Welcome to SENG 480A / 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

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

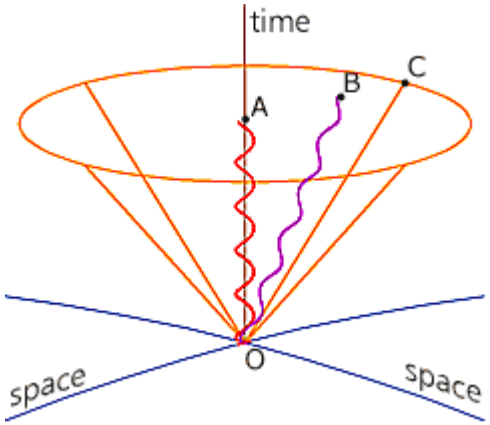
- SAS abilities
  - Assess its own behaviour
  - Observe its context or environment
  - Adapt without shut down

- 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

- Perception of the environment within a volume of space and time
- Comprehension of their meaning
- Projection into the future

—Endsley 1999



Bombardier Challenger 300

Survival

Situational Awareness

Situational Awareness

India

Mindboggling Situational Awareness

Shibuya, Tokyo

Humans are amazingly adaptive

Stream of Context

Instrument and Capture the Stream of Context

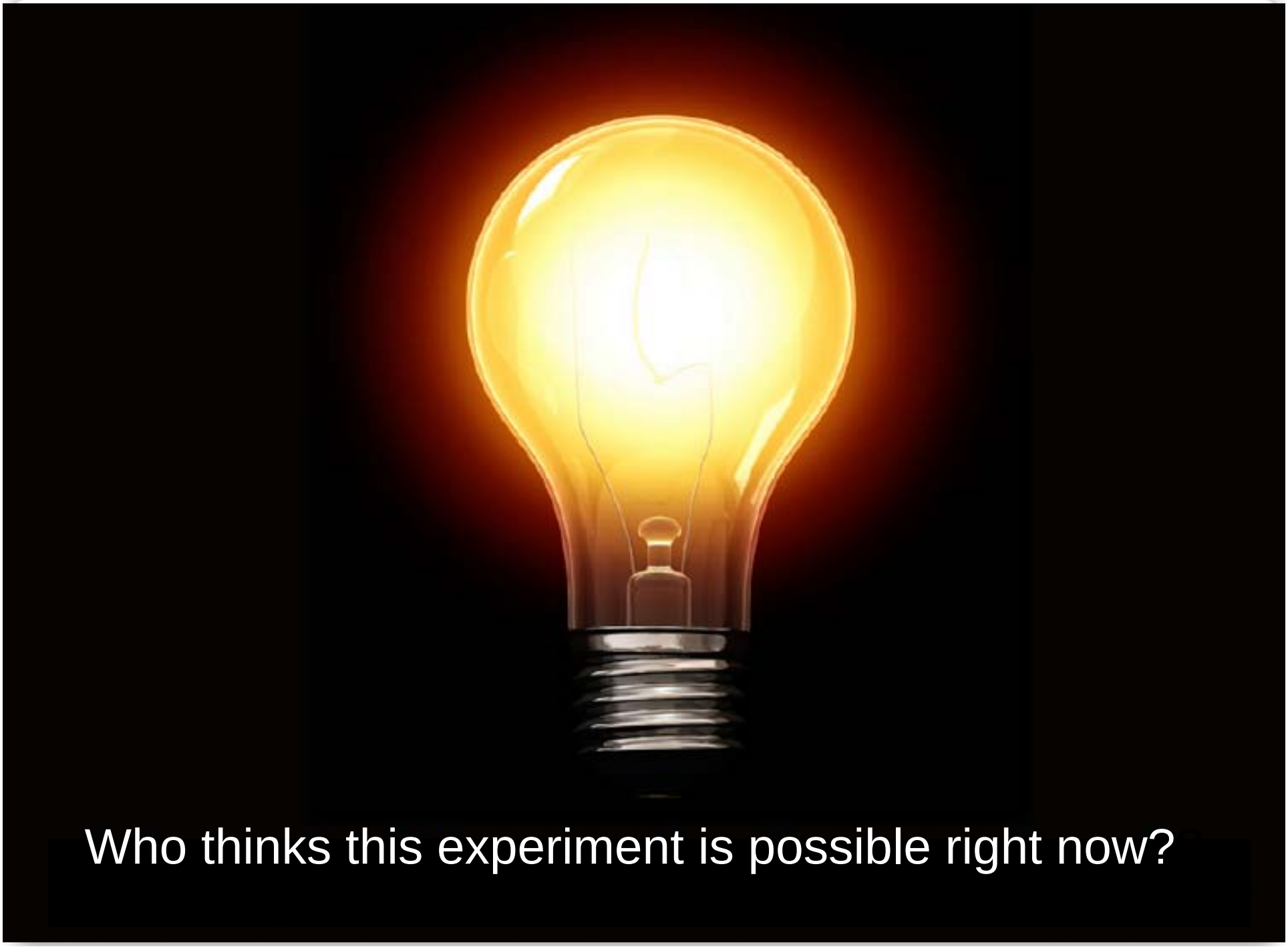
Google

Stream of Context

Instrument and Capture the Stream of Context

Killer Application

The Experiment—Volunteers

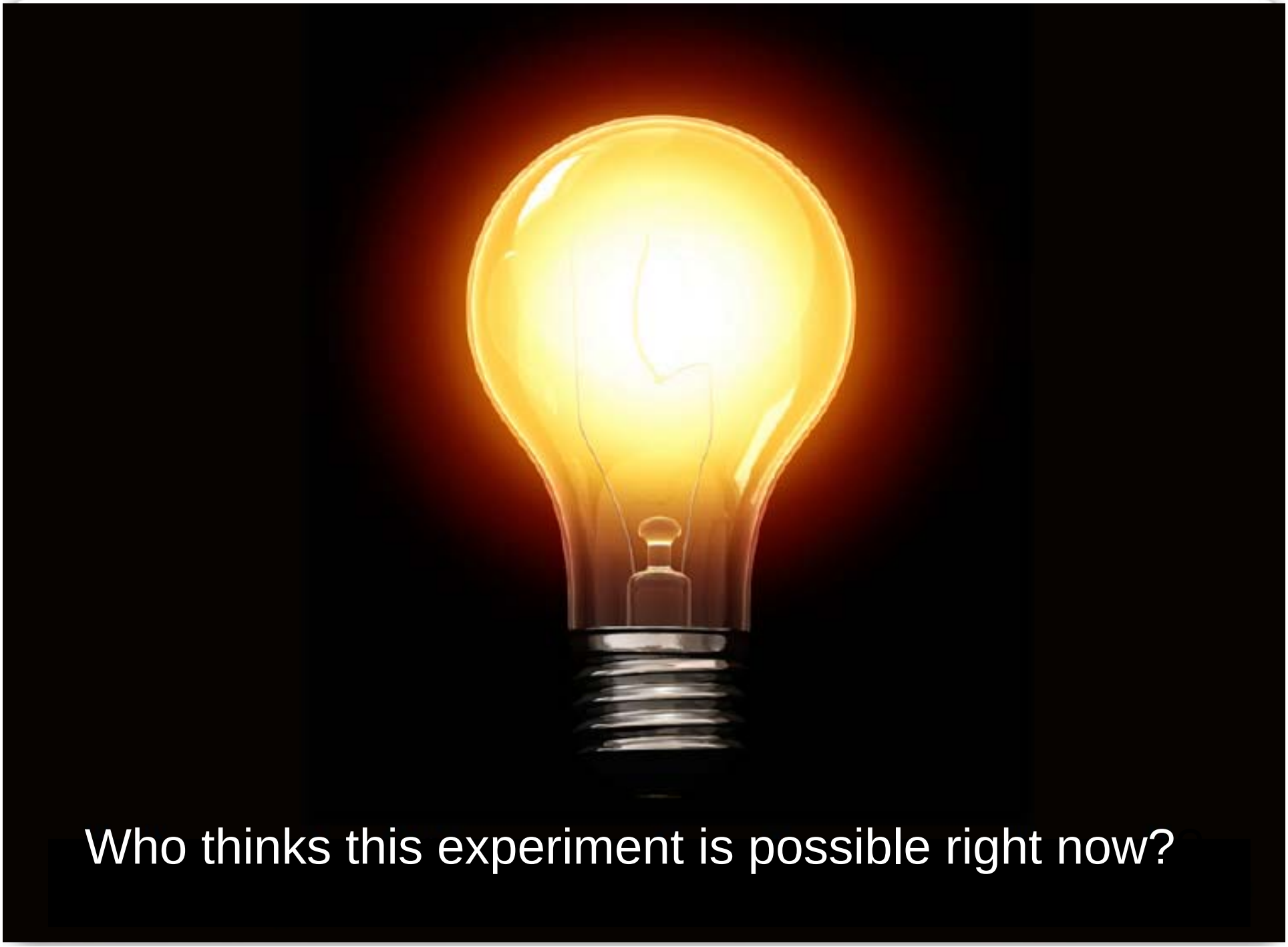

Who thinks this experiment is possible right now?

First Class Participation Assignment

- Form groups of 2-4 people and discuss the following
- Define the SASs in the traffic examples
  - What is the environment of each SAS?
  - What is its own state and how does it change?
- From your own experience
  - What are your own abilities to handle such situations?
  - What environmental context and what of your own context is used in such a situation?
  - How do you actually adapt in such a situation?

- A SAS can alter its behaviour at runtime in response to its perception of
  - its environment
  - its own state

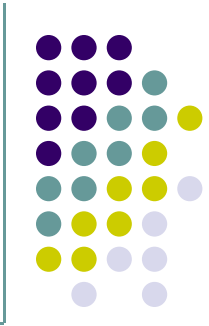
  by adapting itself
- SAS abilities
  - Assess its own behaviour
  - Observe its context or environment
  - Adapt without shut down


Who thinks this experiment is possible right now?

Google Driverless Car — 2:07 mins

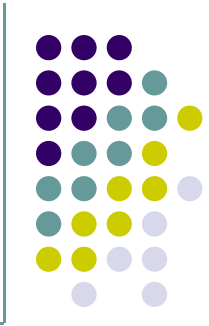
Licensed in Florida, Nevada, California

Course Description



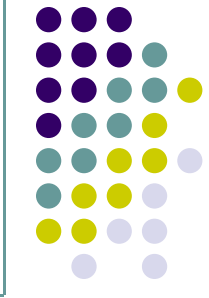
- The simultaneous explosion of information and integration of technology and the continuous evolution from software intensive systems to systems of systems to ultra-large-scale (ULS) systems requires new and innovative approaches for building, running and managing software systems.
- A consequence of this continuous evolution is that software systems must become more versatile, flexible, resilient, dependable, robust, continuously available, energy-efficient, recoverable, customizable, self-healing, configurable, or self-optimizing by adapting to changing contexts and environments.

Course Description



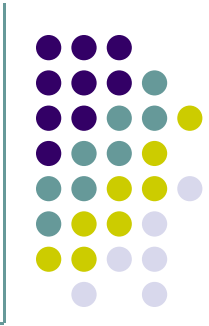
- One of the most promising approaches to achieving such properties is to equip software systems with self-adaptation and self-managing mechanisms.
- The topic of self-adaptive and self-managing systems has been studied in a variety of application areas, including autonomic computing, robotics, control systems, programming languages, software architectures, fault-tolerant computing, and biological computing.
- In this course, we focus on the software engineering aspects, including the methods, architectures, algorithms, techniques and tools support self-adaptive and self-managing behavior and exciting application areas, including autonomic computing and ULS systems.

Course Web Sites



- Course outline
  - Undergraduate students
    - http://courses.seng.uvic.ca/courses/2015/summer/seng/480a
    - http://courses.seng.uvic.ca/courses/2015/summer/csc/485a
  - Graduate students
    - http://courses.seng.uvic.ca/courses/2015/summer/csc/586a
- Course websites
  - http://www.rigiresearch.com/courses/sas
  - Syllabus
  - Lecture slides (pdf)
  - Assignments
  - Materials for reading assignments
  - Everything else you need to know about the course

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

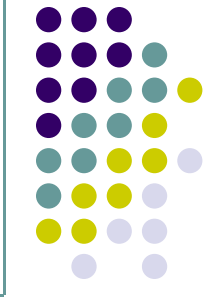
Syllabus Overview

- Dynamical software-intensive systems (1 week)
- ULS systems (2 weeks)
  - Reading Assignment Sections 1-3 of ULS Book
- Feedback control of computing systems (2 weeks)
  - Reading Assignment Hellerstein book/paper
- Autonomic systems (2 weeks)
  - Reading Assignment IBM Autonomic Blueprint & Kephart
- Self-adaptive systems (4 weeks)
  - Reading Assignment SE for Self-Adaptive Systems I & II

Prerequisites and Related Courses

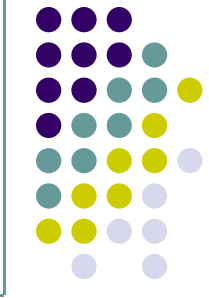
- Prerequisites (ideally, but not required)
  - SENG 371 Software Evolution
  - ELEC 360 Control Theory and Systems
  - Basics of software life cycle
  - Basics of software architecture

Optional Textbooks Great Resources



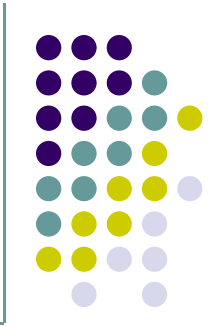
- Northrop, et al.: Ultra-Large-Scale Systems. The Software Challenge of the Future. Software Engineering Institute, Carnegie Mellon University, 134 pages ISBN 0-9786956-0-7 (2006) http://www.sei.cmu.edu/uls
- Hellerstein, Diao, Parekh, Tilbury: Feedback Control of Computing Systems. John Wiley & Sons (2004)
- Kephart, Chess: The Vision of Autonomic Computing. IEEE Computer 36(1):41-50 (2003)
- IBM Corp.: An Architectural Blueprint for Autonomic Computing, Fourth Edition (2006)

Optional Textbooks
Great Resources



- de Lemos, Giese, Müller, Shaw (Eds.): Software Engineering for Self-Adaptive Systems II, LNCS 7475, Springer (2013)
  http://link.springer.com/book/10.1007/978-3-642-35813-5/page/1#
- H.C. Cheng, R. de Lemos, P. Inverardi, J. Magee (Eds.): Software Engineering for Self-Adaptive Systems, LNCS 5525, Springer (2009)
  http://www.springer.com/computer/swe/book/978-3-642-02160-2
- More resources on course website

Assignments



- Reading assignment
  - ULS Book Section 1-3 on-line at
  - http://www.sei.cmu.edu/uls/
  - http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=30519

- Assignment 1
  - A1 will be posted by Thursday

What Is Class Participation?

- Students should be prepared to **speak** in class—it is completely acceptable, indeed encouraged, for students to give a mini-presentation on a relevant subject
- Class participation does not just mean signing in—attendance will be taken regularly
- Class participation means speaking up in class, both with questions and answers

- Note that 7% class participation almost corresponds to a full letter grade (up or down)

Questions?

- Organization of the course?
- Evaluation scheme?

- Study course web site carefully
- Visit course web site regularly
  - Web site and materials will change almost daily
- Other questions?!?

Keep in mind …

- **Ask questions at any time ☺ !! ☺**
- **Let’s make this a truly interactive course!!!**
- **Take full advantage of this opportunity to work on your communication skills ☺ !! ☺**