Adaptive Software Systems
GS/EECS 6432

Marin Litoiu
Department of Electrical Engineering and Computer Science
York University
mlitoiu@yorku.ca
http://www.ceraslabs.com

Syllabus Overview

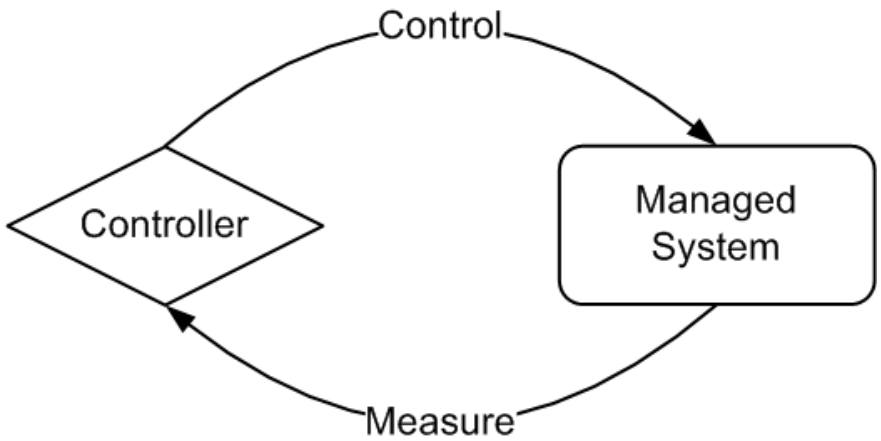
- Introduction to Adaptive Software
- Requirements of Cyber-Physical Systems, Ultra Large Scale Systems, Internet of Things
- Adaptive Architectures
- Feedback Loops
- **MAPE-K: Monitoring, Analysis, Planning, Execution**
- **MIAC, MRAC, Hierarchical Architectures**
- Software Defined Infrastructures
- Cloud Computing, Big Data, Containers
- Runtime Models for Elastic Applications
- Machine Learning, Queuing, Control Theory Linear Models
- Analysis and Planning
- Searched Based, Proportional, Integrative, Derivative Controllers, Control Theory
- Engineering Adaptive Applications
- Policy Languages, Utility Functions
- Case Studies

Summary

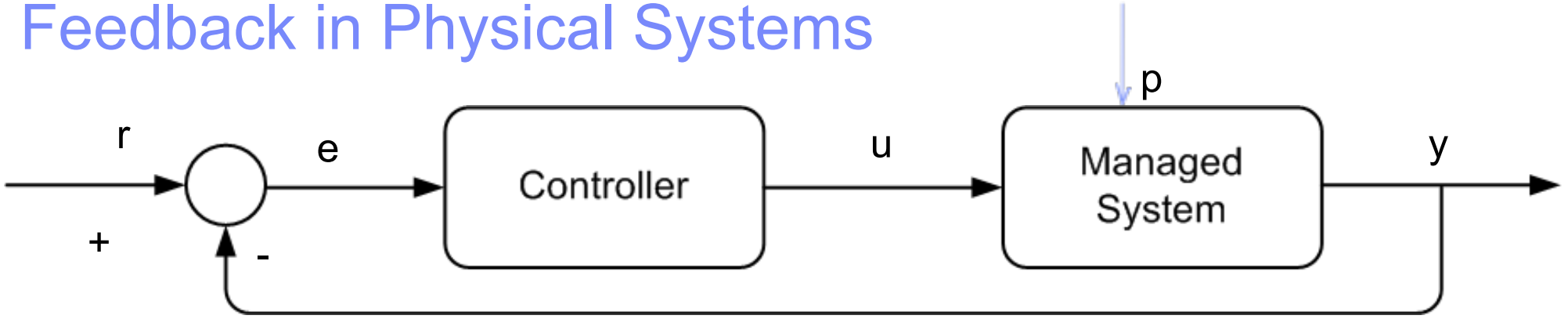
- **We looked at two different domains**
  - IOT systems are software intensive
    - What is adapted is the “thing,” a car, a city, a factory
  - Software systems(autonomic): what is adapted is the software itself
    - You can adapt its performance, security, configuration
- **To adapt (change structure or behavior), need to**
  - Monitor/sense the environment
  - Analyze the context and state
  - Perform changes as the system runs

Summary: Feedback Systems

- **Merriam-Webster's Online Dictionary**
  the return to the input of a part of the output of a machine, system, or process



Feedback in Physical Systems



- **The main goal: compensates for uncertainties (p) at design time**
- **At design time, it is important to know the direction of the disturbances, not the magnitude**
- **It simplifies the design of the system (avoids feedforward)**
  - The system is the best model
- **It is universal and effective, works the same for cruise control, direction control, distance control, etc..**

Summary: Autonomic Computing

- ***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.***
- ***Means***
  - Self configuring
  - Self healing
  - Self optimizing
  - Self protecting

Architectures

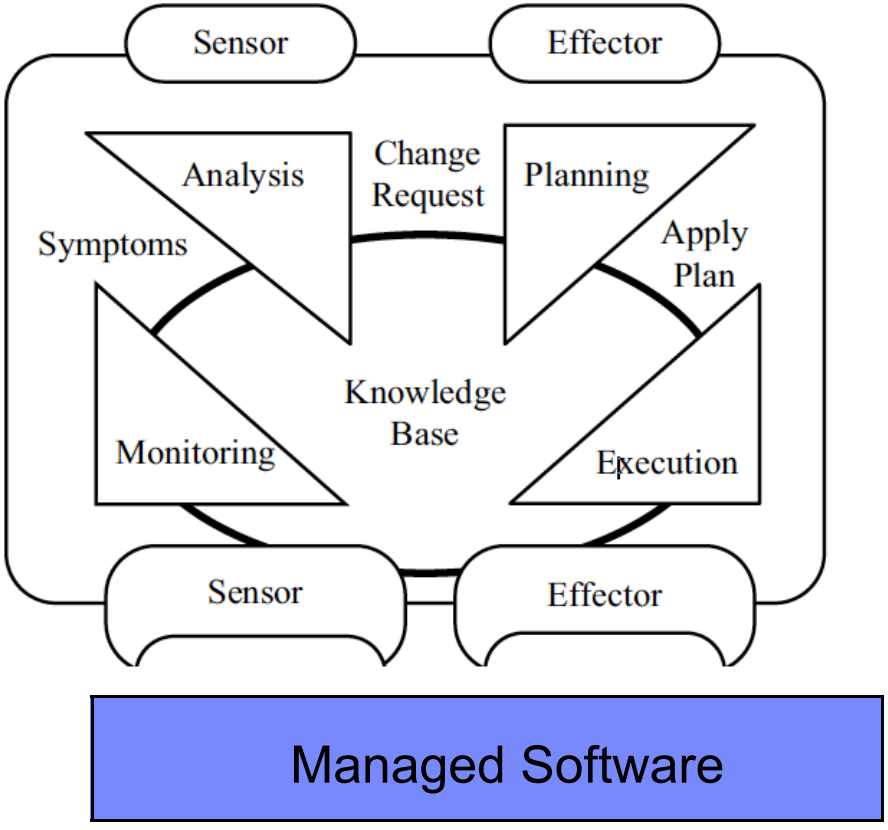
- **Slides based on**

[1] Y Brun, GDM Serugendo, C Gacek, H Giese, H Kienle… - Engineering self-adaptive systems through feedback loops in Software engineering for self-adaptive systems, 2009

[2] Kramer, J., Magee, J.: Self-managed Systems: An Architectural Challenge. In: Future of Software Engineering (FoSE 2007), pp. 259-268, IEEE Computer Society, Washington, DC, USA (2007)

[3] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," in *Computer*, vol. 36, no. 1, pp. 41-50, Jan 2003.

Feedback in Autonomic Computing: MAPE-K Loop[3]

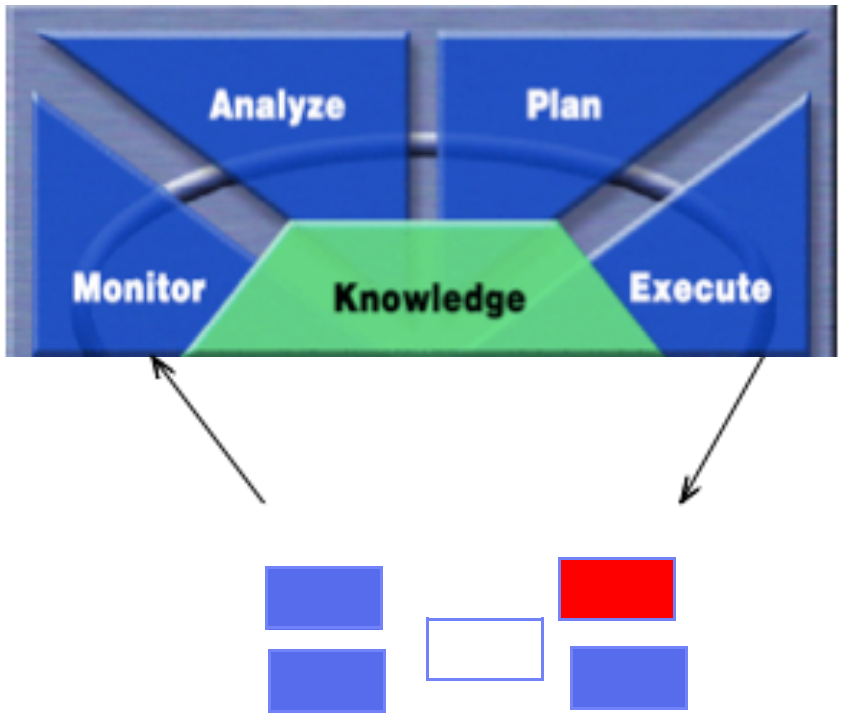


Just another representation of the feedback loop

- Focus of on the software architecture
- Different terminology
  - autonomic/adaptive manager replaces the controller

Allows for Hierarchical Control

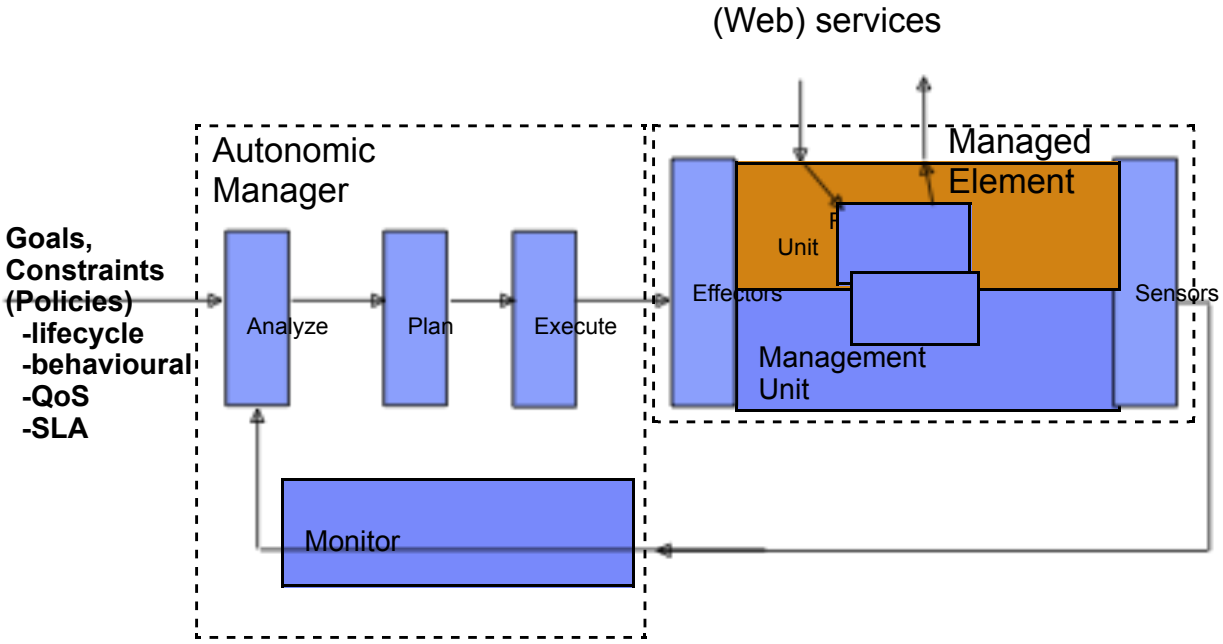
AC Architecture: Autonomic Manager, Managed Element



Autonomic manager

Managed Element

Autonomic Element as a Feedback Loop

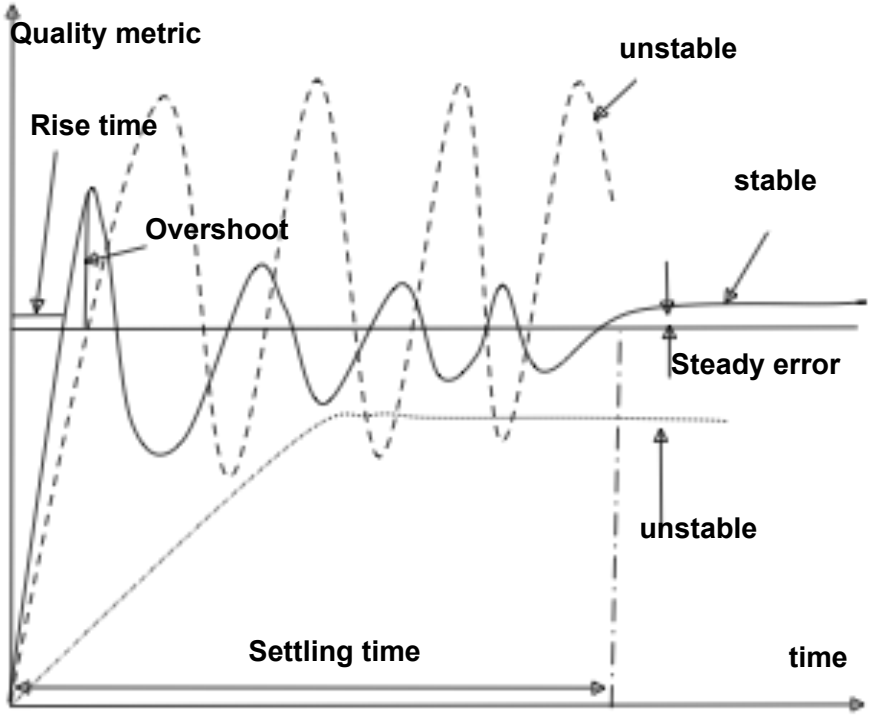


- Managed Element:
  - an object, a process a server, a database
  - Provides services
  - Is traceable, observable, controllable, comprehensible
  - Has sensors and effectors

- Autonomic Manager:
  - Analyzes
  - Plans
  - Executes

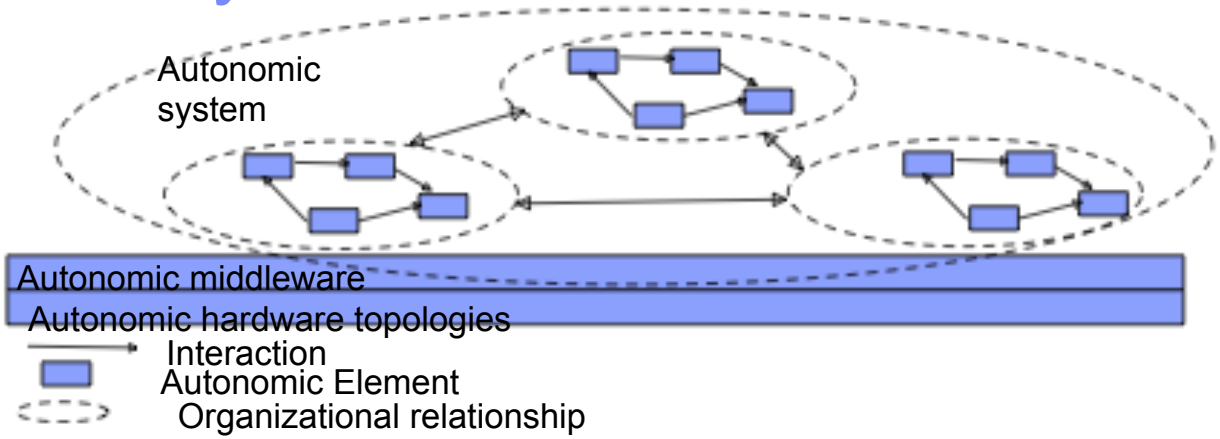
- Monitor:
  - logs, traces
  - etc

Quality of Autonomic Computing/Feedback Control



- short rise time
- short settling time
- small overshoot
- no oscillations
- stability
- tolerance to perturbations

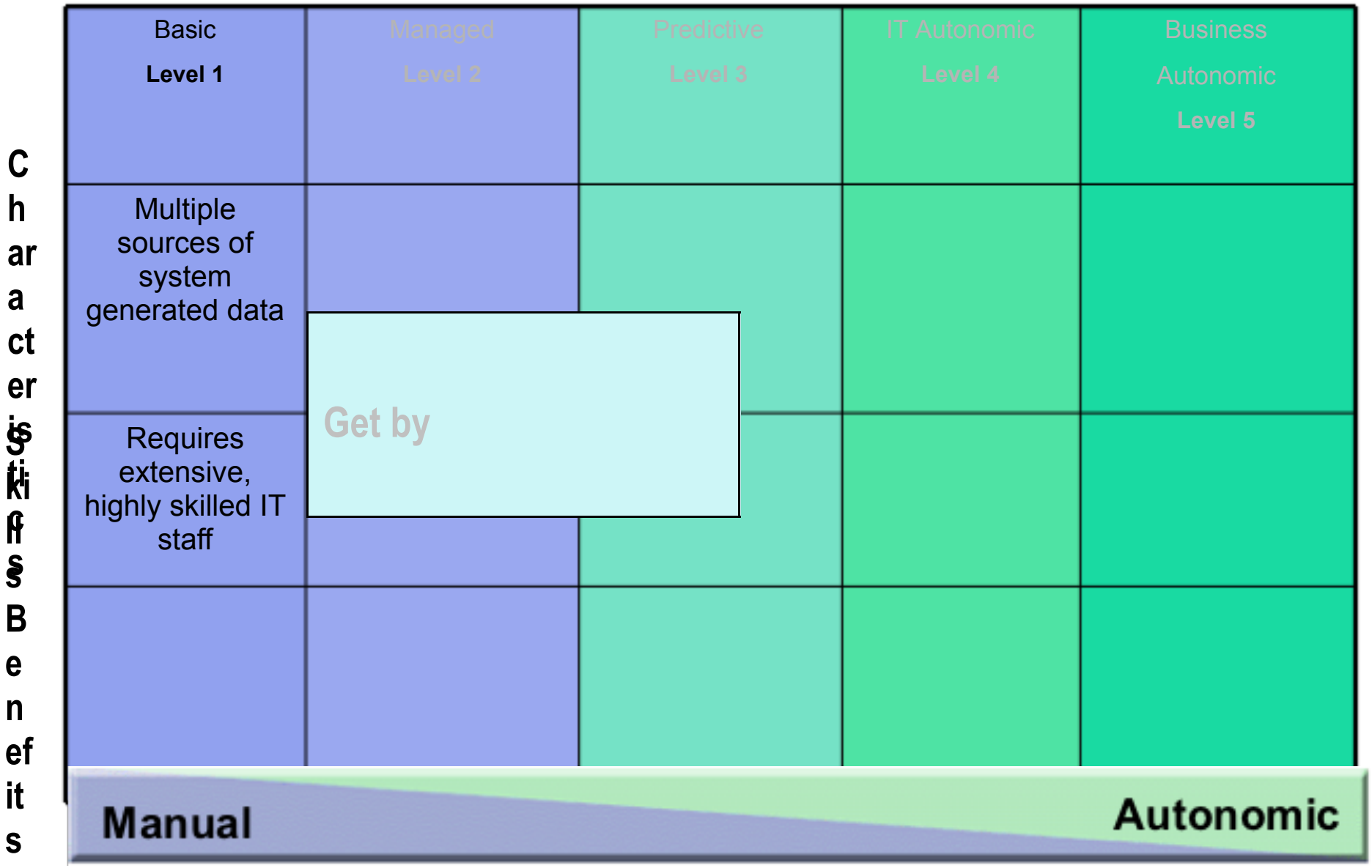
Autonomic Systems



- An autonomic system= a collection of autonomic elements (sometimes with conflicting individual goals)
  - requires conflict resolution
  - resource and goal negotiations
  - autonomic hardware and middleware

The Road to AC adoption (Maturity Level)

Evolving to Autonomic Computing: Basic



Evolving to Autonomic Computing: Managed

Characteristics / Benefits	Basic Level 1	Managed Level 2	Predictive Level 3	IT Autonomic Level 4	Business Autonomic Level 5
	Multiple sources of system generated data	Consolidation of data and actions through management tools			
	Requires extensive, highly skilled IT staff	IT staff analyzes and takes actions			
		Greater system awareness Improved productivity			

Take control

Manual → **Autonomic**

Evolving to Autonomic Computing: Predictive

Characteristics	Basic Level 1	Managed Level 2	Predictive Level 3	IT Autonomic Level 4	Business Autonomic Level 5
	Multiple sources of system generated data	Consolidation of data and actions through management tools	System monitors, correlates and recommends actions		
	Requires extensive, highly skilled IT staff	IT staff analyzes and takes actions	IT staff approves and initiates actions		
Benefits		Greater system awareness Improved productivity	Reduced dependency on deep skills Faster/better decision making		

Built-in analysis

Manual → Autonomic

Evolving to Autonomic Computing: IT Autonomic

Characteristics / Benefits	Basic Level 1	Managed Level 2	Predictive Level 3	IT Autonomic Level 4	Business Autonomic Level 5
	Multiple sources of system generated data	Consolidation of data and ac… n…	System monitors, correlates and …	System monitors, correlates and takes action	
	Requires extensive, highly skilled IT staff	… analyzes and takes actions	… approves and initiates actions	IT staff manages performance against SLAs	
		Greater system awareness Improved	Reduced dependency on deep skills	Balanced human/system interaction	

Dynamic response

Manual → Autonomic

Evolving to Autonomic Computing: Business Autonomic

Characteristics	Basic Level 1	Managed Level 2	Predictive Level 3	IT Autonomic Level 4	Business Autonomic Level 5
	Multiple sources of system generated data	Consolidation of data and actions t manag too	System monitors,	System monitors, correlates and	Integrated components dynamically managed by business rules/policies
	Requires extensive, highly skilled IT staff	IT s analyzes and takes actions	approves and initiates actions	performance against SLAs	IT staff focuses on enabling business needs
Benefits		Greater system awareness Improved productivity	Reduced dependency on deep skills Faster/better decision making	Balanced human/system interaction IT agility and resiliency	Business policy drives IT management Business agility and resiliency

Business impact

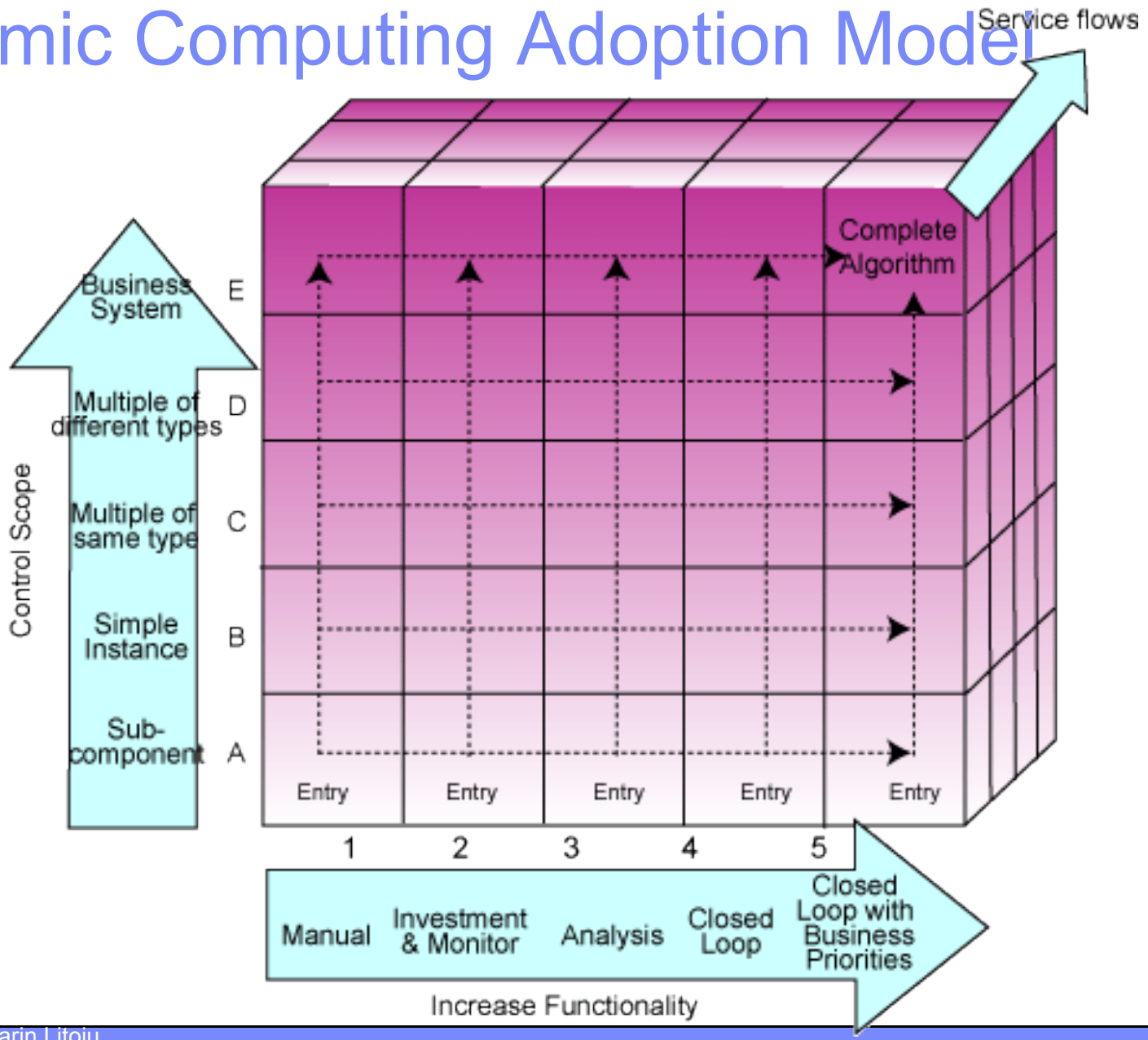
Manual → Autonomic

Evolution; Not Revolution

Characteristics / Benefits	Basic **Level 1**	Managed **Level 2**	Predictive **Level 3**	IT Autonomic **Level 4**	Business Autonomic **Level 5**
	Multiple sources of system generated data	Consolidation of data and actions through management tools	System monitors, correlates and recommends actions	System monitors, correlates and takes action	Integrated components dynamically managed by business rules/policies
	Requires extensive, highly skilled IT staff	IT staff analyzes and takes actions	IT staff approves and initiates actions	IT staff manages performance against SLAs	IT staff focuses on enabling business needs
		Greater system awareness Improved productivity	Reduced dependency on deep skills Faster/better	Balanced human/system interaction IT agility and	Business policy drives IT management Business agility

Manual → **Autonomic**

Autonomic Computing Adoption Model

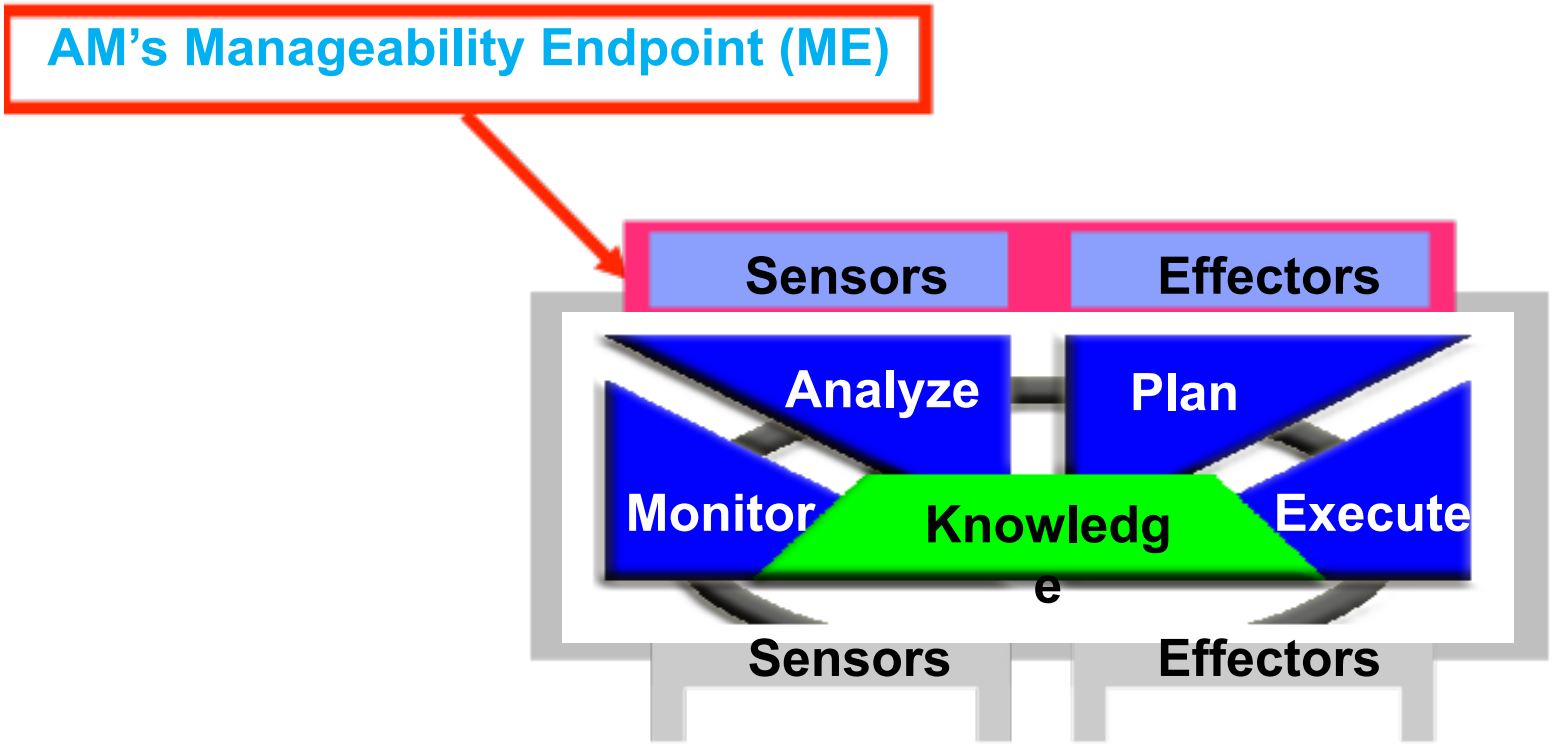


AC Reference Architecture

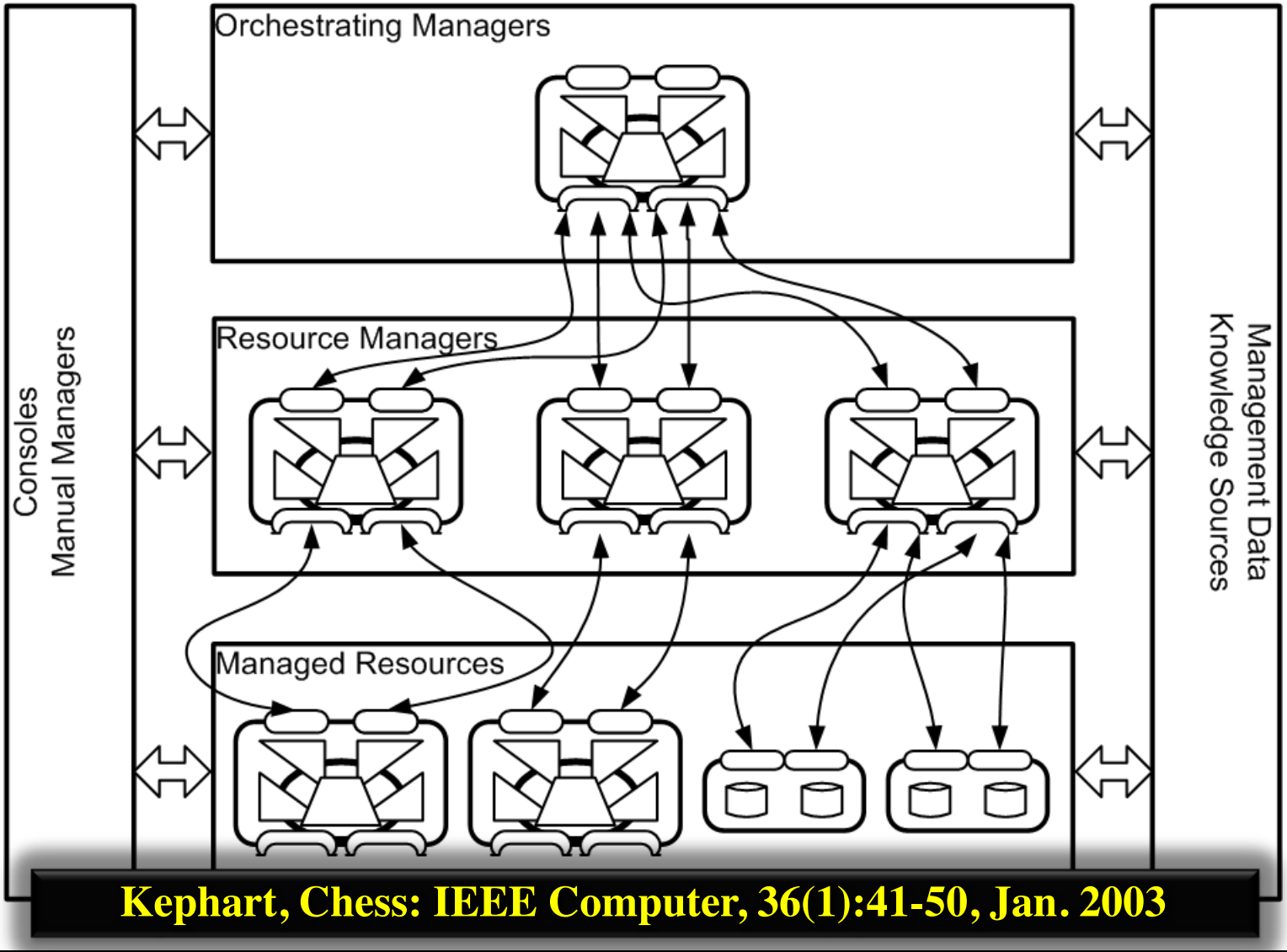
- **An Autonomic System (AS) consists of a collection of Autonomic Elements**
- **An Autonomic Element (AE)**
  - Contains resources and delivers services to humans or other autonomic elements
  - Manages its behaviour in accordance with policies that humans or other AEs have established
  - Acts like an agent
    - Autonomous, proactive, goal-directed
    - Interacts with environment

Kephart, Chess: IEEE Computer, 36(1):41-50, Jan. 2003

Autonomic Manager is a Managed Element



AC Hierarchical Reference Architecture



Kephart, Chess: IEEE Computer, 36(1):41-50, Jan. 2003

Hierarchical MAPE Architecture [2]

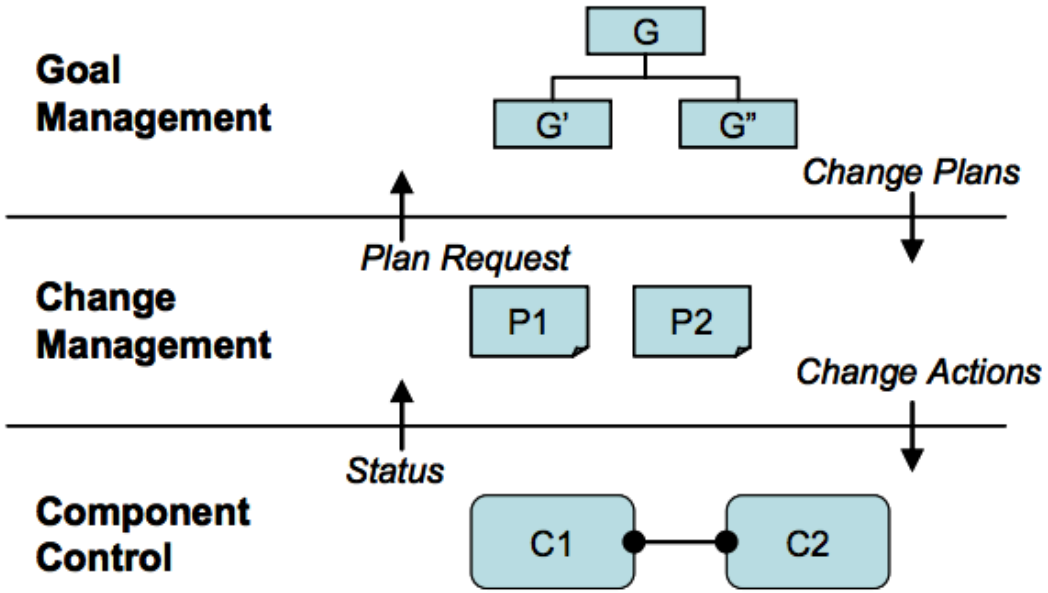


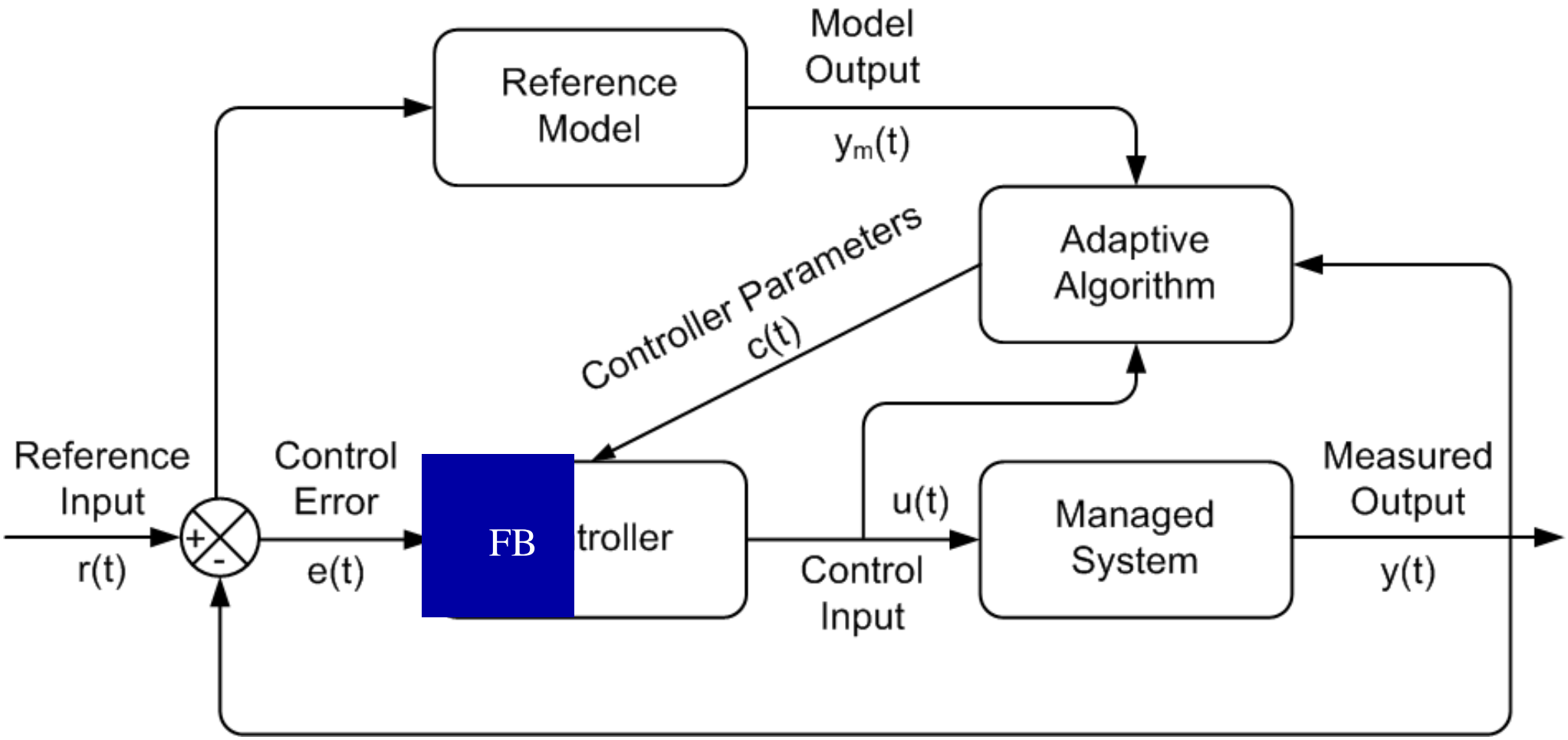
Figure 1 – Three Layer Architecture Model for Self-Management.

Architectures inspired by control theory [1]

Model Reference Adaptive Controllers—MRAC

- **Also referred to as Model Reference Adaptive System (MRAS)**
- **Closed loop controller with parameters that can be updated to change the response of the system**
- **The output of the system is compared to a desired response from a reference model (e.g., simulation model)**
- **The control parameters are updated based on this error**
- **The goal is for the parameters to converge to ideal values that cause the managed system response to match the response of the reference model.**

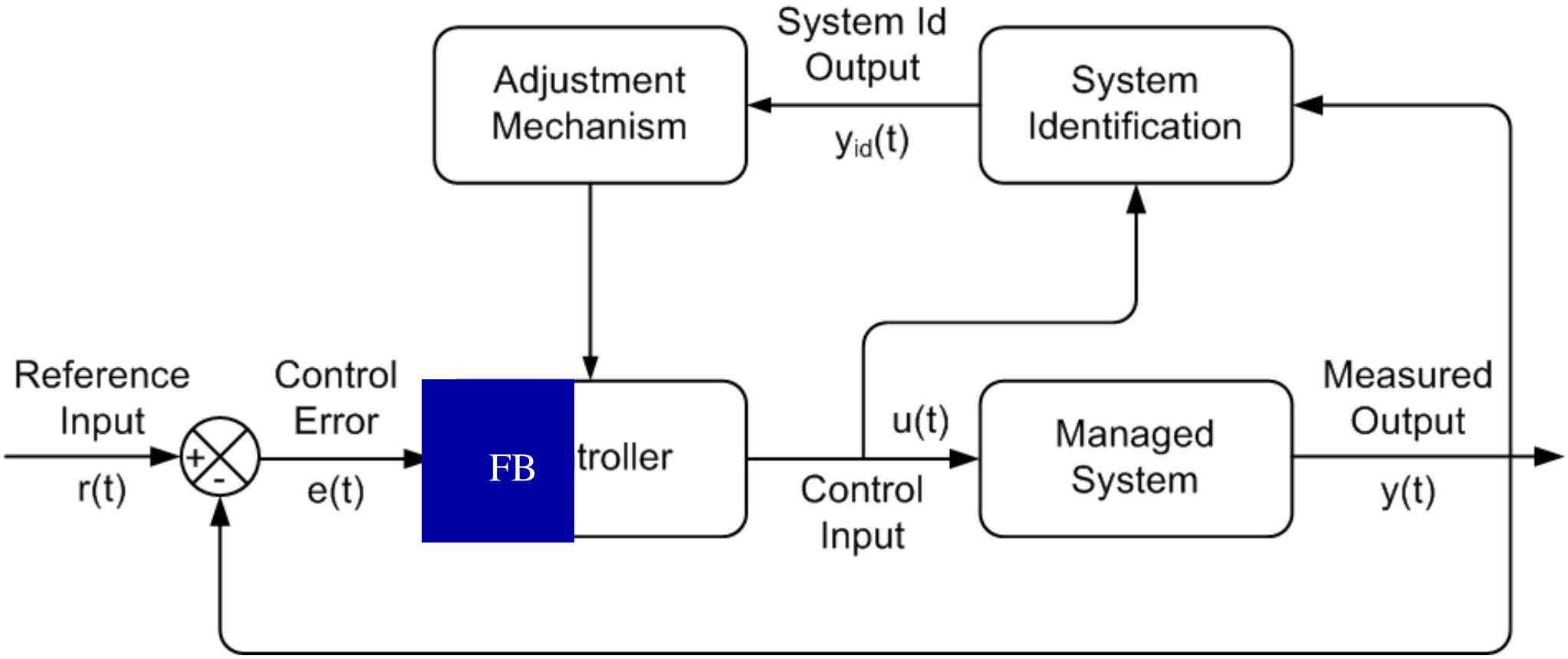
Model Reference Adaptive Controllers—MRAC



Model Identification Adaptive Controllers—MIAC

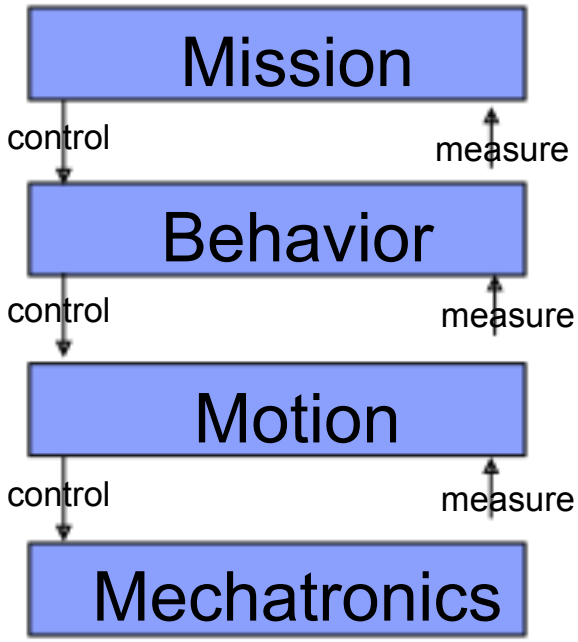
- **Perform system identification while system is running to modify the control laws**
  - Create model structure and perform parameter estimation using the Least Squares method
- **Cautious adaptive controllers**
  - Use current system identification to modify control law, allowing for system identification uncertainty
- **Certainty equivalent adaptive controllers**
  - Take current system identification to be the true system, assume no uncertainty
  - Nonparametric adaptive controllers
  - Parametric adaptive controllers

Model Identification Adaptive Controllers—MIAC



Feedback in CPS/IoT can be Hierarchical, too[1,2]

Hierarchical Autonomic Manager



Managed Thing

Computes paths from A to B; reasons about the cost (risk and time);ranks the paths

Sequence of states: ”Drive-down-road”, “handle intersection,” etc..

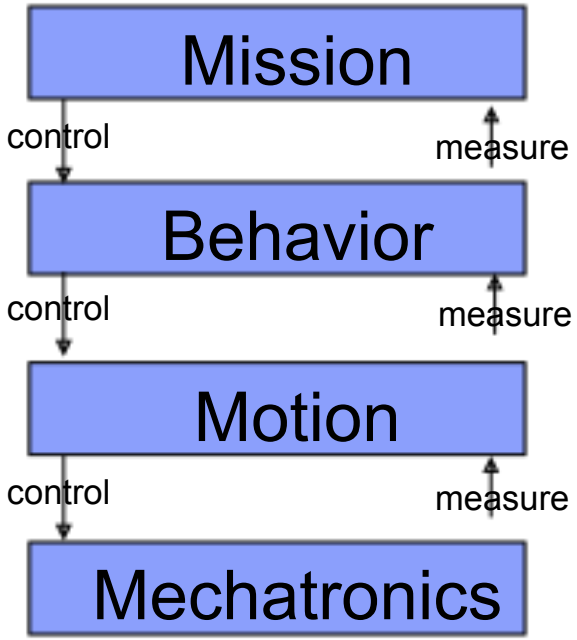
Controls speed, direction

Sensors and Actuators

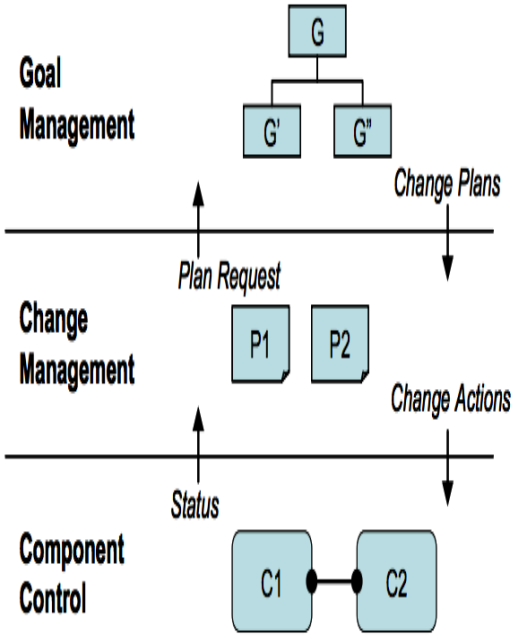
1.Tartan Racing: A Multi-Modal Approach to the DARPA Urban Challenge, 2007
2.Rod Brookes, Elephants Don't Play Chess, 1990

Feedback loops for Smart IoT

Hierarchical Control for PS



Autonomic Control for Software



Further readings (mandatory)

- **Y Brun, GDM Serugendo, C Gacek, H Giese, H Kienle… -Engineering self-adaptive systems through feedback loops in Software engineering for self-adaptive systems, 2009**
- **Kramer, J., Magee, J.: Self-managed Systems: An Architectural Challenge. In: Future of Software Engineering (FoSE 2007), pp. 259-268, IEEE Computer Society, Washington, DC, USA (2007)**

Conclusions

- **Autonomic computing is about self-managed systems**
  - Increase resilience and improve QoS
  - Increase the Return On Investment (skills, maintenance)
- **Autonomic computing is rather evolutionary than revolutionary**
- **Architecture**
  - MAPE-K
    - Can be decentralized or centralized
    - Can be hierarchical
    - In essence is a feedback, but the focus is on components
    - IoT Complex Systems have two main feedback (MAPE-k) loops
      - PS loop controls the “thing”
      - CS loop controls the software
      - How would we manage loop interactions?