표준 기반 자율 시스템 관리 기술

안창원*, 박종근*
*한국전자통신연구원, 디지털홈연구단

Standard-based Autonomic System Management

Ahn, Chang-Won° , Park, Jong-Geun°
Digital Home Division, ETRI
E-mail : {ahn, queue}@etri.re.kr

Abstract

An autonomic system management technology is being developed for self-managing during deployment and on an on-going basis for a production environment so that the system may be deployed and managed in production with minimal human intervention. As networks and distributed systems grow and change, system deployment failures, hardware and software issues, and human error can increasingly hamper the performance and capacity of the components in an IT system, driving up overall costs-even as technology component costs continue to decline. Known as the only solution for an on demand IT environment, the architecture of the autonomic system management will be shown and also the corresponding standards on the way will be introduced in this paper.

1. Introduction

In an on demand business, information technology (IT) professionals must strengthen the responsiveness and resiliency of service delivery—providing seamless service and improving quality of service—while reducing the total cost ownership (TCO) of their operating environments.

However, IT components produced over the past decades are so complex that IT professionals are challenged to effectively operate a stable IT infrastructure. It's the complexity of the IT components themselves that have helped fuel this problem. As networks and distributed systems grow and change, system deployment failures, hardware and software issues, and human error can increasingly hamper the performance and capacity of the components in an IT system, driving up overall costs-even as technology component costs continue to decline.

Software developers have fully exploited a magnitude increase in computational power-producing ever more sophisticated software applications and environments. Exponential growth has occurred in the number and variety of systems and components.

The value of database technology and the Internet have fueled significant growth in storage subsystems, which now are capable of holding petabytes of structured and unstructured information. Networks have interconnected distributed, heterogeneous systems.

Our information society has created unpredictable and highly variable workloads for these networked systems. And these increasingly valuable, complex systems require highly skilled IT professionals to install, configure, operate, tune and maintain them.

2. Autonomic Computing

Autonomic Computing helps to address complexity by using technology to manage technology. The term *autonomic is derived from human biology.* The autonomic nervous system monitors your heartbeat, checks your blood sugar level and keeps your body temperature close to 36.5 ℃ without any conscious effort on your part. In much the same way, self-managing autonomic capabilities anticipate IT system requirements and resolve problems with minimal human intervention. As a result, IT professionals can focus on tasks with higher value to the business.

However, there is an important distinction between autonomic activity in the human body and autonomic activities in IT systems. Many of the decisions made by autonomic capabilities in the body are involuntary. In contrast, self-managing autonomic capabilities in computer systems perform tasks that IT professionals choose to delegate to the technology according to policies. Adaptable policy determines the types of decisions and actions that autonomic capabilities perform.

IT businesses organize these tasks as a collection of best practices and processes such as those defined in the IT Infrastructure Library (ITIL). The more these tasks can be automated, the more opportunity for IT professionals to delegate the management of the IT infrastructure to itself.

The efficiency and effectiveness of typical IT processes are measured using metrics such as elapsed time to complete a process, percentage executed correctly and the cost to execute a process. Self-managing systems can positively affect these metrics, improving responsiveness and quality of service, reducing TCO and enhancing time to value through rapid process initiation and reduced time and skill requirements.

These intuitive and collaborative characteristics of the self-management capabilities enable businesses to operate their business processes and IT infrastructure more efficiently with less human intervention, *decreasing costs and enhancing the organization*’s ability to react to change. For instance, a self-managing system could simply deploy a new resource and then tune the environment to optimize the services delivered by the new resource. This is a notable shift from traditional process that requires a significant amount of analysis before and after deployment to ensure that the resource operates effectively and efficiently.

3. Architecture Concepts

An autonomic computing system is organized into the layers and parts shown in Figure 1. These parts are connected using enterprise service bus patterns that allow the components to collaborate using standard mechanisms such as Web services. The enterprise service bus integrates the various blueprint building blocks, which include:

- Touchpoints for managed resources
- Knowledge sources
- Autonomic managers
- Manual managers

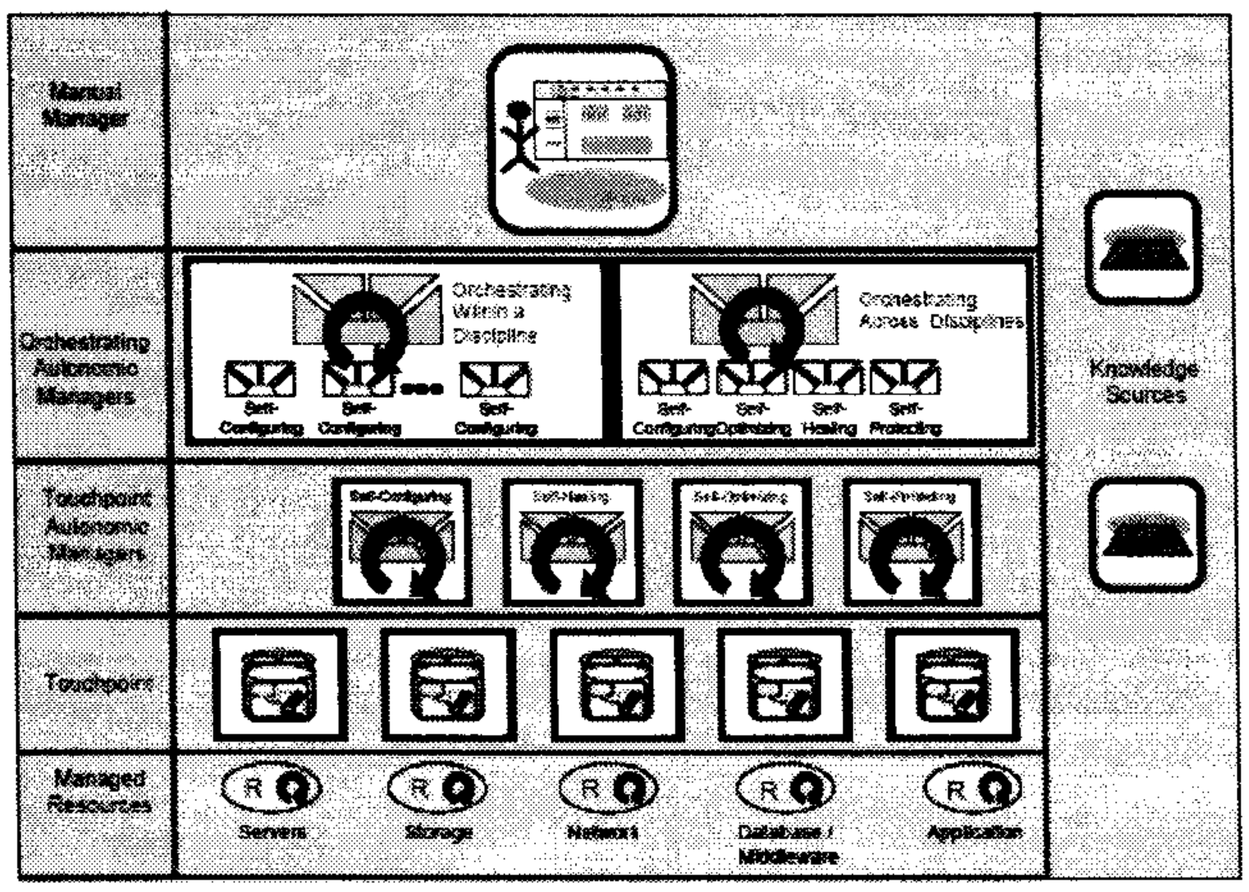

Figure 1. Autonomic Computing Architecture

The lowest layer contains the system components, or managed resources, that make up the IT infrastructure. These managed resources can be any type of resources (hardware and software) and may have embedded self-managing attributes. The next layer incorporates consistent, standard manageability interface for accessing and controlling the managed resources. These standard interfaces are delivered through a touchpoint. Layers three and four automate some portion of the IT process using an autonomic manager.

Layer four contains autonomic managers that orchestrate other autonomic managers. It is these orchestrating autonomic managers that deliver the systemwide autonomic capability by incorporating control loops that have the broadest view of the overall IT infrastructure. The top layer illustrates a manual manager that provides a common system management interface for the IT professional using an integrated solutions console. The various manual and autonomic manager layers can obtain and share knowledge via knowledge sources.

Managed resources

A managed resource is a hardware or software component that can be managed. A managed resource could be a server, storage unit, database, application server, service, application or other entity. As shown in Figure 1, a managed resource might contain its own embedded self-management control loop, in addition to other autonomic managers that might be packaged with a managed resource.

Touchpoint

A touchpoint is an autonomic computing system building block that implements sensor and effector behavior for one or more of a managed resource's manageability mechanisms. It also provides a standard manageability interface. Deployed managed resources are accessed and controlled through these manageability interfaces. Manageability interfaces employ mechanisms such as log files, events, commands, application programming interfaces (APIs) and configuration files. These mechanisms provide various ways to gather details about and change the behavior of the managed resources.

Touchpoint autonomic managers

Autonomic managers implement intelligent control loops that automate combinations of the tasks found in IT processes. Touchpoint autonomic managers are those that work directly with the managed resources through their touchpoints. These autonomic managers can perform various self-management tasks, so they embody different intelligent control loops.

Most autonomic managers use policies (goals or objectives) to govern the behavior of intelligent control loops. Touchpoint autonomic managers use these policies to determine what actions should be taken for the managed resources that they manage.

Orchestrating autonomic managers

A single touchpoint autonomic manager acting in isolation can achieve autonomic behavior only for the resources that it manages. The self-managing autonomic capabilities delivered by touchpoint autonomic managers need to be coordinated to deliver systemwide autonomic computing behavior. Orchestrating autonomic managers provide this coordination function.

An example of an orchestrating autonomic manager is a workload manager. An autonomic management system for workload might include self-optimizing touchpoint autonomic managers for particular resources, as well as orchestrating autonomic managers that manage pools of resources. A touchpoint autonomic manager can optimize the utilization of a particular resource based on application priorities. Orchestrating autonomic managers can optimize resource utilization across a pool of resources, based on transaction measurements and policies. The philosophy behind workload management is one of policy-based, goal-oriented management.

Manual Managers

A manual manager provides a common system management interface for the IT professional using an integrated solutions console. Self-managing autonomic systems can use common console technology to create a consistent human-facing interface for the autonomic managers of IT infrastructure components. As indicated earlier, autonomic capabilities in computer systems perform tasks that IT professionals choose to delegate to the technology, according to policies.

In some cases, an administrator might choose for certain tasks to involve human intervention, and the human interaction with the system can be enhanced using a common console framework, based on industry standards, that promotes consistent presentation to IT professionals.

The primary goal of a common console is to provide a single platform that can host all the administrative console functions in server, software and storage products to allow users to manage solutions rather than managing individual components or products. Administrative console functions range from setup and configuration to solution run-time monitoring and control.

4. Standards for Autonomic Computing

The fundamental nature of autonomic computing systems precludes any single company from delivering an entire autonomic solution. IT infrastructures have heterogeneous systems and must deal with heterogeneous environments in the enterprise.

A proprietary implementation would be like a heart that maintains a regular steady heartbeat but can not adjust to the needs of the body when under stress. Self-managing autonomic computing systems require autonomic managers to be deployed across the IT infrastructure, managing various resources including other autonomic managers from a diverse range of suppliers. Therefore, these systems must be based on open industry standards.

One significant development in the area of autonomic computing management standards was the March 9, 2005 announcement by OASIS of the ratification of the WS-DM 1.0 specification. The WS-DM 1.0 specification, available from OASIS, consists of two major parts: Management Using Web Services (MUWS) and Management Of Web Services (MOWS). The specification addresses a broad array of management topics relevant for autonomic computing, including properties, operations, events, capabilities and management interfaces. These WS-DM management topics can be realized in touchpoint manageability interfaces, using sensor and effector interfaces.

Although Web Services are not the only method for accomplishing autonomic computing as indicated earlier, they provide a standard basis for management interfaces, and so the WS-DM 1.0 specification offers an important standards basis for constructing autonomic systems.

Another significant development related to autonomic computing standards is the launch of the OASIS Solution Deployment Descriptor (SDD) technical committee. This committee will define XML schema for SDD, as well as package format to associate SDDs, resource content, and software artifacts. SDDs are intended to describe the aggregation of installable units at all levels of the software stack. The resulting XML schema shall be partitioned to allow for layered implementations covering the range of applications from the definition of autonomic units of software to complex, multi-platform, heterogeneous solutions.

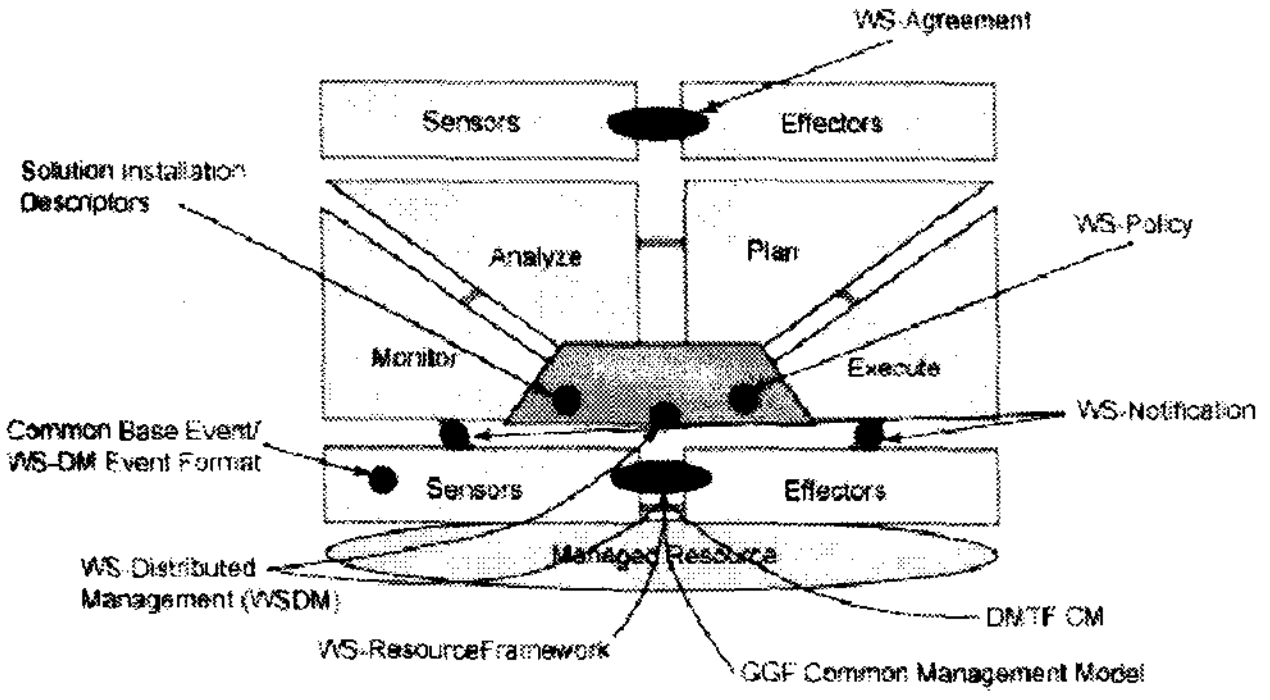

Figure 2 Representative standards used by an autonomic manager

DMTF (Distributed Management Task Force)

DMTF, developer of the Common Information Model (CIM), is the technology industry organization leading the development, adoption and interoperability of management standards and initiatives for enterprise and Internet environments. CIM is the breakthrough standard for the exchange of management information in a platform-independent and technology-neutral way, streamlining integration and reducing costs by enabling end-to-end multi-vendor interoperability in management systems.

CIM-based management in a Web services environment requires that the CIM Schema (classes, properties and methods) be rendered in XML Schema and WSDL (the Web Services Description Language). To achieve this, CIM's definitive MOF (the Managed Object Format) must be mapped to WSDL and XML Schema via an explicit algorithm that can be programmed for automatic translation.

WS-CIM (Web Services Common Information Model) define the Define the algorithmic mapping of CIM MOF to XML Schema (mapping the classes, properties and method parameters) and WSDL fragments (mapping the class methods).

Application Working Group is to develop CIM Schema Definitions of management for Web-based and networked application services. Two committees are organized to produce specifications.

The Application Management Working Committee created the content for the Common Information Model (CIM) data schema and defined extensions to the Software Standard Groups to support distributed applications. This schema models an application, application system, and movement of an application component through the application lifecycle states: deployable, installable, executable, and running. Modeling of management of the 'running' application was out of the scope of this work group. The result of this working group is reflected in the CIM_Application schema versions 2.2 through 2.4.

The Application Management Runtime Working Committee will develop a model for defining and managing the aspects of a 'running' application. This schema will model the configuration, operations, metrics, and status of complex, traditional, client-server, distributed, and web-based type applications. The working group will develop a MOF, Visio, and whitepaper for the new schema. The result of this working group should be submitted for the

CIM_Application schema version 2.5.

Commonly described as "utility", "autonomic", "grid", or "on-demand" computing, consumers require that utility computing solutions be comprised of multi-vendor components and that those components interoperate seamlessly at customer installations. While interoperating, these components must individually and in total express fundamentally improved management functionality, security, reliability, availability, and extensibility to support the Utility Computing (UC) vision.

The goals of the Server Management Working Group are to define a platform independent, industry standard management architecture instantiated through wire level protocols built upon IP based technologies that extend the CIM schema (presenting the work in parallel to the Sys/Dev WG) to represent new server system topologies, leverage the CIM/XML protocol and identify enhancements if necessary and define a CLI protocol (syntax & semantics), profiles for different server system topologies in order to support base-level compliance, and an architectural model for understanding the semantic behavior of server management components.

IETF (Internet Engineering Task Force)

Polciy-Core Information Model presents the object-oriented information model for representing policy information developed jointly in the IETF Policy Framework WG and as extensions to the Common Information Model (CIM) activity in the Distributed Management Task Force (DMTF).

The policy classes and associations defined in this model are sufficiently generic to allow them to represent policies related to anything. However, it is expected that their initial application in the IETF will be for representing policies related to QoS (DiffServ and IntServ) and to IPSec. Policy models for application-specific areas such as these may extend the Core Model in several ways.

The preferred way is to use the PolicyGroup, PolicyRule, and PolicyTimePeriodCondition classes directly, as a foundation for representing and communicating policy information. Then, specific subclasses derived from PolicyCondition and PolicyAction can capture application-specific definitions of conditions and actions of policies. Two subclasses, VendorPolicyCondition and VendorPolicyAction, are also included in this document, to provide a standard extension mechanism for vendor-specific extensions to the Policy Core Information Model.

This document fits into the overall framework for representing, deploying, and managing policies being developed by the Policy Framework Working Group. DMTF standardization of the core policy model is the responsibility of the SLA Policy working group in the DMTF.

OASIS (Organization for the Advancement of Structured Information Standards)

Web services allow applications to communicate across platforms and programming languages using standard protocols based on XML. OASIS is defining many of the infrastructure standards that enable Web services as well as the implementation standards that are used in specific communities and across industries.

The purpose of the Web Services Security (WS-Security) Technical Committee (TC) is to deliver a technical foundation for implementing security functions such as integrity and confidentiality in messages implementing higher-level Web services applications and to continue work on the Web Services security foundations as described in the WS-Security specification, which was written within the context of the Web Services Security Roadmap as published in April

2002. The work of the WS-Security TC will form the necessary technical foundation for higher-level security services which are to be defined in other specifications. The TC shall not further develop the security roadmap, nor shall the roadmap constitute a normative part of the output of the TC.

The Web Services Distributed Management TC is defining Web Services architecture to manage distributed resources with two sets of specifications: Web Services Distributed Management: Management Using Web Services (MUWS) and Web Services Distributed Management: Management Of Web Services (MOWS) specifications.

The purpose of the Web Services Resource Framework (WS-RF) TC is to define a generic and open framework for modeling and accessing stateful resources using Web services. This includes mechanisms to describe views on the state, to support management of the state through properties associated with the Web service, and to describe how these mechanisms are extensible to groups of Web services.

Web Services Notification (WS-N) TC is to define a set of specifications that standardize the way Web Services interact using the Notification pattern. In the Notification pattern a Web service, or other entity, disseminates information to a set of other Web services, without having to have prior knowledge of these other Web Services.

SNIA (Storage Networking Industry Association)

The SNIA is focused on easing end-user challenges through high level knowledge exchange, thought leadership and promoting storage standards. To achieve this, the SNIA runs a coordinated series of activities throughout North American in addition to supporting its worldwide affiliates to provide a central source of unbiased knowledge.

To accelerate the emergence of SANs in the market, the industry requires a standard management interface that allows different classes of hardware and software products supplied by multiple vendors to reliably and seamlessly interoperate for the purpose of monitoring and controlling resources. The SNIA Storage Management Initiative (SMI) was created to develop this specification (SMI-Specification or SMI-S), the definition of that interface. This standard provides for heterogeneous, functionally rich, reliable, and secure monitoring/control of mission critical global resources in complex and potentially broadly distributed multi-vendor SAN topologies. As such, this interface overcomes the deficiencies associated with legacy management.

GGF (Global Grid Forum)

The purpose of the Open Grid Services Architecture (OGSA) Working Group is to achieve an integrated approach to future OGSA service development via the documentation of requirements, functionality, priorities, and interrelationships for OGSA services. Topic areas are common resource model and service domain mechanisms, but the precise set to be addressed will be determined in early discussions.

The GGF Scheduling and Resource Management Area is concerned with various issues relating to resource scheduling and resource management in Grid environments. To make use of distributed resources within the Grid at the same time to solve a problem a Super-Scheduling Service is necessary. Through this service access to and use of various resources managed by different schedulers in use within a Grid will be possible. The Grid Resource Allocation Agreement Protocol (GRAAP) Working Group addresses the protocol between a Super-Scheduler (Grid Level Scheduler) and local Schedulers necessary to reserve and allocate resources in the Grid as a building block for this service.

The purpose of the Open Grid Services Infrastructure (OGSI) Working Group is to review and refine the Grid Service Specification and other documents that derive from this specification, including OGSA-infrastructure-related technical specifications and supporting informational documents. As new set of draft specifications, WS-RF, was released based on the concepts of OGSI and enhanced by experts from the Web Services community in January 2004, these specifications will be submitted to a standards organization in the near future.

The Open Group

The Enterprise Management Forum works to develop a common manageability infrastructure that can be used by both applications developers and management system vendors to create an open management environment in which complex solutions can be constructed without artificial barriers to their management. The Forum provides both specifications and open source reference implementations to enable the creation of such environments, along with applicable test suites and certification programs.

The Enterprise Management Forum defines the C binding for application response measurement (ARM). ARM is a standard for measuring service levels of single-system and distributed applications. ARM measures the availability and performance of transactions, both visible to the users of the business application and those visible only within the IT infrastructure.

5. Conclusions

Autonomic computing is about shifting the burden of managing systems from people to technologies. When the self-managing technologies can collaborate, the elements of a complex IT system can work together and manage themselves based on a shared view of systemwide policy and objectives.

This paper has presented a high-level architecture to assist in delivering autonomic computing near future. The architecture uses intelligent control loop implementations to monitor, analyze, plan and execute, leveraging knowledge of the environment. The architecture reinforces that self-management uses intelligent control loop implementations to monitor, analyze, plan and execute, leveraging knowledge of the environment. These control loops can be embedded in resource run-time environments or delivered in management tools.

Standards are critical to the broad adoption of autonomic computing technology. As illustrated by the recent OASIS announcement of the WS-DM 1.0 standard, the approval of the WS-DM standard is a significant milestone for system management in heterogeneous environments.

Many standards are at play in the information technology industry and many are relevant for autonomic computing technology. Several such standards were presented here, along with background and rationale for using existing and developing new standards for autonomic computing architecture. As autonomic computing technology continues its ascent, it will do so on a foundation of open standards.

[References]

[1] Autonomic Computing, http://www.ibm.com/

[2] DMTF, http://www.dmtf.org/

[3] IETF, http://www.ietf.org/

[4] OASIS, http://www.oasis-open.org/

[5] SNIA, http://www.snia.org/

[6] GGF, http://www.ggf.org/

[7] Open Group, http://www.opengroup.org/