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

Announcements

- Marking
  - A2 graded
  - Midterm graded
  - A3 being graded
  - Marks are posted on website
- Group presentations of A3
  - Excellent
  - One more today
- A4
  - posted
  - Due Thu, Aug 1
- Grad student presentations
  - July 23 – July 31
- Teaching Eval
- Review for final exam
  - Wed, Aug 7
- Last day of classes
  - Wed, Aug 7
- Final exam
  - Tue, Aug 13, 9:00-12:00 am in ECS 124
  - Closed books, closed notes
  - Materials: entire course
  - Format: like midterm

2

A2 Marking Guide

Part I (50 marks)

- Describe managed resource and describe its properties in detail: model, sensors, effectors. **(10 marks)**
- Define policy for managing the resource **(5 marks)**
- Defines events which are exchanged across the manageability interface. **(5 marks)**
- Design AM: all four stages. **(20 marks)**
- Identify information to be store in knowledge base. **(5 marks)**
- Describe the type of feedback that is used in the system: positive, negative, bipolar. **(5 marks)**

Part II (50 marks)

In Part II you are to implement an autonomic element consisting of the managed resource of your choice and an autonomic manager governed by a policy.

- Implement the managed resource you have chosen. **(5 marks)**
- Implement an autonomic manager to manage the resource. Code the four phases of the the MAPE-K loop and the knowledge as separate components. Make sure that the documents exchanged among the components are well defined. **(20 marks)**
- Implement a manageability interface to close the feedback loop between the managed resource and the autonomic manager. **(5 marks)**
- Make the autonomic manager policy driven. **(5 marks)**
- Demonstrate that your implementation is compliant with respect to the your chosen policy. **(5 marks)**
- Document the design and implementation of your project. **(10 marks)**

3

Graduate Student Research Paper Presentations

- Garlan, D., Cheng, S.-W., Huang, A.-C., Schmerl, B., Steenkiste, P.: Rainbow: Architecture-Based Self-Adaptation with Reusable Infrastructure. *IEEE Computer* 37(10):46-54 (2004) — **Angela Rook, July 23**
- Kramer, J., Magee, J.: Self-Managed Systems: An Architectural Challenge. In: *ACM /IEEE International Conference on Software Engineering 2007 Future of Software Engineering (ICSE),* pp. 259-268 (2007) — **Pratik Jain, July 23**
- Oreizy, P., Medvidovic, N., Taylor, R.N.: Runtime Software Adaptation: Framework, Approaches, and Styles. In: *ACM/IEEE International Conference on Software Engineering (ICSE 2008),* pp. 899-910 (2008) —**Alessia Knauss, July 24**
- Brun, Y., Di Marzo Serugendo, G., Gacek, C. Giese, H. Kienle, H.M., Litoiu, M., Müller, H.M., Pezzè, M., Shaw, M.: *Engineering Self-Adaptive Systems through Feedback Loops.* SE for Self-Adaptive Systems, pp. 48-70 (2009) — **Samra Ramandeep, July 24**
- Kaushik, R.T., Cherkasova, L., Campbell, R.H., Nahrstedt, K.: Lightning: self-adaptive, energy-conserving, multi-zoned, commodity green cloud storage system, *ACM International Symposium on High Performance Distributed Computing (HPDC 2010),* 332-335 (2010) — **Andi Bergen, July 26**

4

Graduate Student Research Paper Presentations

- Villegas, N.M., Müller, H.A., Tamura, G., Duchien, L., Casallas, R.: A framework for evaluating quality-driven self-adaptive software systems. In: *Proc. 6th Int. Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2011),* pp. 80-89 (2011) — **Lorena Castaneda, July 30**
- Ebrahimi, S., Villegas, N.M., Müller, H.A., Thomo, A.: SmarterDeals: a context-aware deal recommendation system based on the SmarterContext engine. *CASCON 2012:* 116-130 (2012) — **Nina Taherimakhsousi, July 30**
- McKinley, P.K., Sadjadi, M., Kasten, E.P., Cheng, B.H.C.: Composing Adaptive Software. *IEEE Computer* 37(7):56-64 (2004) — **Carlos Gomez, July 31**
- Tewari, V., Milenkovic, M.: Standards for Autonomic Computing, *Intel Technology Journal,* 10(4):275-284 (2006) — **Nitin Goyal, July 31**

5

Guidelines for Grad Presentations

- Format of presentation
  - Presentation 15-20 mins
  - Q&A 5 mins
  - Practice talk (!)
- Slides
  - High quality
  - Submit slides 2 days before presentation to instructor for approval
  - Submit final slides 1 day after presentation for posting on website
- Talk outline
  - Motivation
  - Problem
  - Approach
  - Relation to what we heard in the course so far
  - Contributions of the paper

6

MIDTERM DISCUSSION
FINAL EXAM — PREREVIEW

7

Dr. H.A. Müller — SENG 480B / CSC 485C / CSC 586B — Spring 2013
Self-Adaptive and Self-Managing Systems

Midterm Test

This test contains four questions worth a total of 100 points. Each question is worth 25 marks. This is a 50 minutes, closed books, closed notes, no calculators, no phones, and no gadgets test. Answer all questions in examination booklets.

1. ULS Systems
   - **a)** What are the main characteristics of Ultra Large Scale (ULS) systems? [15]
   - **b)** Explain in detail why the web is qualifies as a ULS system [10].
2. Autonomic Systems
   - **a)** Define and describe the notion of an autonomic manager [15].
   - **b)** Describe the purpose of each component in the MAPE-K loop in detail [10].
3. Feedback Systems
   - **a)** Explain the differences between positive and negative feedback [10].
   - **b)** Describe the notion of a PID controller in detail [15].
4. Self-Adaptive Systems
   - **a)** Define and describe the notion of a self-adaptive system [10].
   - **b)** What kind of information must be monitored and collected to make adaptation decisions at runtime? [15].

8

Characteristics of ULS Systems

- Ultra-large size in terms of
  - Lines of code
  - Amount of data stored, accessed, manipulated, and refined
  - Number of connections and interdependencies
  - Number of hardware elements
  - Number of computational elements
  - Number of system purposes and user perception of these purposes
  - Number of routine processes, interactions, and “emergent behaviours”
  - Number of (overlapping) policy domains and enforceable mechanisms
  - Number of people involved in some way
  - ……

9

What is an ULS System

- A ULS System has unprecedented scale in some of these dimensions
  - Lines of code
  - Amount of data stored, accessed, manipulated, and refined
  - Number of connections and interdependencies
  - Number of hardware elements
  - Number of computational elements
  - Number of system purposes and user perception of these purposes
  - Number of routine processes, interactions, and “emergent behaviours”
  - Number of (overlapping) policy domains and enforceable mechanisms
  - Number of people involved in some way

ULS systems will be interdependent webs of software-intensive systems, people, policies, cultures, and economics.

Scale Changes Everything

- Characteristics of ULS systems arise because of their scale
  - Decentralization
  - Inherently conflicting, unknowable, and diverse requirements
  - Continuous evolution and deployment
  - Heterogeneous, inconsistent, and changing elements
  - Erosion of the people/system boundary
  - Normal failures
  - New paradigms for acquisition and policy

These characteristics may appear in today’s systems, but in ULS systems they dominate. These characteristics undermine the assumptions that underlie today’s software engineering approaches.

From Systems of Systems to Ecosystems

- A ULS system comprises a dynamic community of interdependent and competing organisms in a complex and changing environment

- The concept of an ecosystem connotes complexity, decentralized control, hard-to-predict reactions to disruptions, difficulty of monitoring and assessment

In many ways, legacy systems are already participating in socio-technical ecosystems

12

Decentralized Ecosystems

- For 40 years we have embraced the traditional centralized engineering perspective for building software
  - Central control, top-down, tradeoff analysis
- Beyond a certain complexity threshold, traditional centralized engineering perspective is no longer sufficient and cannot be the primary means by which ultra-complex systems are made real
  - **Firms** are engineered—but the structure of the **economy** is not
  - The protocols of the **Internet** were engineered—but not the **Web** as a whole
- **Ecosystems** exhibit high degrees of complexity and organization—but not necessarily through engineering

13

ULS Systems Solve Wicked Problems

- ***Wicked problem***
  An ill-defined design and planning problem having incomplete, contradictory, and changing requirements.
- Solutions to wicked problems are often difficult to recognize because of complex interdependencies.
- This term was suggested by H. Rittel & M. Webber in "Dilemmas in a General Theory of Planning," *Policy Sciences 4, Elsevier* (1973)
- Wicked problems are problems that are not amenable to *analytic, reductionist analysis.*

14

Characteristics of Wicked Problems

- You don't understand the problem until you have developed a solution
  - There is no definitive formulation of the problem.
  - The problem is ill-structured
  - An evolving set of interlocking issues and constraints
- There is no stopping rule
  - There is also no definitive Solution
  - The problem solving process ends when you run out of resources
- Every wicked problem is essentially unique and novel
  - There are so many factors and conditions, all embedded in a dynamic social context, that no two wicked problems are alike
  - No immediate or ultimate test of a solution
  - Solutions to them will always be custom designed and fitted
- Solutions are not right or wrong
  - Simply better, worse, good enough, or not good enough.
  - Solutions are not true-or-false, but good-or-bad.
- Every solution to a wicked problem is a one-shot operation.
  - You can't learn about the problem without trying solutions.
  - Every implemented solution has consequences.
  - Every solution you try is expensive and has lasting unintended consequences (e.g., spawn new wicked problems).
- Wicked problems have no given alternative solutions
  - May be no feasible solutions
  - May be a set of potential solutions that is devised, and another set that is never even thought of.

15

Dr. H.A. Müller SENG 480B / CSC 485C / CSC 586B Spring 2013
Self-Adaptive and Self-Managing Systems

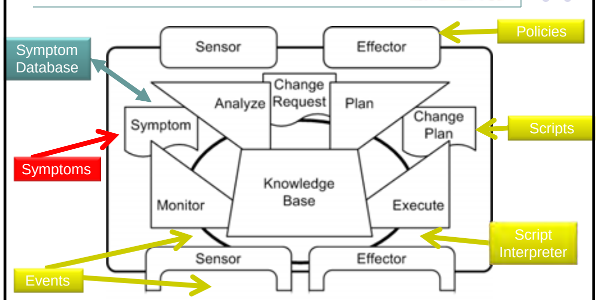
Midterm Test

This test contains four questions worth a total of 100 points. Each question is worth 25 marks. This is a 50 minutes, closed books, closed notes, no calculators, no phones, and no gadgets test. Answer all questions in examination booklets.

1. ULS Systems
   a) What are the main characteristics of Ultra Large Scale (ULS) systems? [15]
   b) Explain in detail why the web is qualifies as a ULS system [10].
2. Autonomic Systems
   a) Define and describe the notion of an autonomic manager [15]
   b) Describe the purpose of each component in the MAPE-K loop in detail [10].
3. Feedback Systems
   a) Explain the differences between positive and negative feedback [10]
   b) Describe the notion of a PID controller in detail [15].
4. Self-Adaptive Systems
   a) Define and describe the notion of a self-adaptive system [10]
   b) What kind of information must be monitored and collected to make adaptation decisions at runtime? [15].

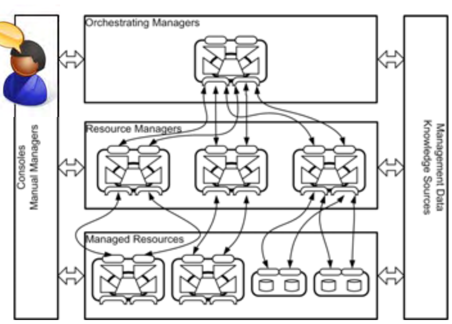
16

MAPE-K Loop Standards & Interfaces



17

ACRA AC Reference Architecture



18

MAPE-K Loop

Monitor

- Senses the managed process and its context
- Collects data from the managed resource
- Provides mechanisms to aggregate and filter incoming data stream
- Stores relevant and critical data in the knowledge base or repository for future reference.

Analyzer

- Compares event data against patterns in the knowledge base to diagnose symptoms and stores the symptoms
- Correlates incoming data with historical data and policies stored in repository
- Analyzes symptoms
- Predicts problems

19

MAPE-K Loop

Planner

- Interprets the symptoms and devises a plan
- Decides on a plan of action
- Constructs actions
  - building scripts
- Implements policies
- Often performed manually

Execute Engine

- Executes the change in the managed process through the effectors
- Perform the execution plan
- Often performed manually

20

MAPE-K Loop
Knowledge Base

- The four components of a MAPE-K loop work together by exchanging knowledge through the **knowledge base** to achieve the control objective.
- An autonomic manager
  - maintains its own knowledge
    - Information about its current state as well as past states
  - But also has access to knowledge which is shared among collaborating autonomic managers
    - Configuration database, symptoms database, business rules, provisioning policies, or problem determination expertise

21

Symptoms

- A *symptom* is a form of knowledge that indicates a possible problem or situation in the managed environment.
  - For example, "high fever" might be defined as a temperature "greater than 39 degrees Celsius"
  - The symptom is defined by the expression "temperature greater than 39 degrees Celsius" and described as "high fever"
- Symptoms are
  - Recognized in the monitor component of the MAPE-K loop
  - Used as a basis for analysis of a problem or a goal
  - Based on predefined elements—for example, definitions and descriptions in a symptoms DB
- Symptom definition
  - Expresses conditions used by the monitor to recognize the existence of a symptom
  - Specifies the unique characteristics of a particular symptom that is recognized.
- Symptoms are not just for self-healing
  - Symptoms are connected to self-healing because their primary intent is to indicate a problem
  - Symptoms can also be used as triggers for other kinds of problems
  - Virtually all kinds of problems or predictions may start due to the occurrence of a symptom

IBM: Symptoms Reference Specification Version 2.0 2006

22

Symptom Artifacts

- Symptom element
  - Contains all information necessary to create a new symptom occurrence
- Symptom occurrence
  - Contains the run-time information associated with a specific instance of a symptom element
  - Each occurrence basically refers to the same symptom as it is defined in the symptom element, but the context to which it is applied may vary.
- Correlation engine
  - Contains the logic used to create symptom elements
  - As input the correlation engine receives external stimuli and checks if a symptom occurrence should be created as a response.

IBM: Symptoms Reference Specification Version 2.0 2006

23

Symptom Artifacts

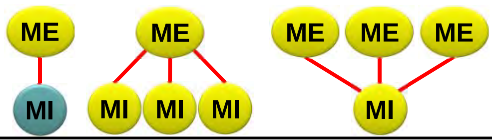
- Symptom metadata
  - The generic part of the information that composes a symptom
  - It is present on all kinds of knowledge, and is used when knowledge must be treated generically, even though it is a symptom element
  - This is the "what" part of a symptom
- Symptom schema
  - The specific part of the information that composes a symptom
  - It is the template that is used when a symptom occurrence is created
  - The symptom schema contributes to the "what" part of a symptom
- Symptom definition
  - A generic piece of logic that can be used to recognize a symptom
  - As expected, this logic should be compatible with the respective correlation engine that will be used to process the symptom
  - This is the "how" part of a symptom

IBM: Symptoms Reference Specification, Version 2.0, 2006

24

Manageability Endpoints

- A Manageability Endpoint (ME) exposes the state and the management operations for a resource
- An autonomic manager communicates with a manageability endpoint through the Manageability Interface (MI)



25

Manageability Interface

- An MI for monitoring and controlling a managed resource consists of sensors and effectors
- Sensors obtain data from the resource
  - read state variables in the ME
- Effectors perform operations on the resource
  - call methods in the ME
- Critical success factors for AC initiative
  - Separating AMs and MEs
  - Standardizing MIs

26

Characteristics of Autonomic Systems

- Self awareness, reflexivity, identity
  - Possesses a system identity
  - Must know itself
  - Needs detailed knowledge of its components, current status interconnections with other systems and available resources to manage itself
- Able to configure and reconfigure itself under varying and unpredictable conditions
  - For example, adaptive algorithms running on each subsystem could learn the best configurations to deliver functionality in different ways to achieve mandated performance
- Continually seek to optimize its operations
  - Adaptive algorithms for monitoring and execution

27

Characteristics of Autonomic Systems

- Systems that self-manage
  - self-configure, self-tune, self-repair, self-protect, …
- For a software system to be autonomic, it needs to support a range of behaviours; then
  - Self-configuring means choosing a suitable behaviour, based on user preferences, context, …
  - Self-tuning means choosing behaviours that optimize certain qualities (performance, year-end profits, …)
  - Self-repairing means shifting execution to another behaviour, given that the current one is failing
  - Self-protecting means choosing a behaviour that minimizes risks (attacks, viruses, …)

28

What Do Self-Managing Systems Deliver?

Increased Responsiveness

Adapt to dynamically changing environments

Business Resiliency

Discover, diagnose, and act to prevent disruptions

Operational Efficiency

Tune resources and balance workloads to maximize use of IT resources

Secure Information and Resources

Anticipate, detect, identify, and protect against attacks

*Self – **

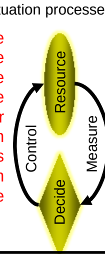
29

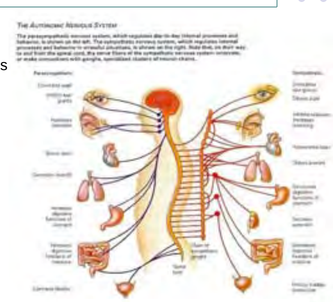
Most Famous Feedback System Autonomic Nervous System (ANS)

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

30

Interesting Architectural Note

- Architecturally the ANS seems to separate the normal day-to-day internal processes from the exceptional, stressful situation processes
  - Parasympathetic
    - Day-to-day internal processes
  - Sympathetic
    - Stressful situation processes
- Could we use this interesting architectural design decision for self-managing and self-adaptive systems?

31

Dr. H.A. Müller | SENG 480B / CSC 485C / CSC 586B Self-Adaptive and Self-Managing Systems | Spring 2013

Midterm Test

This test contains four questions worth a total of 100 points. Each question is worth 25 marks. This is a 50 minutes, closed books, closed notes, no calculators, no phones, and no gadgets test. Answer all questions in examination booklets.

1. ULS Systems
   a) What are the main characteristics of Ultra Large Scale (ULS) systems? [15]
   b) Explain in detail why the web is qualifies as a ULS system [10].
2. Autonomic Systems
   a) Define and describe the notion of an autonomic manager [15].
   b) Describe the purpose of each component in the MAPE-K loop in detail [10].
3. Feedback Systems
   a) Explain the differences between positive and negative feedback [10].
   b) Describe the notion of a PID controller in detail [15].
4. Self-Adaptive Systems
   a) Define and describe the notion of a self-adaptive system [10].
   b) What kind of information must be monitored and collected to make adaptation decisions at runtime? [15].

32

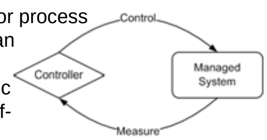
Types of Feedback

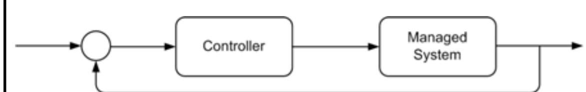
- **Negative feedback**
  - Stabilizes operation; regulates within a set and narrow range
  - Classic examples
    - Thermostat control
    - Homeostasis
- **Positive feedback**
  - Increase, accelerate, or enhance output created by a stimulus that has already been activated
  - Classic example
    - Audio feedback—sound from loudspeakers enters a poorly placed microphone and gets amplified, and as a result the sound gets louder and louder
    - Blood platelet accumulation, which, in turn, causes blood clotting in response to a break or tear in the lining of blood vessels
    - Release of oxytocin to intensify the contractions that take place during childbirth
- **Bipolar feedback**
  - Either increase or decrease output
  - Bipolar feedback is present in many natural and human systems
  - Feedback is usually bipolar in natural environments producing synergic and antagonistic responses to the output of system

33

Feedback Systems

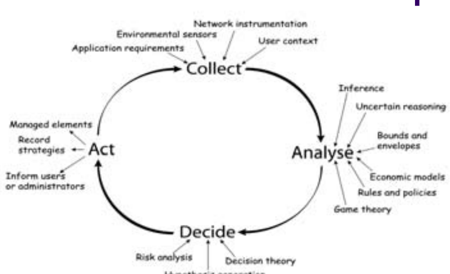
- **Merriam-Webster's Online Dictionary**
  the return to the input of a part of the output of a machine, system, or process (as for producing changes in an electronic circuit that improve performance or in an automatic control device that provide self-corrective action)





34

Autonomic Feedback Loop



Dobson, S. et al.: A Survey of Autonomic Communications. ACM Trans. on Autonomous and Adaptive Systems (TAAS) 1(2):223-259 (2006)

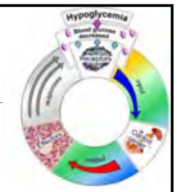
35

Types of Feedback

- **Negative feedback**
  - Stabilizes operation; regulates within a set and narrow range
  - Classic examples
    - Thermostat control
    - Homeostasis
- **Positive feedback**
  - Increase, accelerate, or enhance output created by a stimulus that has already been activated
  - Classic example
    - Audio feedback—sound from loudspeakers enters a poorly placed microphone and gets amplified, and as a result the sound gets louder and louder
    - Blood platelet accumulation, which, in turn, causes blood clotting in response to a break or tear in the lining of blood vessels
    - Release of oxytocin to intensify the contractions that take place during childbirth
- **Bipolar feedback**
  - Either increase or decrease output
  - Bipolar feedback is present in many natural and human systems
  - Feedback is usually bipolar in natural environments producing synergic and antagonistic responses to the output of system

36

Physiological Regulation *Homeostasis*



- ***Homeostasis*** is the property of a system that regulates its internal environment and tends to maintain a stable, constant condition
- In animals the internal environment of our bodies must have certain conditions within tolerable limits to continue the healthy functioning.
- This is done by a process called negative feedback control, where various receptors and effectors bring about a reaction to ensure that such conditions remain favourable—the control of blood sugar concentrations, water concentrations, or temperature.
- Physiological homeostasis = Physical equilibrium
  - Glucose level in the bloodstream drops
  - Person requires glucose in cells to meet the demand for ATP—Adenosine triphosphate
  - The body detects this with a particular receptor designed for this function
  - These receptors release hormones, chemical messages that initiate the start of the feedback mechanism
  - The hormones travel to their target tissue and initiate a corrective response
  - In this case, the response is the secretion of more glucose into the bloodstream

37

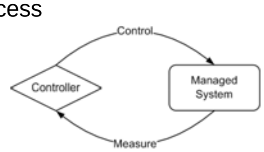
Ice-Albedo Feedback

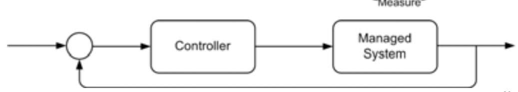
- Albedo
  - The amount of energy reflected by a surface; scale from zero to one
  - For dark colors albedo close to zero; light ones close to one
- Arctic sea ice is covered with snow all winter.
- Bright white, the snow-covered ice has a high albedo so it absorbs very little of the solar energy that gets to it.
- Because Earth's temperature is climbing, the snow on top of the ice melts earlier in the spring
- There is more time during the summer for the compounding cycle of melting ice, lowering albedo, trapping of more solar energy, and more ice to melt.
- Albedo feedback is positive because the initial temperature change is amplified.

38

Feedback Control System

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





39

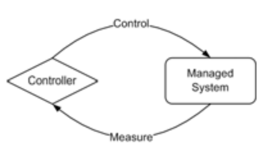
Control Theory

- A theory that deals with influencing the behavior of dynamical systems
- An interdisciplinary subfield of science, which originated in engineering and mathematics

40

Control Systems are Ubiquitous

- Cruise control
- Fuel injection
- Flight control
- Climate Control
- Health Care
- Model Helicopters
- Rumba
- iRobots
- Radiotheraphy



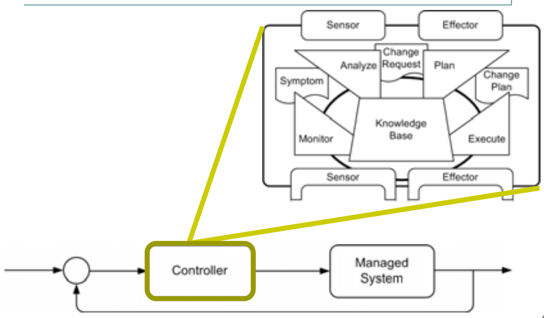
41

Control System Goals: Self-Management

- Regulation
  - Thermostat, target service levels
- Tracking
  - Robot movement
  - Adjust TCP window to network bandwidth
- Optimization
  - Best mix of chemicals
  - Minimize response times

42

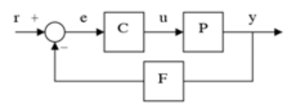
Controller as an Autonomic Element



43

Closed Loop Controller or Feedback Controller

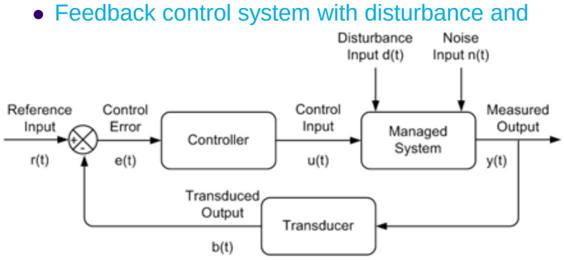
- The output *y(t)* of the feedback system is fed back through a sensor measurement *F* to the reference value *r(t)*.
- The controller *C* then takes the error *e* (difference) between the reference and the output to change the inputs *u* to the control process *P*.
- SISO
  - Single-input-single-output (*SISO*) control system
  - Variables are simple scalar values (i.e., r(t), e(t), u(t), y(t)
- MIMO
  - Multi-Input-Multi-Output systems, with more than one input/output, are common
  - Variables are vectors



44

Realization of a Dynamic Architecture

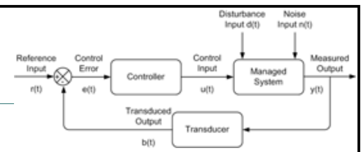
- Feedback control system with disturbance and



Hellerstein, Diao, Parekh, Tilbury: Feedback Control of Computing Systems. John Wiley & Sons (2004)

45

Realization of a Dynamic Architecture



- Reference input
  - Goal, objectives, specified desired output
- Control Error
  - Reference input minus transduced output
- Control Input
  - Parameters which affect behavior of the system—number of threads, CPU, memory
- Disturbance input
  - Affects control input—arrival rate
- Controller
  - Change control input to achieve reference input—design is based on a model of the managed system
- Managed system
  - Dynamical system, process, plant—often characterized by differential equations
- Measured output
  - Measurable feature of the system—response time
- Noise input
  - Affects measured output
- Transducer
  - Transforms measured output to compare with reference

Controller Algorithm based on Managed System Model

- *"All models are wrong, some models are useful."*
  - generally attributed to the statistician George Box
- The design of the controller algorithm is based on a model of the managed system or process
- Approaches
  - Analytical modeling: physical and mathematical laws
  - Experimental modeling: data fitting from observed input and output
- The control algorithm changes *u(t)* based on the error *e(t) = r(t) - b(t)*
  - Proportional—if *e(t)* is high, then *u(t)* should be high
  - Integrative—eliminates transients; sum of all previous errors
  - Derivative—anticipate the trends; rate of change of the error
  - PID—computation based on the error (proportional), the sum of all previous errors (integral) and the rate of change of the error (derivative)

47

PID Controller

- The PID algorithm is the most popular feedback controller algorithm used
- It is a robust easily understood algorithm that can provide excellent control performance despite the varied dynamic characteristics of processes
- PID algorithm consists of three basic modes:
  - **P**roportional mode
  - **I**ntegral mode
  - **D**erivative mode

48

P, PI, or PID Controller

- When utilizing the PID algorithm, it is necessary to decide which modes are to be used (P, I or D) and then specify the parameters (or settings) for each mode used.
- Generally, only three basic algorithms are used: P, PI or PID

$$\underbrace{u(t) = K_p}_{\text{proportional gain}} e(t) + \underbrace{K_i}_{\text{integral gain}} \int_0^t e(t)dt + \underbrace{K_d}_{\text{derivative gain}} \frac{de(t)}{dt}$$

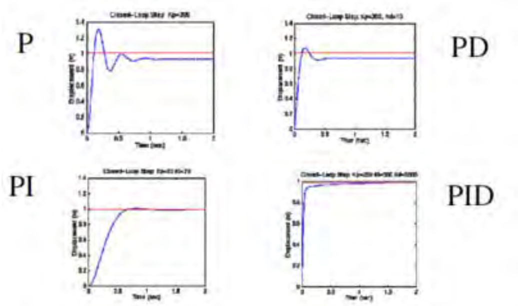
49

Controller Effects

- A proportional controller (P) reduces error responses to disturbances, but still allows a steady-state error.
- When the controller includes a term proportional to the integral of the error (I), then the steady state error to a constant input is eliminated, although typically at the cost of deterioration in the dynamic response.
- A derivative control typically makes the system better damped and more stable

50

PID Controller



Closed-Loop Response

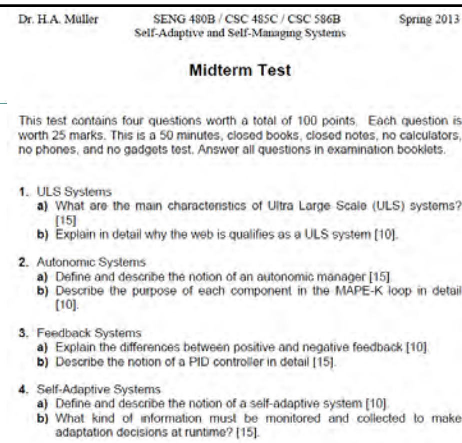
	Rise time	Max overshoot	Settling time	Steady-state error
P	Decrease	Increase	Small change	Decrease
I	Decrease	Increase	Increase	Eliminate
D	Small change	Decrease	Decrease	Small change

52

PID Controller

- Output feedback
  - From Proportional action
  - Compare output with set-point
- Eliminate steady-state offset or error
  - From Integral action
  - Apply constant control even when error is zero
  - Eliminates transients; sum of all previous errors
- Anticipation
  - From Derivative action
  - React to rapid rate of change before errors grows too big
  - Anticipate the trends; rate of change of the error

53



Dr. H.A. Müller | SENG 480B / CSC 485C / CSC 586B Self-Adaptive and Self-Managing Systems | Spring 2013

Midterm Test

This test contains four questions worth a total of 100 points. Each question is worth 25 marks. This is a 50 minutes, closed books, closed notes, no calculators, no phones, and no gadgets test. Answer all questions in examination booklets.

1. ULS Systems
   a) What are the main characteristics of Ultra Large Scale (ULS) systems? [15]
   b) Explain in detail why the web is qualifies as a ULS system [10].
2. Autonomic Systems
   a) Define and describe the notion of an autonomic manager [15]
   b) Describe the purpose of each component in the MAPE-K loop in detail [10].
3. Feedback Systems
   a) Explain the differences between positive and negative feedback [10]
   b) Describe the notion of a PID controller in detail [15].
4. Self-Adaptive Systems
   a) Define and describe the notion of a self-adaptive system [10].
   b) What kind of information must be monitored and collected to make adaptation decisions at runtime? [15].

54

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

56

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

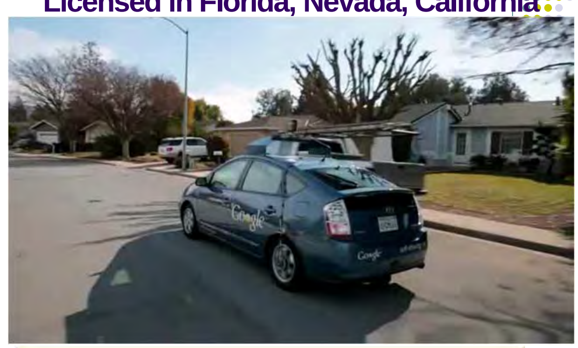
57

Context is Big Data

58

Capture Context with Sensors and Wearable Computers

Google Driverless Car
Licensed in Florida, Nevada, California



http://www.youtube.com/watch?v=cdqQpa1pUU

60

Monitoring in Dynamical Systems

- Perform critical regression tests dynamically to observe satisfaction of requirements
  - Testing run-time (and design-time) governance
  - Govern and enforce rules and regulations
- Perform V&V operations (transformations) regularly to ascertain V&V properties
  - Monitor compliance and conformance
  - Assess whether services are used properly
  - Recognizing normal and exceptional behaviour
- Monitor functional & non-functional requirements when the environment evolves
  - SLAs
  - Assess and maintain quality of service (QoS)
  - Manage tradeoffs

61

END MIDTERM DISCUSSION

62