Welcome to SENG 371 Software Evolution Spring 2013

A Core Course of the BSEng Program

Hausi A. Müller, PhD PEng
Professor, Department of Computer Science
Associate Dean Research, Faculty of Engineering
University of Victoria

Announcements

- Course website
  - Up and running
  - http://www.engr.uvic.ca/~seng371
- Labs start this week
  - Instructors
    - Lorena Castaneda
    - Pratik Jain
    - Przemek Lach
  - This week
    - Visualization tools
- Assignment 1
  - Due Jan 28
  - Cite your sources
  - Part I — Useful definitions
  - Part II — Growing systems in emergent organizations
  - Part III — Ultra large scale systems (ULS)

2

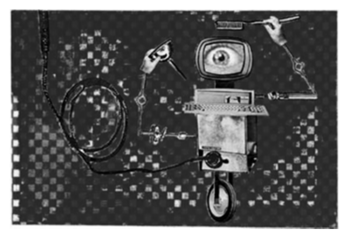
Reading assignments

- IBM Corporation: An Architectural Blueprint for Autonomic Computing, Fourth Edition (2006) http://people.cs.kuleuven.be/~danny.weyns/csds/IBM06.pdf
- Truex, Baskerville, Klein: Growing Systems in Emergent Organizations. Communications of the ACM, 42(8):117-123 (1999). http://portal.acm.org/citation.cfm?id=310930.310984&coll=GUIDE&dl=GUIDE,ACM&CFID=2240896&CFTOKEN=98671917
- Northrop, et al.: 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

3

IBM's Complexity Solution

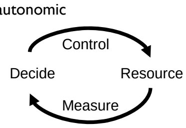
- Automation through self-adaptive and self-managing systems or autonomic computing



4

What is Autonomic Computing?

- Webster's definition
  - Acting or occurring involuntarily; automatic: an autonomic reflex
  - Relating to, affecting, or controlled by the autonomic nervous system or its effects or activity
  - Autonomic nervous system: that part of the nervous system that governs involuntary body functions like respiration and heart rate
- IBM's definition
  - An approach to self-managed computing systems with a minimum of human interference
  - The term derives from the body's autonomic nervous system, which controls key functions without conscious awareness or involvement

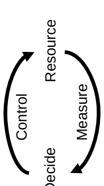


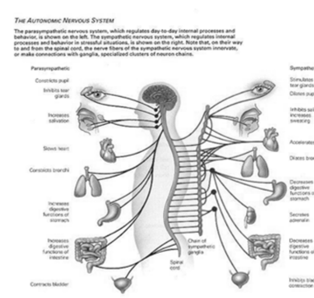
5

The Most Famous Autonomic System

- Autonomic nervous system
- 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



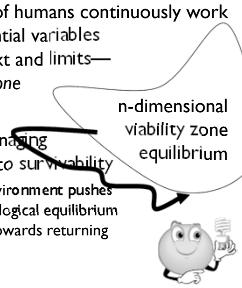


Monitor and Regulate

6

Ideas for Adaptive Systems in Nature
Viability Zone

- The internal mechanisms of humans continuously work together to maintain essential variables within physiological context and limits—the *n-dimensional viability zone*
- The goal of human self-managing behavior is directly linked to survivability
  - If the external or internal environment pushes the system outside its physiological equilibrium zone, the system will work towards returning to the equilibrium zone



Aubin, Bayen, Saint-Pierre: *Viability Theory: New Directions*, Springer (2011)

7

What autonomic or self-managing systems deliver

Increased Responsiveness

Adapt to dynamically changing environments

Business Resiliency

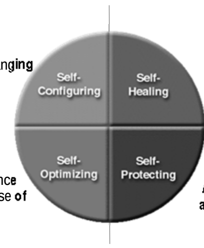
Discover, diagnose, and act to prevent disruptions

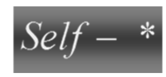
Operational Efficiency

Tune resources and balance workloads to maximize use of IT resources

Secure Information and Resources

Anticipate, detect, identify, and protect against attacks

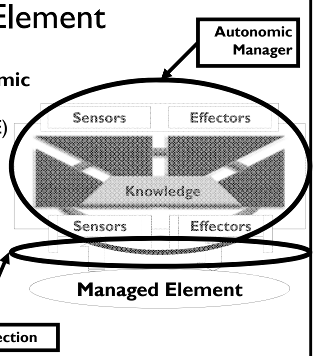




8

Autonomic Element

- Consists of an **Autonomic Manager (AM)** and an Autonomic Element (AE)
- Manager and managed element form a **level of indirection**
  - Spatially and temporally separate entities
  - Enterprise Service Bus



9

Monitor	Analyzer
• 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.	• 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

MAPE-K Loop

10

Planner	Execute Engine
• Interprets the symptoms and devises a plan • Decides on a plan of action • Constructs actions ◦ building scripts • Implements policies • Often performed manually	• Executes the change in the managed process through the effectors • Perform the execution plan • Often performed manually

MAPE-K Loop

11

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

12

Design Considerations Monitor

- The monitor function provides the mechanisms that collect, aggregate, filter and report details collected from a managed resource.
  - What kinds of data and events are collected from which sources, sensors, or probes?
  - Are there common event formats?
  - What is the sampling rate and is it fixed or varying?
  - Are the sampled sources fixed or do they change dynamically?
  - What are appropriate filters for the data streams?

13

Design Considerations Monitor

- A large portion of the knowledge base is monitored information
  - How much information is needed for future reference?
  - With the detailed level of reporting and logging functions within software systems today, it is important to monitor and store data that is really going to be of use to the control loop
  - If large amounts of log data, for example, are stored, performance might deteriorate because data is constantly being monitored when it has no relevance to the system.

14

Design Considerations Analyzer

- The analyzer provides mechanisms to correlate and model complex situations
  - Embodies the control model together with the planner
  - Time-series forecasting and queuing models
- These mechanisms allow the autonomic manager to learn about the IT environment and help predict future situations.
  - How are the collected data represented and stored?
  - What are appropriate diagnosis methods to analyze the data?
  - How is the current state of the system assessed?
  - How much past state needs to be kept around?
  - How are critical states archived?
  - How are common symptoms recognized (e.g., symptoms db)?

15

Design Considerations Planning Engine

- The planning engine provides mechanisms to construct the actions needed to achieve goals and objectives.
- It uses policy information to guide its work.
  - How is the future state of the system inferred and how is a decision reached?
    - With off-line simulation, quality of service (QoS) objectives, or utility goal functions
  - What models and algorithms are used for trade-off analysis?
  - What are the priorities for adaptation across multiple control loops and within a single control loop?
  - Under what conditions should adaptation be performed?
    - Allow for head-room or avoid system thrashing while considering timing issues relating to the required adaptations

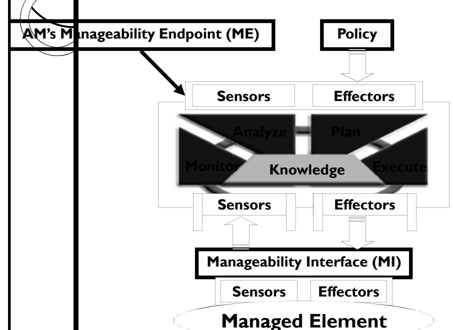
16

Design Considerations Execution Engine

- The execution engine provides mechanisms to control the execution of a plan by updating the managed element dynamically.
  - What are the managed elements and how can they be manipulated?
    - By parameter tuning
    - By injecting new algorithms
  - Are changes of the system pre-computed opportunistically assembled, composed, or generated?
    - Switching between known configurations

17

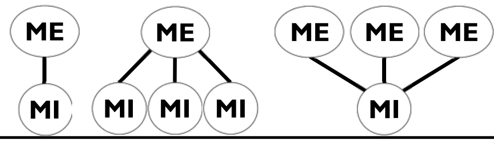
Autonomic Manager



18

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)

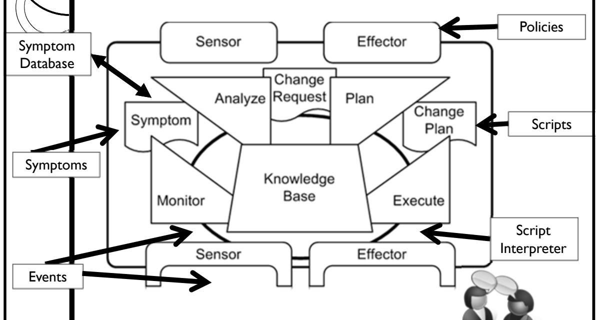


Manageability Interface

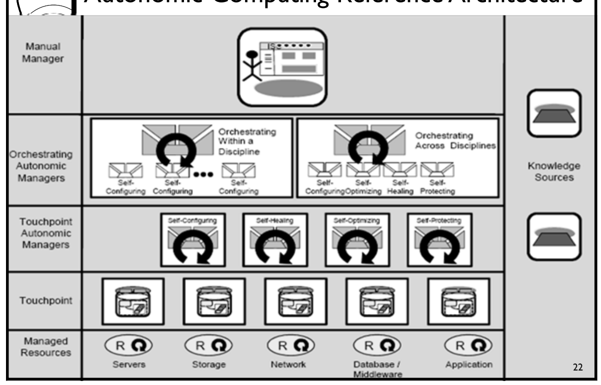
- An MI for monitoring and controlling a managed resource consists of sensors and effectors
- Sensors obtain data from the resource
  - read state variables in the ME
- Effectors perform operations on the resource
  - call methods in the ME
- Critical success factors for AC initiative
  - Separating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

MAPE-K Loop Standards & Interfaces

MD Metaphor



ACRA
Autonomic Computing Reference Architecture



ACRA
Autonomic Computing Reference Architecture

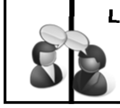
Level 5—highest
- Manual manger who operates a common system management interface

Level 4
- Autonomic Managers to integrate and orchestrate several self-* capabilities for a particular domain (e.g., DB, weather station)
- Implements system-wide capabilities

Level 3
- Implements specific self-* using Autonomic Managers (AM)

Level 2
- Consistent, standard Manageability Interfaces (MI) for accessing and controlling the managed resources in a uniform manner
- The MIs are implemented with Manageability Endpoints (ME)

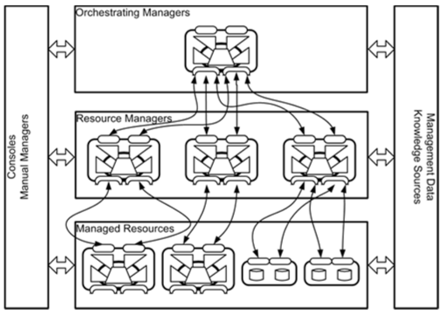
Level 1—lowest
- System components or managed resources (hardware, software) possibly with embedded self-management



Manual Manager

- Management or integrated solutions console
- Enables a human to perform and delegate management functions
- Collaborates with or orchestrates autonomic managers
- Set-up, configuration, run-time monitoring, control
- Manage trust—different levels of feedback
- Connecting knowledge source
- Specifying policies

ACRA
Autonomic Computing Reference Architecture



25

Class Participation Assignment

- Pick a self-managing scenario
- Define managed resources
- Define managing goals
- Define trade-off choices

26

Useful Papers under Resources Course Web Site

- Ganek, A.G., Corbi, T.A.: The Dawning of the Autonomic Computing Era. IBM Systems Journal 42(1):5-18 (2003)
- Kephart, J.O., Chess, D.M.: The Vision of Autonomic Computing. IEEE Computer 36(1):41-50 (2003)
- Kluth, A.: Information Technology: Make It Simple. The Economist (2004)
- Oreizy, P., Medvidovic, N., Taylor, R.N.: Architecture-Based Runtime Software Evolution. (Most Influential Paper Award at ICSE 2008) In: ACM/IEEE International Conference on Software Engineering (ICSE 1998), pp. 177-186 (1998)
- Huebscher, M.C., McCann, J.A.: A Survey of Autonomic Computing—Degrees, Models, and Applications. ACM Computing Surveys, 40 (3):7:1-28 (2008)
- Müller, H.A., Kienle, H.M., Stege, U.: Autonomic Computing: Now You See It, Now You Don't—Design and Evolution of Autonomic Software Systems. In: De Lucia, A.; Ferrucci, F. (eds.): Software Engineering International Summer School Lectures: University of Salerno. LNCS, Springer-Verlag, Heidelberg, pp. 32–54 (2009)
- Dobson, S., Denazis, S., Fernandez, A., Gaiti, D., Gelenbe, E., Massacci, F., Nixon, P., Saffre, F., Schmidt, N., Zambonelli, F.: A Survey of Autonomic Communications. ACM Transactions on Autonomous and Adaptive Systems (TAAS) 1(2):223-259 (2006)

27