

# An Architectural Approach to Autonomic Computing

Steve R. White,  
James E. Hanson,  
Ian Whalley,  
David M. Chess,  
and Jeffrey O. Kephart  
*IBM Thomas J. Watson Research Center*



*Presented as a class work by:  
Eiman Alshammari*

# Outline



**What is the problem?**



**Behavioral properties of autonomic elements.**



**Interfaces and interactions among autonomic elements.**



**Building a system with autonomic behaviors**

# Problem

Collecting Autonomic Elements  $\neq$  Autonomic System

# Solution

Right architecture

=

Achieving autonomic behavior at the system level.

=

Autonomic elements to implement the desired function

+

Additional autonomic elements to enable the needed system-level behaviors

+

Design patterns for system self-management

# Autonomic element behaviors

## Required behaviors:

- Self-management
  - Self Configuration
  - Self Healing Optimizing its own behavior
  - Self Protection
  - Local Problem Handling

# Autonomic element behaviors

## Required behaviors:

- Relationships (Establish & Maintain)
  - Describe its service accurately
  - Understand and being understood
  - Abide by the terms of its agreements (Policy).
  - Capable of negotiating

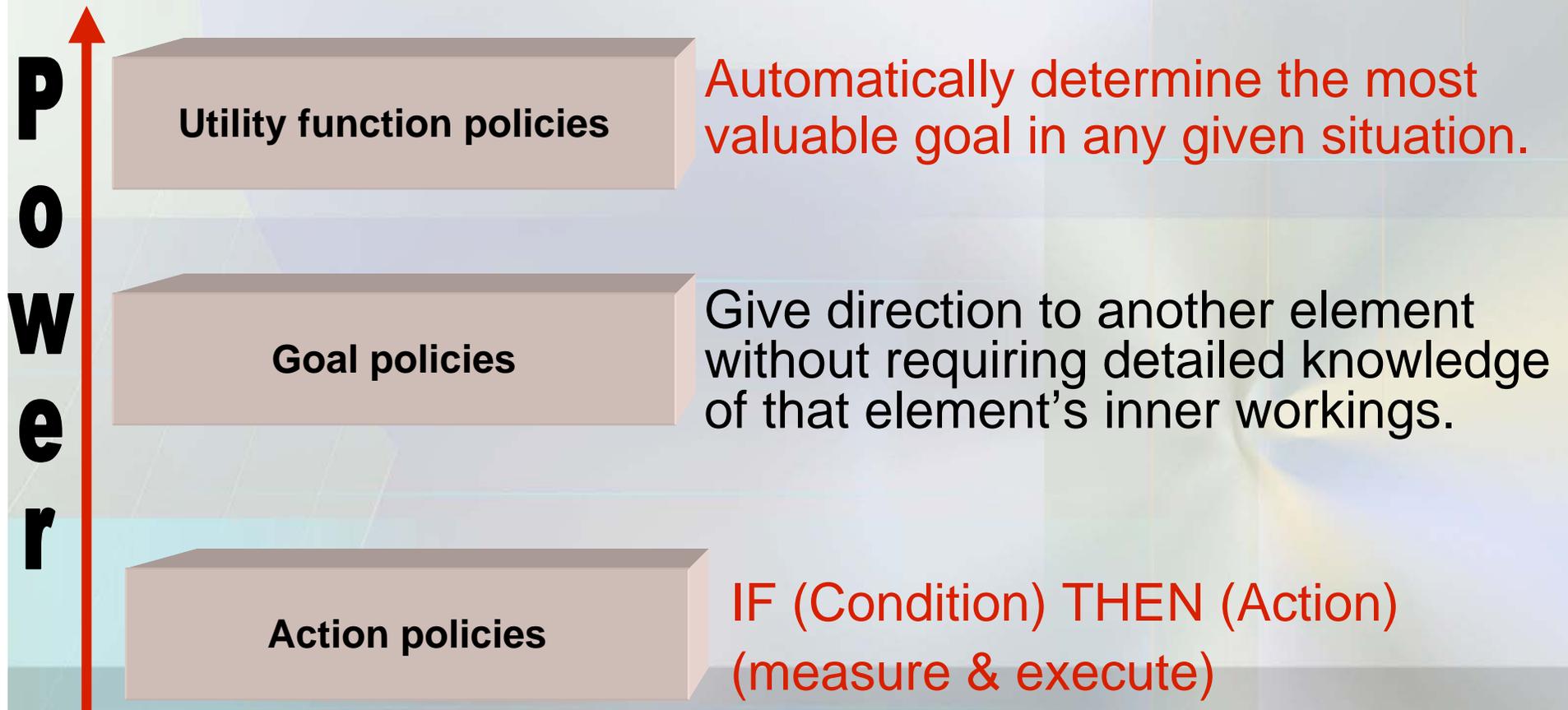
### *Obligations*

- Terms of its agreements.
- Receiving and Abiding by policies.

# Recommended behaviors

- *Not required but strongly encouraged.*
- Ask for a realistic set of requirements
- Offer a range of performance, reliability, availability and security associated with its service.
- Translate requirements (self-assembly)
- Protect itself against inappropriate service requests and responses. (Authentication)

**Policies = A representation of desired behaviors or constraints on behavior.**



# Autonomic Element Interactions

- Interfaces
  - Monitoring and Test Interfaces (allow monitoring)
  - Lifecycle interfaces (determine life cycle)
  - Policy interfaces (sending new policies)
  - Negotiation and binding interfaces (request service)
- Relationships
  - Invoked at Run-time and are a result of Negotiations.
- Interaction Integrity
  - Autonomic Elements communicate with other entities via Service Interfaces.
  - Comm. Within an element not accessible in any way outside of the element.

# Infrastructure elements

Assembling an autonomic system requires:

- Registry
  - Find another element
  - Publish capability of an element
- Sentinel
  - Monitoring services to other elements.
- Aggregator
  - Combines two or more elements to provide improved service.
- Broker
  - Assists an element in carrying out tasks requiring complex relationships.
- Negotiator
  - For complex negotiations.

# Design Patterns

- Self-configuration
  - Relations are constructed dynamically
  - Local Management
- Self-protection
- Self-healing
- Self-optimization

# Self Protection Design Patterns

- Principle: System as a whole should function even if individual element fails.
  - Protection against undesirable system behavior.
  - Protection against system penetration by attackers.
- Access control.
- Security policies

# Self Healing

Functionality Interfaces

End-to-End availability Mgmt Interfaces

# Self Optimization Design Patterns & Interfaces

Use of Market like Mechanisms to resolve  
conflicts

Resource-Arbiter

Parallel Clusters

Interfaces

# Conclusion

The Paper Described an architectural approach for composing autonomic elements to form autonomic systems.