An Architectural Approach to Autonomic Computing

CBSS Course Presentation
Provided by: Sepinood H. Gashti
July 2005

References


- Architectural blueprint for autonomic computing, IBM 2003


Outline

- Introduction
- Autonomic Computing
- Autonomic System Architecture
  - Autonomic Element
  - Interaction
  - Infrastructure Elements
- Common Design Patterns
- Related Works
- Summary

Introduction

- Management problems
  - Cost & time for administration and troubleshooting
  - Complexity crisis
- Self-managing system
  - Behaviour specified by sys admins via high-level policies
  - System and its components figure out how to carry out policies
Autonomic Computing

- A computing system that senses its operating environment, models its behaviour in that environment, and takes action to change the environment or its behaviour.

- A Self-managing autonomic system has the properties of self-configuring, self-healing, self-optimization and self-protection. (self-CHOP)

- Inspired from autonomous behaviour of the human body

Self-CHOP

- 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
Reference Architecture

Autonomic System Architecture

- Autonomic element
- External interfaces and behaviour for individual component
- Interaction among components
Autonomic Systems

How to build a system with autonomic behaviour?

1. A collection of autonomic elements that implement the desired function.
2. Additional autonomic elements to implement system functions that enable the needed system level behaviours.
3. Design patterns for system self-CHOP.

Autonomic Element (AE)

- AEs are responsible for:
  - Managing their own behavior
  - Interacting with other autonomic elements

- AEs contain:
  - one autonomic manager
  - Zero or more managed element(s)
Managed Elements

Controlled through its sensors and effectors by manager

Manager obtains details through its sensor.
Two Modes:
1. Manager requests details
2. Resource provides details

Manager changes details through its effector.
Two Modes:
1. Manager initiates the change
2. Resource requests the change

Properties: Identification, Metrics, State, Configuration, Relationships, Host, Uses, Commands, Logs, Events, APIs

Autonomic Manager

A component that implements the control loop

Analyses observed situations to determine if some change needs to be made.

Autonomic Manager

Creates or selects a plan to make a desired change

Monitors knowledge and executes

Shares accumulated knowledge across the elements.

Makes the changes by performing the plan

Single Resource
Heterogeneous Group
Homogeneous Group
Business System
Knowledge

- Knowledge Types
  - Solution Topology Knowledge
  - Problem Determination Knowledge
  - Policy Knowledge

- Two Mechanisms to obtain knowledge
  - Effector Interface
  - Monitor Part

Policy

- Policy
  - Action policy: If (Condition) THEN (Action)
  - Goal policy: direction
  - Utility function policy: Priority
Interaction Between Components

- Interfaces
  - Monitoring and testing interfaces
  - Lifecycle interfaces
  - Policy interfaces
  - Negotiation interfaces

Infrastructure Elements

Assist other element in doing their tasks

- Sentinel
- Aggregator
- Broker
- Negotiator
Design Patterns

- Self-Configuring
  - Goal driven self-Assembly
    - Initialization
    - Registration

- Self-Healing
  - Self-regenerating cluster
  - Required interfaces
    - Sending state
    - Receiving state,
    - Querying planned outage (Availability management)
    - Scheduling planned outages (Availability management)

Design Patterns (2)

- Self-optimization
  - Market-based control (buyer & seller)
  - Resource arbiter
  - Required interfaces
    - Query service
    - Query service level bounds
    - Requesting a service level

- Self-Protecting
  - Some aspects are similar to self-healing
  - Prevention: Policy-based management
    - Security policy
Related Works


- Richard Anthony, Alun Butler, Mohammad Ibrahim. *Layered Autonomic Systems*, University of Greenwich, ICAC’05

Summary

- Needs for AC
- Describing a proposed architecture for AC
  - AE
  - Infrastructure AE
  - Design patterns for self-CHOP
An Architectural Approach to Autonomic Computing

CBSS Course Presentation
Provided by: Sepinood H. Gashti
July 2005