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

Dr. Hausi A. Müller
Professor
Department of Computer Science
University of Victoria

http://courses.seng.uvic.ca/courses/2013/summer/seng/480b
http://courses.seng.uvic.ca/courses/2013/summer/csc/485b
http://courses.seng.uvic.ca/courses/2013/summer/csc/586b

1

Quiz 3

- Are you sitting next to the same person you did on Tue?
- Did you look up any term or resource related to this course since Tue?
- This course involves a lot of reading! How much reading have you done so far?

2

Course Web Sites

- Course outline
  - Undergraduate students
    - http://courses.seng.engr.uvic.ca/courses/2010/spring/seng/480b
    - http://courses.seng.uvic.ca/courses/2013/summer/seng/480b
  - Graduate students
    - http://courses.seng.uvic.ca/courses/2013/summer/csc/586b
- 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

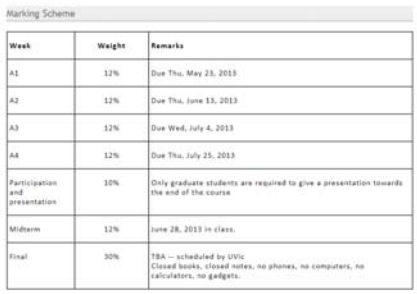
3

Course Website

http://www.rigiresearch.com/courses/sas

4

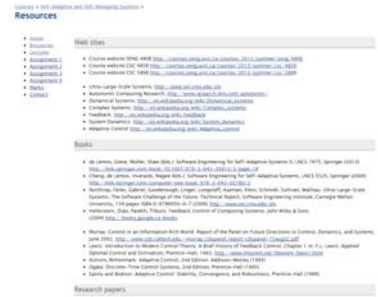
Course Website



http://www.rigiresearch.com/courses/sas

5

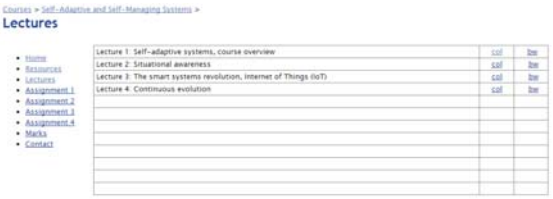
Course Website



http://www.rigiresearch.com/courses/sas

6

Course Website


Courses > Self-Adaptive and Self-Managing Systems >
Lectures

- Home
- Resources
- Lectures
- Assignment 1
- Assignment 2
- Assignment 3
- Assignment 4
- Marks
- Contact

Lecture 1: Self-adaptive systems, course overview	ppt	pdf
Lecture 2: Situational awareness	ppt	pdf
Lecture 3: The smart systems revolution, Internet of Things (IoT)	ppt	pdf
Lecture 4: Continuous evolution	ppt	pdf

http://www.rigiresearch.com/courses/sas

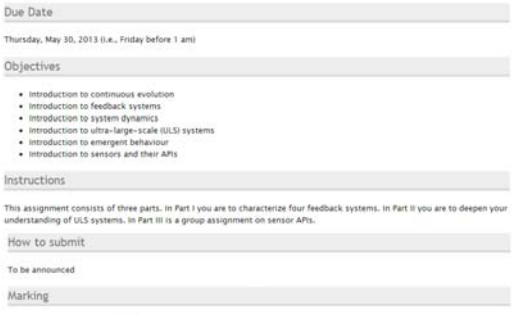
7

Assignments

- Reading assignment
  - ULS Book Section 1-3 on-line at
  - http://www.sei.cmu.edu/uls/the_report.html
    - 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
- Assignment 1
  - A1 will be posted by Wed

8

Assignment 1— Instructions


Due Date

Thursday, May 30, 2013 (i.e., Friday before 1 am)

Objectives

- Introduction to continuous evolution
- Introduction to feedback systems
- Introduction to system dynamics
- Introduction to ultra-large-scale (ULS) systems
- Introduction to emergent behaviour
- Introduction to sensors and their APIs

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.

How to submit

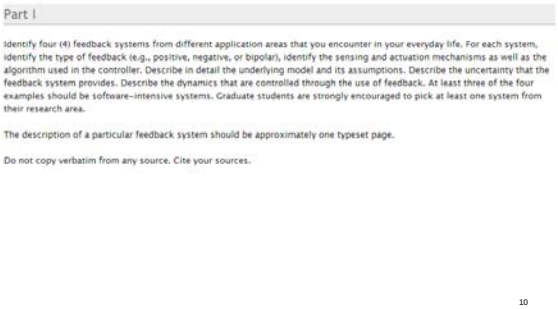
To be announced

Marking

Part I is worth 35%; Part II 35%; and Part III 30%.

9

Assignment 1 — Part I


Part I

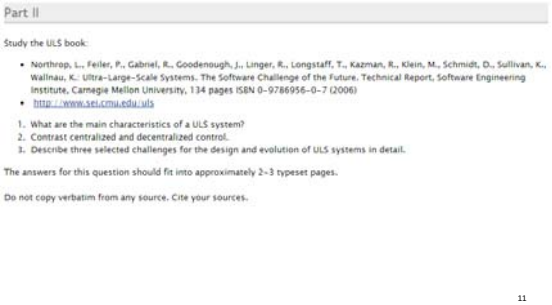
Identify four (4) feedback systems from different application areas that you encounter in your everyday life. For each system, identify the type of feedback (e.g., positive, negative, or bipolar), identify the sensing and actuation mechanisms as well as the algorithm used in the controller. Describe 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 three of the four examples should be software-intensive systems. Graduate students are strongly encouraged to pick at least one system from their research area.

The description of a particular feedback system should be approximately one typeset page.

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

10

Assignment 1 — Part II


Part II

Study the ULS book:

- 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

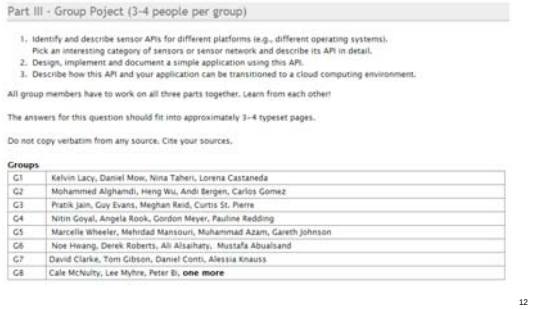
1. What are the main characteristics of a ULS system?
2. Contrast centralized and decentralized control.
3. Describe three selected challenges for the design and evolution of ULS systems in detail.

The answers for this question should fit into approximately 2-3 typeset pages.

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

11

Assignment 1 — Part III


Part III - Group Poject (3-4 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!

The answers for this question should fit into approximately 3-4 typeset pages.

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

Groups

G1	Kelvin Lacy, Daniel Mow, Nina Taheri, Lorena Castaneda
G2	Mohammed Alghamdi, Heng Wu, Andi Bergen, Carlos Gomez
G3	Pratik Jain, Guy Evans, Meghan Reid, Curtis St. Pierre
G4	Nitin Goyal, Angela Rook, Gordon Meyer, Pauline Redding
G5	Marcelle Wheeler, Mehrdad Mansouri, Muhammad Azam, Gareth Johnson
G6	Noe Hwang, Derek Roberts, Ali Alsaihaty, Mustafa Abualsand
G7	David Clarke, Tom Gibson, Daniel Conti, Alessia Knauss
G8	Cale McNulty, Lee Myhre, Peter Bi, **one more**

12

Deadlines

- Assignment 1
  - Thu, May 30 due
- Assignment 2
  - Thu, Jun 20 due
- Assignment 3
  - Thu, Jul 11 due
- Assignment 4
  - Thu, Jul 25 due
- Breaks
  - Reading Jun 4-11
  - Reading July 2
- Midterm
  - Fri, Jun 28
  - In class, closed books, closed notes
- Final
  - Aug 2013 to be scheduled by university
  - 3 hours, closed books, closed notes

13

Second Class Participation Assignment

- The execution environment for future software systems will not necessarily be known a priori at design time and, hence, the application environment of such a system cannot be statically anticipated.
- Such systems necessarily will have to reconcile the static view with the dynamic view by breaking the traditional division among development phases by moving some activities from design time to run time.

14

Second Class Participation Assignment

- The resulting systems push design decisions towards runtime and exhibit capabilities to reason about the systems' own state and environments.
- Discuss this problem and its issues in groups of 3-4 students and try to figure out what it all means
- Pick one person to present the findings to the class

15

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 basketball
  - Shopping

17

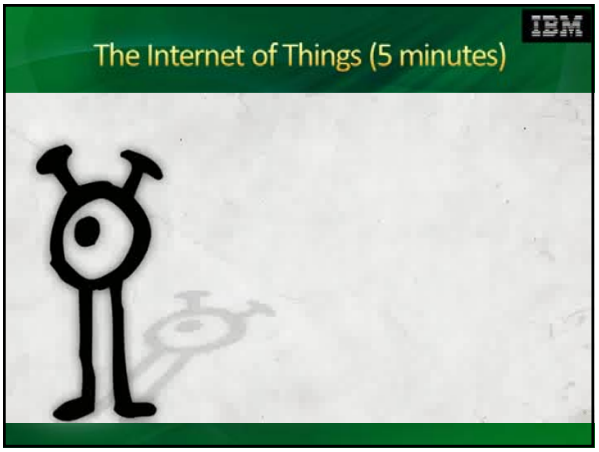
Intuitively we know how critical and valuable context is.
But context is complicated.

"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

18



Dynamical Software Systems

- Today, there are several research communities dealing with highly dynamical and evolving software-intensive systems
- The fundamental assumption
  - The execution environment for these systems will not be fully known a priori at design time—only be partially known
  - Thus, the application environment of such a system cannot be anticipated statically at design time
- One strategy to approach this problem
  - To reconcile the static view with the dynamic view by breaking the traditional division among software development phases and by moving some activities from design time to runtime
- What the approaches of different communities have in common is
  - To push design decisions towards runtime
  - To exhibit capabilities to reason about the system's own state and its environment
  - Different communities concentrate on different business goals and technological solutions

23

Continuous Evolution

- **Traditional (flawed) assumption:** software systems should
  - support organizational stability and structure
  - be low maintenance
  - strive for high degrees of user acceptance
- **Continuous evolution:** software systems
  - should be under constant development
  - can never be fully specified
  - are subject to constant adjustment and adaptation [Truex99]
- **Good news**
  - for the software engineering community (adaptive, autonomic, reverse engineering in particular) since this view guarantees research problems for years to come
- **Bad news**
  - most software engineering textbooks will have to be rewritten

Truex et al., Growing Systems in Emergent Organizations, CACM, 1999

Managing Tradeoffs

- 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?

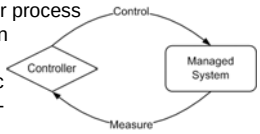
Independent Studies Confirm the Notion of Continuous Evolution

- **German SE Manifest [Broy06]**
  - Challenges for Software Engineering Research
- **ICSE 2006 Keynote [Boehm06]**
  - SE theses and antitheses for every decade from 1950 to 2020
  - He argues that "the ability of organizations and their products, systems, and services to compete, adapt, and survive will depend increasingly on software and on the ability to integrate related software-intensive systems into systems of systems."
- **SEI ULS [ULS06]**
  - Systems of systems are likely to evolve into Ultra-Large-Scale (ULS) socio-technical ecosystems
  - ULS ecosystems require a radically new perspective with respect to design and evolution, orchestration and control, as well as monitoring and assessment.
  - SEI study suggests that traditional top-down engineering approaches are insufficient to tackle the complexity and evolution problems inherent in decentralized, continually evolving software

26

Feedback Systems

- **Merriam-Webster's Online Dictionary**
  the return to the input of a part of the output of a machine, system, or process (as for producing changes in an electronic circuit that improve performance or in an automatic control device that provide self-corrective action)



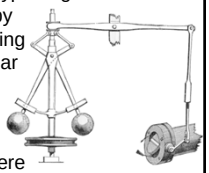
27

Feedback is ubiquitous
in natural and
engineered systems

28

Early Engineered Feedback System
Steam Engine with Governor

- A **centrifugal governor** is a specific type of governor that controls the speed of an engine by regulating the amount of fuel (or working fluid) admitted, so as to maintain a near constant speed whatever the load or fuel supply conditions. It uses the principle of proportional control.



- It was invented for steam engines where it regulates the admission of steam into the cylinder. Also internal combustion engines and striking clocks.
- James Watt designed his first governor in 1788 for steam engines, but never claimed the centrifugal governor to be an invention of his own.

29

Natural Feedback Systems

- **Biological Systems**
  - Physiological regulation (homeostasis)
  - Bio-molecular regulatory networks
- **Environmental Systems**
  - Microbial ecosystems
  - Pelagic and terrestrial ecosystems
  - Global carbon cycle
- **Financial Systems**
  - Markets and exchanges
  - Supply and service chains

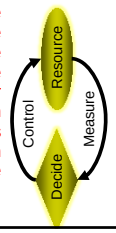
30

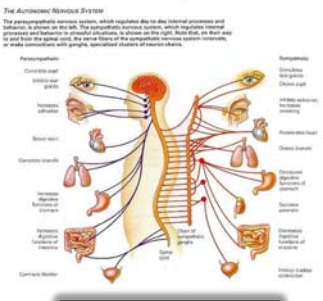
Most Famous Feedback System
Autonomic Nervous System (ANS)

Autonomic nervous system (ANS)

- Parasympathetic
  - Day-to-day internal processes
- Sympathetic
  - Stressful situation processes

Temperature
Heart rate
Breathing rate
Blood pressure
Blood sugar
Pupil dilation
Tears
Digestion
Immune response





Monitor and Regulate

31