DEPARTMENT OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING
CONCORDIA UNIVERSITY
Montreal, Qc, Canada

Towards an Autonomic Element Architecture for ASSL

(SEAMS 2007)
ICSE 2007 Workshop on
"Software Engineering for Adaptive and Self-managing Systems"
Minneapolis, MN, USA
May 26-27, 2007

by Emil Vassev & Joey Paquet

Outline

Index

- Problem Statement
- Autonomic Computing Background
- Related Work
- ASSL Multi-Tier Specification Model
- AE Architecture for ASSL
- Conclusion & Future Work

Problem Statement

Current Situation:
Tremendous increase of application complexity:

- Service-Oriented architecture running on grids.
- Multi-core CPUs, integrated communication and management.

Solution:
Self-adaptive and autonomic computing (AC) systems.

Problem:
Crucial need of programming techniques and technologies:

- imply AC principles;
- provide us with programming concepts for implementing autonomic systems (AS).

Autonomic Computing – State of the Art

Four basic self-managing objectives:

- self-configuration;
- self-optimization;
- self-healing;
- self-protection.

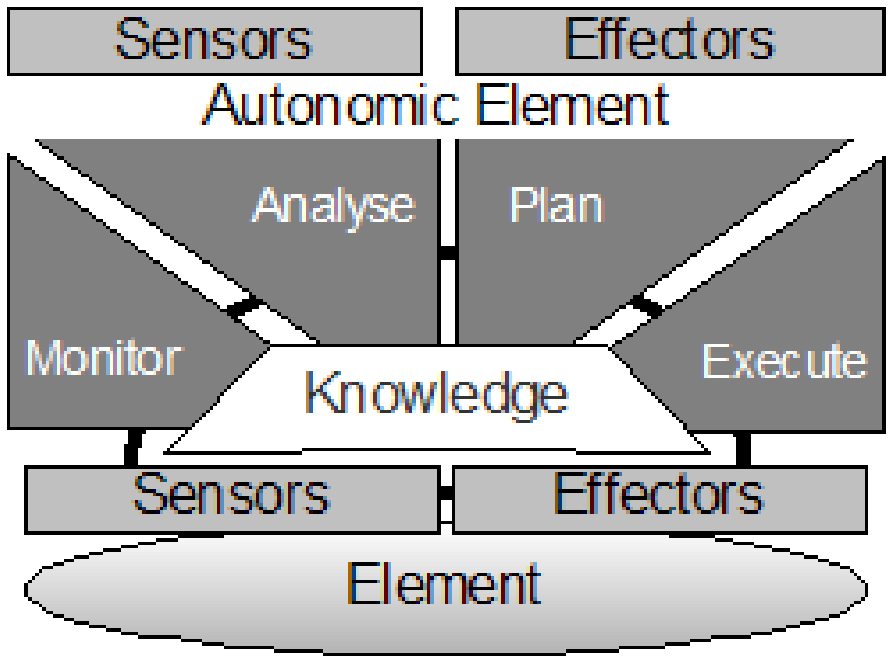
Self-Protection. The AS defends itself from accidental or malicious external attacks, which requires awareness of potential threats and the means to manage them.

Related Work

IBM Research AE Architecture Model

Characteristics:

- Control loop - functionally related units - monitor, analyzer, planner, and executor; all of them sharing knowledge;
- Manageability Interface;
- Managed element.



ASSL Multi-Tier Specification Model

Autonomic System
- AS Service Level Objectives
- AS Self-Management Policies
- Metric Space
- Architecture

Autonomic System Interaction Protocol (ASIP)
- Public AS Messages & Negotiation Protocol
- Public Communication Channels
- Public Communication Functions

Autonomic Element
- AE Service Level Objectives
- AE Self-Management Policies
- Friends
- Autonomic Element Interaction Protocol (AEIP)
  - Private AE Messages & Negotiation Protocol
  - Private Communication Channels
  - Private Communication Functions
  - Managed Resource Interface
- Recovery Protocol
- Behavior
- Outcomes
- Actions
- Events
- Metrics Space

Rationale

- scalable specification model;
- provides judicious selection and configuration of infrastructure elements and mechanisms of ASs;
- decomposes an AS in two directions – 1) into levels of functional abstraction; 2) into functionally related sub-tiers;
- presents the system from three different perspectives – 1) AS; 2) ASIP; 3) AE;
- matches the AS builder needs;
- flexible approach to specification.

Autonomic System Tier

Autonomic System

- AS Service Level Objectives
- AS Self-Management Policies
- Metric Space
- Architecture

AS Service-Level Objectives

- A higher-level form of behavioral specification that establishes objectives (ex. performance), eventually leading the system to determine the actions required to achieve those objectives.
- Our concept assumes that the AS service level objectives (AS SLO) are a global task whose realization is to be distributed among the AEs.
- Each AE has its own service level objectives (AE SLO), which are subordinates of the AS SLO.
- ASSL specifies the AS SLO as correlations between the AEs' SLO.

ASSL – Autonomic System Tier

```
ASSELF_MANAGEMENT {
  SELF_OPTIMIZING {
      SWITCH: ON;
      PRIORITY: 1;
      EVENT lowPerformance: forall AE {
            AE.METRIC_SPACE.performance <= 200;
      }
      EVENT normPerformance: forall AE {
            AE.METRIC_SPACE.performance > 200;
      }
      FLUENT inLowPerformance {
            INITIATES: lowPerformance;
            TERMINATES: normPerformance;
      }
      MAPPING {
            CONDITION: inLowPerformance;
            ACTION {
                  foreach AE in AS { AE.ACTIONS.spawnWorker }
            }
      }
  }
  ....
}
```

AS Metric Space

The AS-level metrics can be:

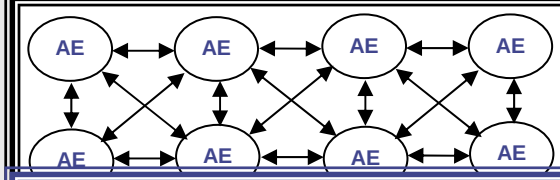
- resource metrics – measure resources;
- quality metrics – measure qualities like performance, response time etc;
- scalar metrics – measure predefined dynamic AS variables.

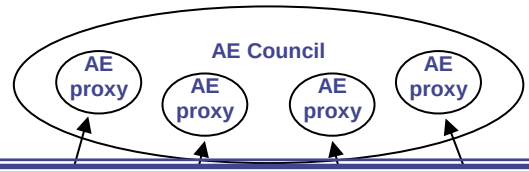
We express metrics by specifying:

- a type of the metric - it could be resource, quality, scalar, or composite;
- a description of the metric;
- a measure unit of the metric – seconds for time, MB for size etc.;
- threshold classes specifying the range of valid values;
- logical expressions over the threshold classes if needed.

ASSL – Autonomic System Tier

AS Architecture





AE proxy advantages:

1. Increase the flexibility and simplicity of our architecture - the AEs are able to deal with their councils as a normal AE.
2. Increase security - can manage secure connection between the AE and the council.
3. Can be used for forensics - AE proxies could be coupled with the recovery protocol, and can be used in analysis of situations in which there was a malicious attack on the system, or there was a crash.

Autonomic System Interaction Protocol (ASIP) Tier

Autonomic System Interaction Protocol (ASIP)

- Public AS Messages & Negotiation Protocol
- Public Communication Channels
- Public Communication Functions

Public AS Messages & Negotiation Protocol

- AEs can exchange information in the form of messages.
- Public AS messages - recognizable by all the AEs.
- AEs communicate them over public channels.
- AEs use empty ASSL structures for queries and their complete homolog for responses.

- Negotiation Protocol – a set of messages needed to start and close a message-exchanging session.

ASSL – Autonomic System Interaction Protocol

Public Communication Channels

- abstract means of communication, connecting AEs;

```
ASIP {
    MESSAGES {....}
    CHANNELS {
        FINAL CHANNEL poster {
            ACCEPT: ID/ANY;
            ACCESS: SEQUENTIAL/ DIRECT;
            DIRECTION: IN/OUT/BYDIRECTIONAL;
        }
        CHANNEL urgency {....}
    }
    FUNCTIONS {….}
}
```

Public Communication Functions

- routines using the communication channels to propagate messages among the AEs;
- a communication function could use a single or multiple channels, or operate in unicast or broadcast mode;

Example:

A routine can propagate a message to all the input and bidirectional public channels, or choose to rely on a bus to send very large messages.

- public communication functions – common to all the AEs.

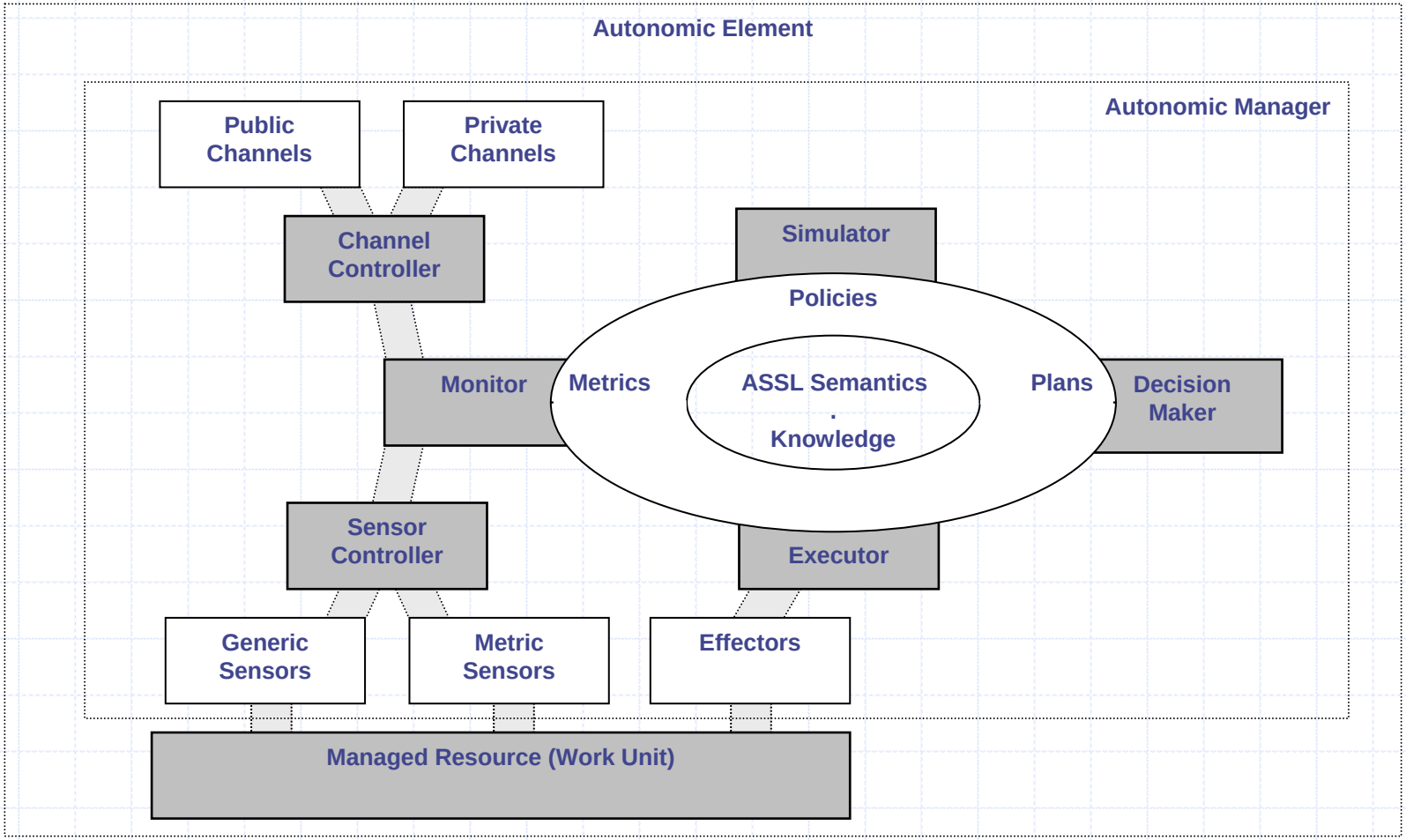
Autonomic Element Tier

Autonomic Element

- AE Service Level Objectives
- AE Self-Management Policies
- Friends
- Autonomic Element Interaction Protocol (AEIP)
  - Private AE Messages & Negotiation Protocol
  - Private Communication Channels
  - Private Communication Functions
  - Managed Resource Interface
- Recovery Protocol
- Behavior
- Outcomes
- Actions
- Events
- Metrics Space

AE Architecture for ASSL

ASSL AE Architecture Model



Public & Private Channels

- The AE uses a set of channels to communicate with other AEs or environmental entities.
- Specified by ASIP and AEIP.

Why communication channels are a better means of communication than the IBM's set of sensors and effectors?

1. **We explicitly specify the messages they can accept - prevents malicious attacks via unspecified messages.**
2. **Private channels enhance the security level - only AE's friends can use the private channels.**
3. **Channels can be used for forensics, i.e. for gathering evidence - can be used in analysis of situations in which there was a malicious attack on the system.**

Channel Controller

- Operates the channels - can modify channels' settings, close, open, create channels or arrange channels into busses.
- Responsible for sending and receiving messages over the communication channels - implements the communication functions specified by the ASIP and AEIP.

Advantages?

1. **Brings great flexibility to channel management - takes care about the runtime changes in the channels.**
2. **Achieves the separation of the abstract concepts of pure messaging and the message operation implementation - sending, receiving, breaking large-sized messages into chunks for transmitting them over busses and so on.**

Metric and Generic Sensors

- Metric sensors measure specific metrics.
- Both kinds operate over the managed element.
- Generic Sensors explore the metric space and discover new metrics .
- The notion of generic sensors is still under discussion.

Advantage of having both Metric and Generic sensors?

Flexibility - our model uses predefined metric sensors built for measuring metrics known at design time, but also uses generic sensors for measuring metrics unknown from the initial specification.

Sensor Controller

- Controls the metric and generic sensors, by tuning and operating them.
- AEs could implement algorithms for optimizing the metric space.

Advantage?

Flexibility - we separate the concept of measuring metrics from the concept of operating measurement tools.

Effectors

- Manageability interface used by the executor to control the managed element.

Advantage?

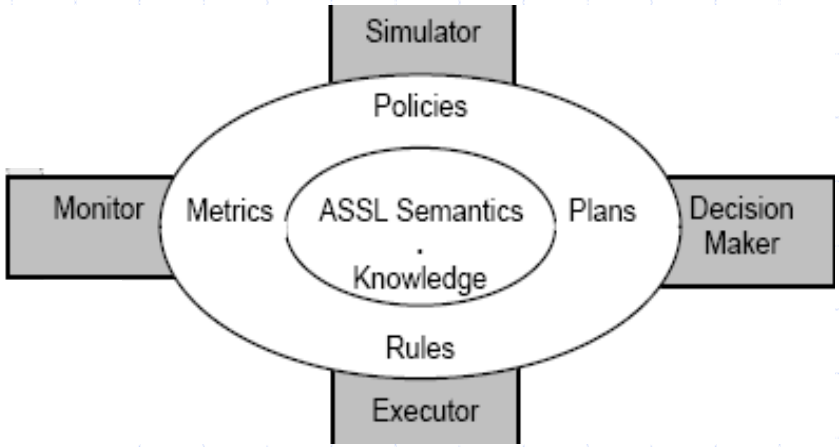
Effectors could also be used for forensics. Any function call can be registered together with the returned result or raised exception, thus helping in further analysis.

AE Architecture for ASSL

AE Control Loop

- Monitor, Simulator, Executor, and Decision Maker.
- ASSL Core & ASSL Bus.

Advantage?



In our architecture model, we separate simulation from analysis and planning, thus increasing the overall performance and manageability.

Conclusion & Future Work

- ASSL - a framework that provides a multi-tier specification model for AS.
- ASSL allows expressing ASs, as a set of interacting AEs, at three main levels - AS level, ASIP level, and AE level.
- The new architecture model for AE emphasizes on the ASSL specification - forms the shared knowledge.
- The arch. model can be validated before building the system.

- Currently working on specifying two systems – DMS (distributed) and ASTRM (reactive).
- Next: 1) developing the ASSL framework; 3) validating the ASSL framework – generating AS DMS.
- Possible collaboration with NASA – ANTS project.

THANK YOU!

Questions?