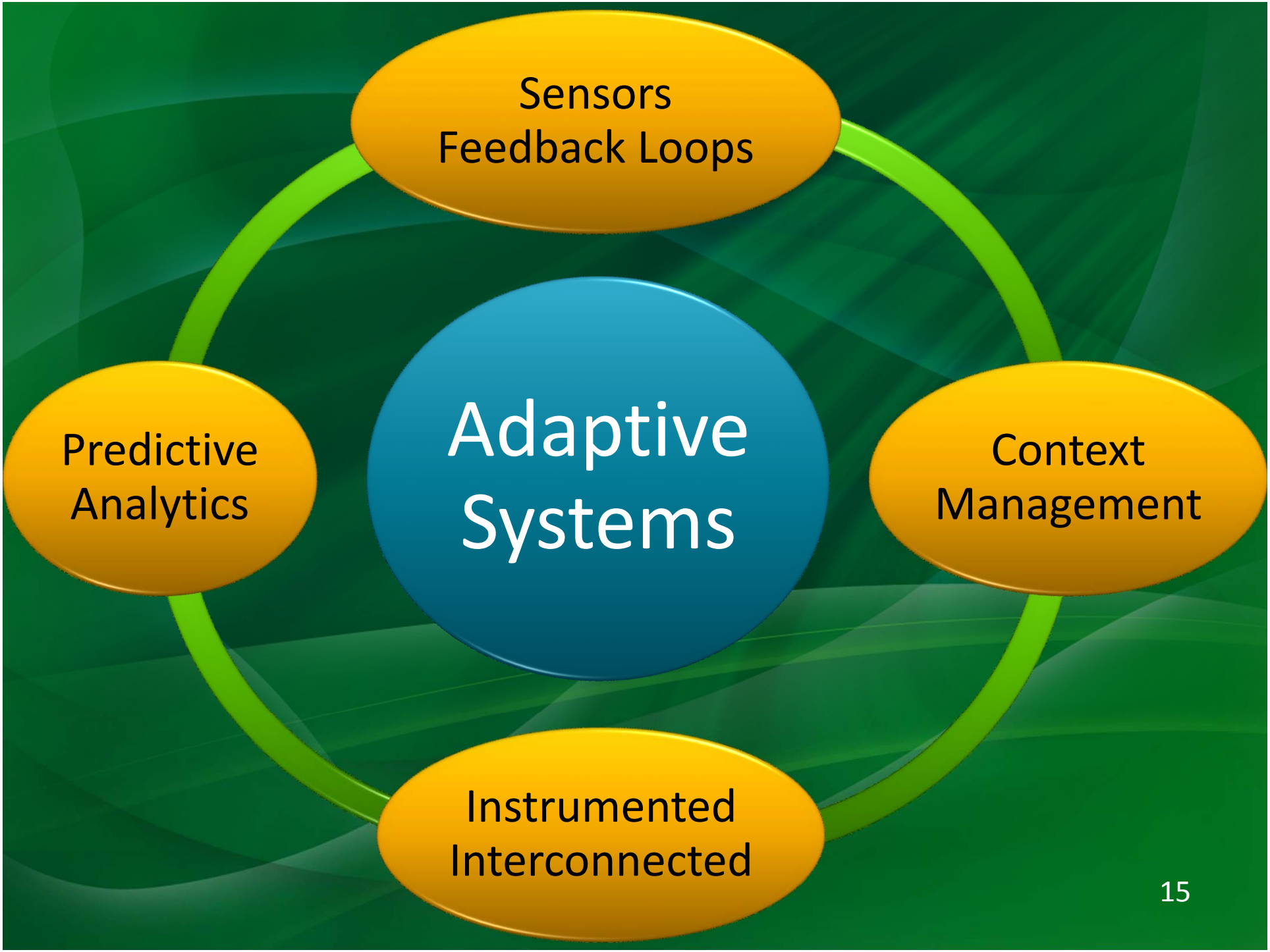
Adapt on the Fly—at Runtime

- Sensors give us the ability to monitor things
- Depending on the measures, things can adapt via actuator

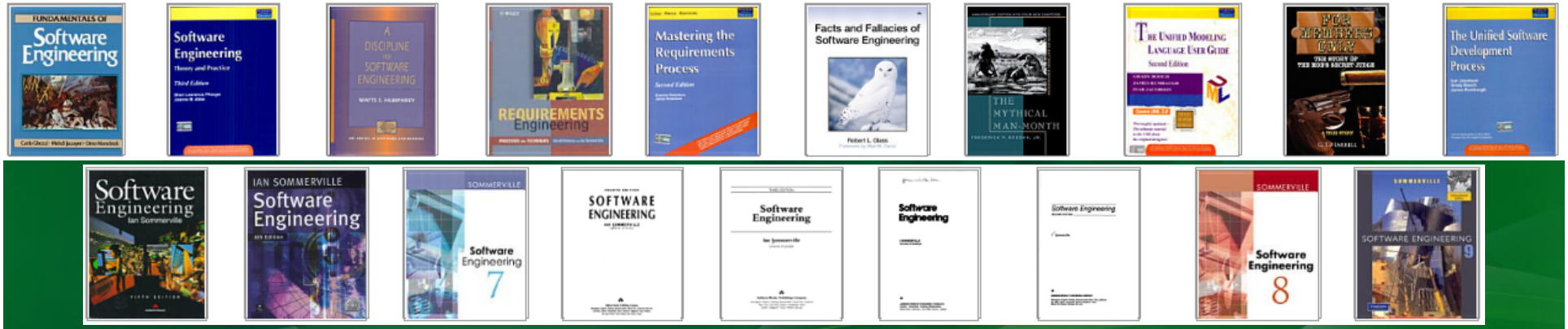


Smarter System Characteristics

Smarter systems adapt at runtime

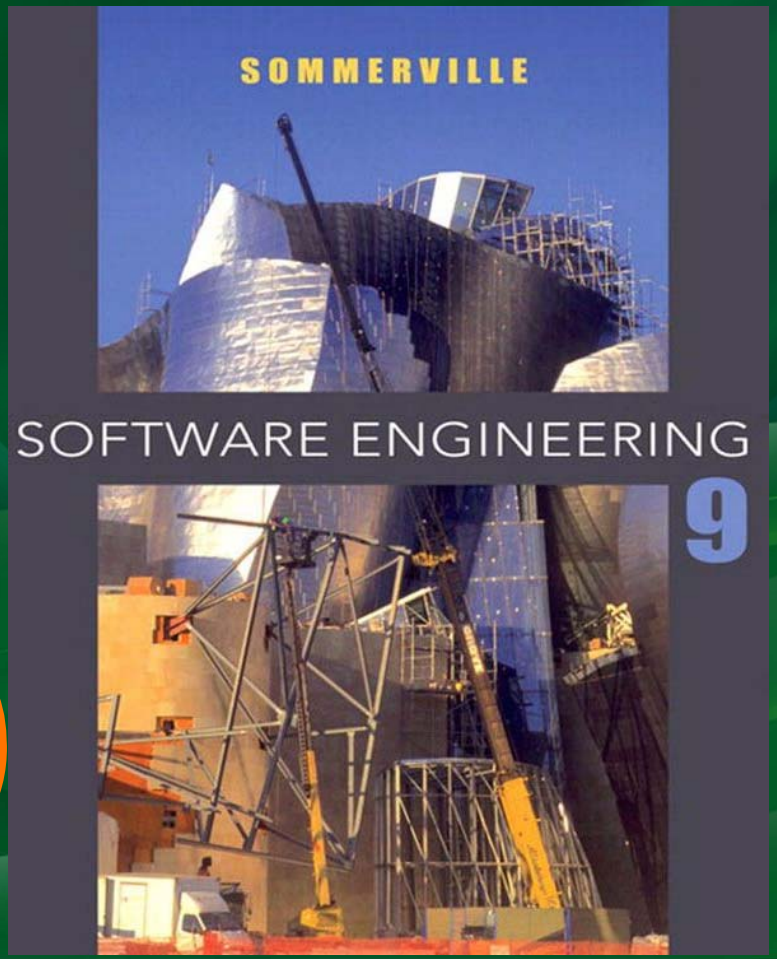

Sensors
Feedback Loops
Predictive
Analytics
Adaptive
Systems
Context
Management
Instrumented
Interconnected

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?

Ian Sommerville
Software Engineering
9th Edition
2010



Contents at a Glance

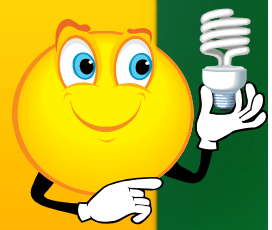
We need a new discipline

Software Engineering @ Runtime

Software Engineering @ Runtime

- Requirements@runtime
- Models@runtime
- Monitoring@runtime
- V&V@runtime
- Adaptation@runtime
- Analysis@runtime
- CM@runtime
- Assurance@runtime

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

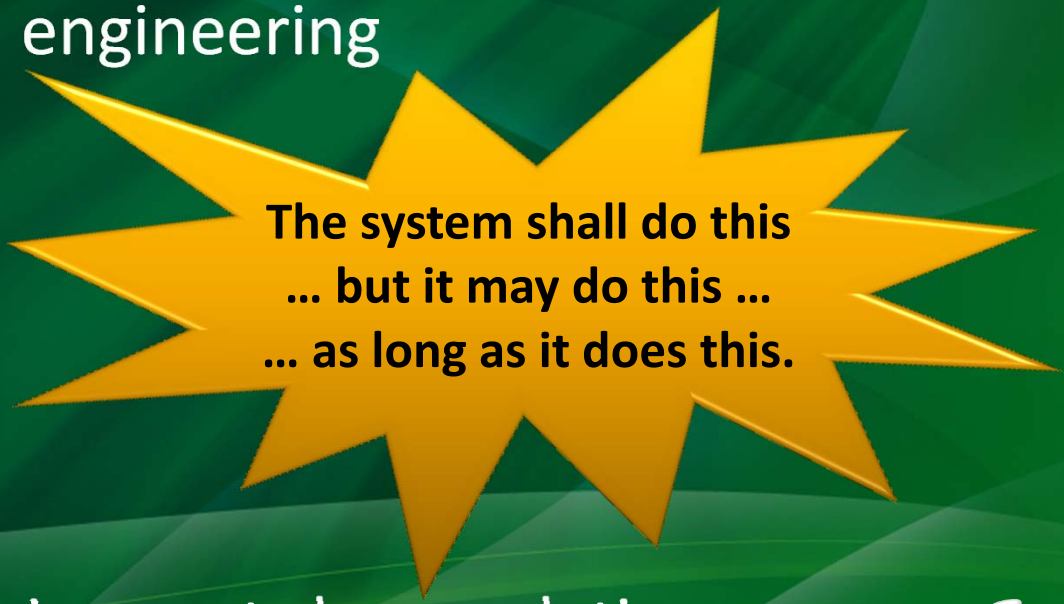


Boundary between development-time and run-time is disappearing

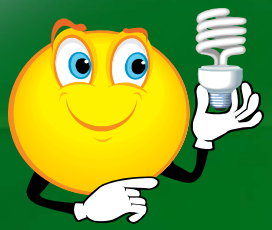
- Baresi, Ghezzi: The disappearing boundary between development-time and run-time. In: *FSE/SDP Workshop on Future of Software Engineering Research (FoSER 2010),* pp. 17-22 (2010)

Requirements @ Runtime

- **From** satisfaction of requirements through traditional, top-down engineering



- **To** satisfaction of requirements by regulation of complex, decentralized systems



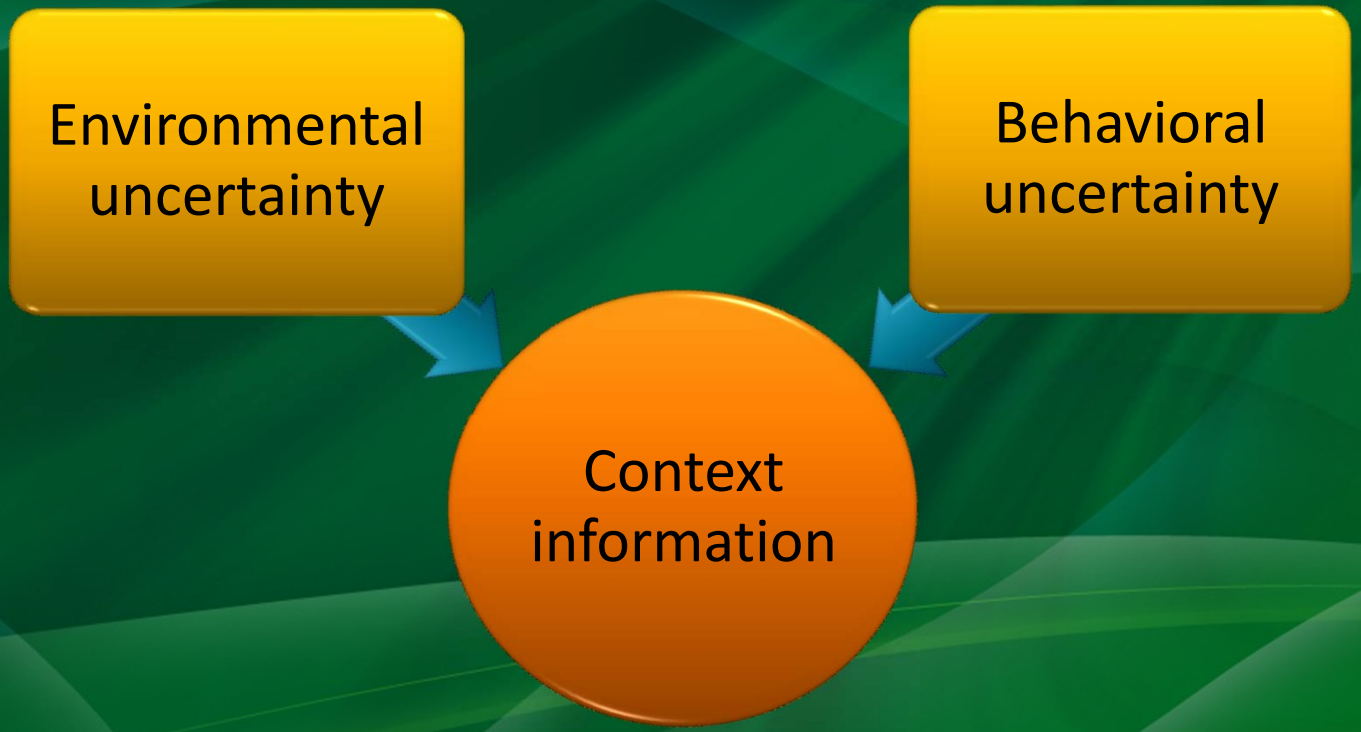
How much environment uncertainty can we afford? What's the cost?
What benefits do we accrue by accommodating context uncertainty?

Models @ Run.time

- Runtime model representation and management
- Models @ Run.time need reflection
- Goal models for NF requirements
- Runtime verification of statecharts
- Dynamic context models
- UML behavioral models at runtime
- Applying MDE tools at runtime
- GUI runtime adaptation models
- Model synchronization
- Models for security analysis

- Bencomo: Workshop Series on Models@run-time, http://www.comp.lancs.ac.uk/~bencomo/WorkshopMRT.html
- Bencomo: Workshop Series on Requirements@run.time, http://www.comp.lancs.ac.uk/~bencomo/RRT/
- Dagstuhl Seminar: Models@run.time, 2011 http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=11481

Context Models @ Runtime



- Coutaz, Crowley, Dobson, Garlan: Context is key, *CACM* 48(3) (2005)
- Whittle et al.: RELAX: A language to address uncertainty in self-adaptive systems requirements, *Requirements Engineering* 15(2):177-196 (2010)
- Inverardi, Mori: Feature-oriented evolutions for context-aware adaptive systems. In: *Proc. IWPSE-EVOL,* pp. 93-97, (2010)

Make Context First Class

Context representation

- Modeling of relevant context
- Context management strategies
- Adaptation of context models at runtime

Context management

- Adaptive context management strategies
- Gathering, provisioning
- Context reasoning

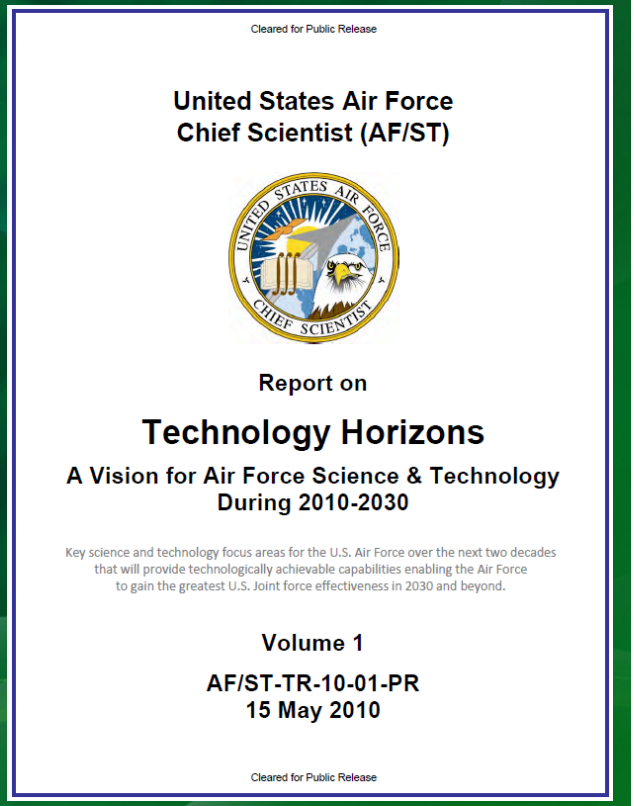
Make feedback loops first class

Autonomic control loop

- Müller, Pezzè, Shaw: Visibility of control in adaptive systems, *Proc. Second Int. Workshop on Ultra-Large-Scale Software-Intensive Systems (ULSSIS 2008)*, pp. 23-26 (2008)

Assurance @ Runtime
Make V&V @ Runtime First Class

- V&V ensures that software satisfies requirements and quality attributes
- Runtime V&V ensures proper system operation during adaptation
- Certifiable V&V methods are critical for smart systems


Cleared for Public Release

United States Air Force
Chief Scientist (AF/ST)

Report on

Technology Horizons

A Vision for Air Force Science & Technology During 2010-2030

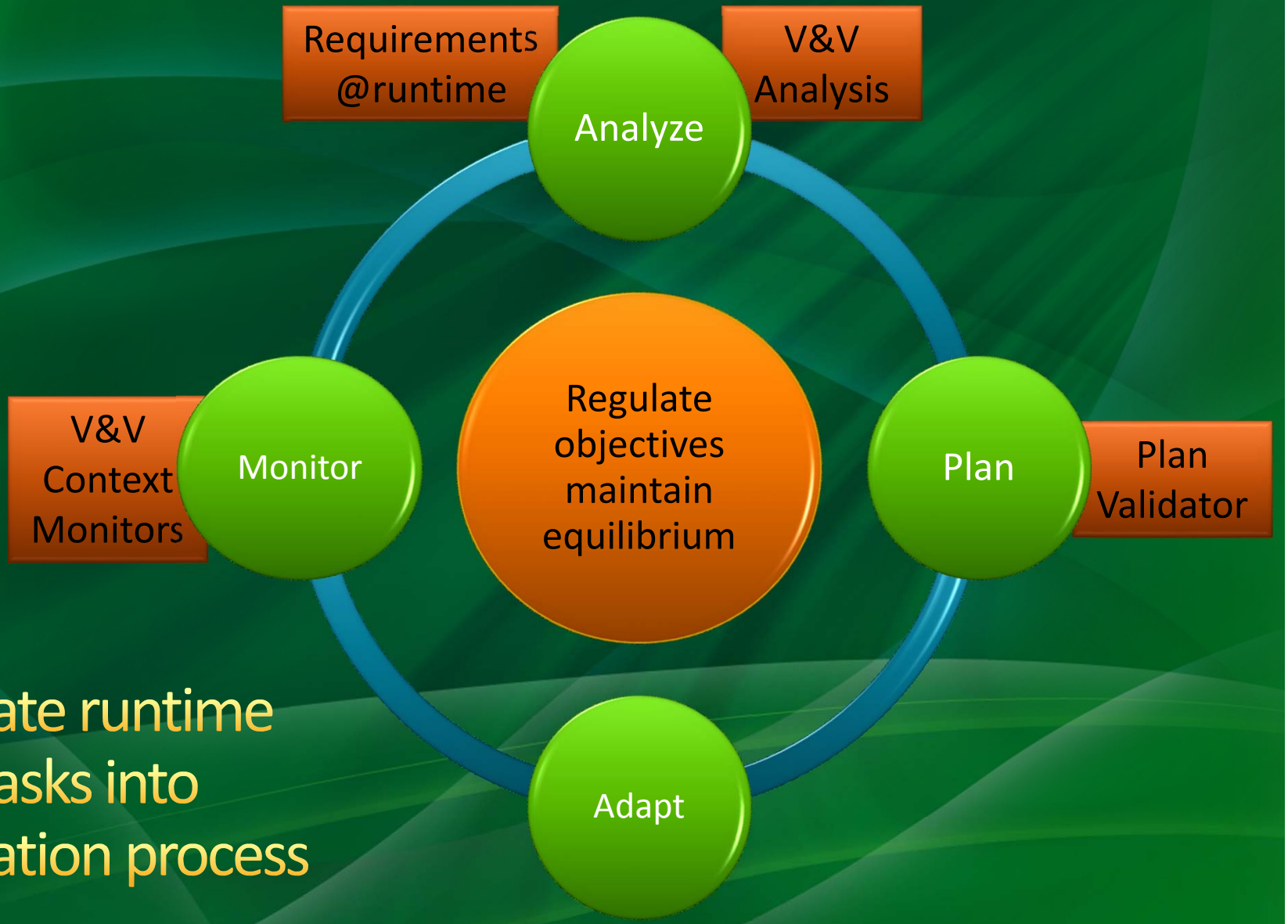
Key science and technology focus areas for the U.S. Air Force over the next two decades that will provide technologically achievable capabilities enabling the Air Force to gain the greatest U.S. Joint force effectiveness in 2030 and beyond.

Volume 1

AF/ST-TR-10-01-PR
15 May 2010

Cleared for Public Release

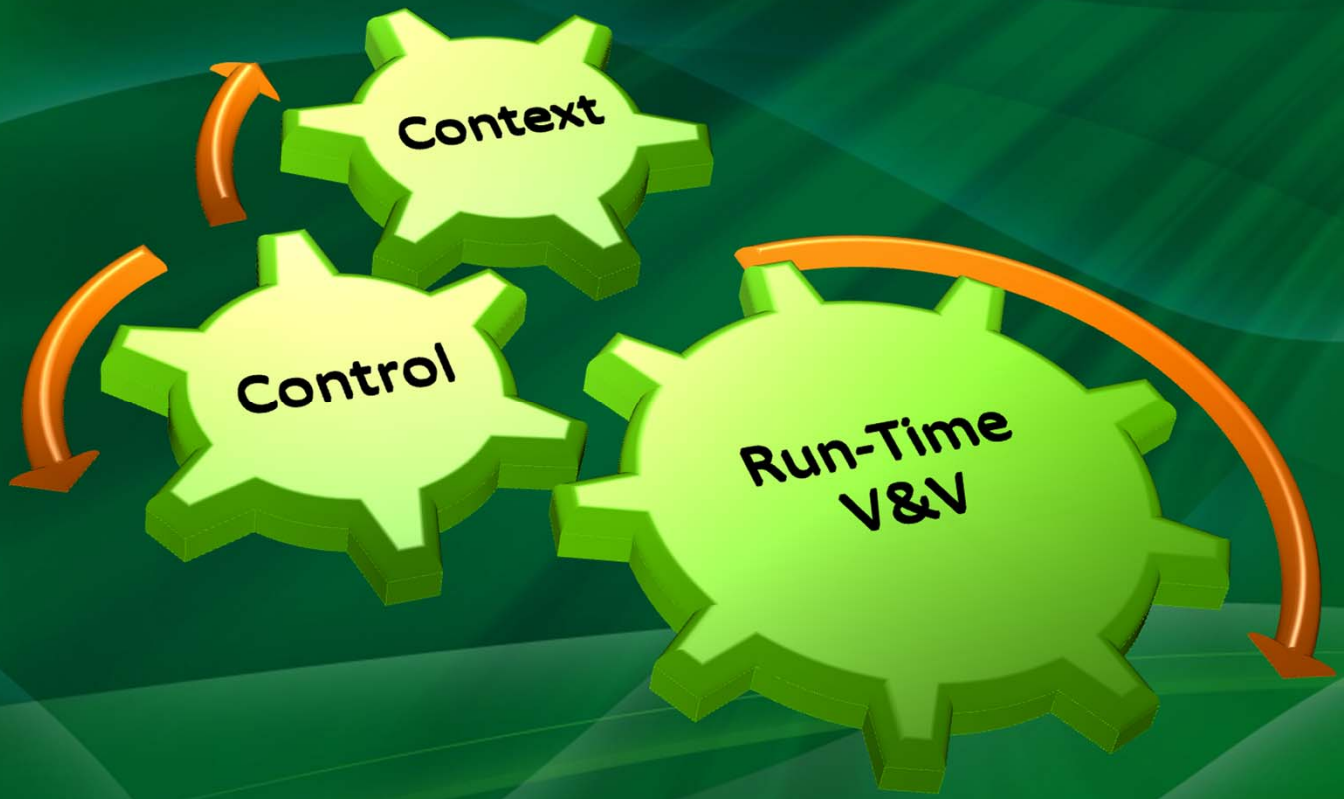
- Dahm: Technology Horizons: A Vision for Air Force Science & Technology During 2010-2030. TR USAF (2010)
- Villegas, et al.: A Framework for Evaluating Quality-Driven Self-Adaptive Software Systems, In: *Proc. 6th ACM/IEEE Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2011),* pp. 80-89 (2011)
- Tamura, Villegas, Müller, et al.: Towards practical runtime verification and validation of self-adaptive software systems. In: de Lemos, Giese, Müller, Shaw (Eds.), *Software Engineering for SAS,* Springer (2012)



Integrate runtime V&V tasks into adaptation process

- Tamura, Villegas, Müller, et al.: Towards practical runtime verification and validation of self-adaptive software systems. In: de Lemos, Giese, Müller, Shaw (Eds.), *Software Engineering for SAS,* Springer (2012)

Control Science



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.

The Internet of Things (5 minutes)

http://www.youtube.com/watch?v=sfEbMV295Kk