

© [2006] IEEE. Reprinted, with permission, from [Frank Chiang, Robin Braun, Towards a Management Paradigm with a Constrained Benchmark for Autonomic Communications, Computational Intelligence and Security (CIS), 2006 International Conference on Print, 2006]. This material is posted here with permission of the IEEE. Such ermission of the IEEE does not in any way imply IEEE endorsement of any of the University of Technology, Sydney's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org. By choosing to view this document, you agree to all provisions of the copyright laws protecting it

Towards a Management Paradigm with a Constrained Benchmark for Autonomic Communications

Frank Chiang, Robin Braun
Faculty of Engineering
University of Technology Sydney
NSW 2007, AUSTRALIA
frankj@eng.uts.edu.au

Shaowu He
School of Software
XiDian University
Xi'an Shaanxi, CHINA

Abstract

This paper describes a management paradigm to give effect to autonomic activation, monitoring and control of services or products in the future converged telecommunications networks. It suggests an architecture that places the various management functions into a structure that can then be used to select those functions which may yield to autonomic management, as well as guiding the design of the algorithms. The validation of this architecture, with particular focus on service configuration, is done via a genetic algorithm — Population Based Incremental Learning (PBIL). Even with this centralized adaptation strategy, the simulation results show that the proposed architecture and benchmark can be applied to this constrained benchmark, produces effective convergence performance in terms of finding nearly optimal configurations under multiple constraints.

1 Introduction

The management of current telecommunication networks involves a strong reliance on expert intervention from human operators. The centralized infrastructure in traditional network management systems forces the human operators to have wide ranging expertise on how to discover changes, configure services, recover from failures and alarms, and optimize managed resources to maximize QoS, etc. However, the increasing complexity the networks, the highly distributed nature of Network Elements (NEs) as well as the growing multidimensional inter-dependencies between NEs is beginning to indicate that network management is rapidly reaching the point where manual/automatic systems will no longer suffice. Autonomic systems are essential. (By automatic we mean systems that react according to predefined rules. However, by Autonomic we mean systems that create their own adaptation strategies driven

by system objectives.) There is an urgent need to explore the distributed *autonomic* ways to manage future complex distributed electronic environments.

This paper describes a telecommunications management architecture that both acts as a reference to conventional systems and as a guiding structure to potential autonomic action in selected areas. It covers a number of essential functions: adaptive system objectives setting-up, information domains searching, end-to-end monitoring, service discovery, service selection, service composition, service provisioning or activation. This architecture is based on the TMF [1] entity-based 4 layer telecommunications management structure. It does not relate in any way to the ISO 7 layer communications model, except to note that physical message passing between the layers can be accomplished by electronic communications systems based on the 7 layer structure. In a way, it can be seen that the 7 layer structure is orthogonal to our management architecture [2].

The remainder of the paper is organized as follows. Section 2 demonstrates new management paradigm. Section 3 and 4 presents a constrained benchmark structure including information model, objective function and a reference model is presented. As a validation test, the simulation results of email service provisioning in Section 5 show the effectiveness of our autonomic solution to this constrained benchmark framework by the nature-inspired adaptation strategy - PBIL. Finally we conclude the contributions of this paper.

2 A new management paradigm

2.1 Understanding of Autonomic Communication

Self-management is considered as the key characteristic for autonomic communication in Horn's report in 2001 and in IBM redbook at the year 2003, and self-management as a computational vision described by IBM in their auto-

nomonic communication blueprint. As the increasing challenges of pervasive computing and infrastructureless networks (E.g., P2P networks and Wireless Ad-hoc Sensor Networks (WASNs)), self-managed network is playing a key role and is regarded as the solution to these challenges in Pervasive computing and MANET. However, the autonomic scenario is NOT equivalent to the self-management scenario which is tackled at a computational level.

Recent researches on autonomic communication reported by Strassner and Kephart pointed out that autonomy is a higher level notation to the computational level and is therefore more than *self-management*, that is driven by high-level business objectives or being specified by human operators. Although we share same view on this point, we emphasize that only a distributed electronic system with *learning and adaptation strategies* can be called autonomic systems, which can adapt to changing system objectives and circumstances, and satisfy on-demanding business-driven services initiatives. It is the authors' belief that a successful ACN should develop in two directions as illustrated in our paper [3].

We define autonomic communication for this purpose as follows: *Distributed communication systems with the learning and adaptation capability to cope well with dynamic, uncertain and complex environments - that is, immediately adapt their strategies in accordance with high-level business objectives and rules in order to maximize service satisfaction within available services and managed resources.*

2.2 A structure for autonomic behavior

The layered structure lends itself to the selected introduction of Autonomic behavior related to specific functions. For example, the product setup process involves the allocation of specific services to specific product components. In a conventional system, this may be done by the system engineer as part of a system configuration process, according to a set of business and design rules. On the other hand, such a function could be done autonomically at operational time using autonomic adaptation strategies that may perhaps be econometric, or based on trust and reliability or even swarming behavior. An example we have described is the function of configuring MMS mailbox servers to specific customer MMS mailboxes in accordance with the SLA between customers and providers. We carry on this by introducing market force concepts to a number of selected agents residing in the Management Layer [4][5]. In so doing, we allow them to have measures of autonomy, with intelligence, goals and desires, and social awareness.

3 A constrained benchmark structure

3.1 A Reference Model

Our analysis is on the basis of Object-Oriented Principles (OOP). We consider a TMF entity-based network operation system with n_p product instances for c_p classes of products; n_c product component instances for c_c classes of product components; n_s service instances for c_s classes of services; and n_r resource instances for c_r classes of resources. The cost elements between instances of $(n_p \times n_c)$, $(n_c \times n_s)$, $(n_s \times n_r)$ construct link cost matrices, which are assumed to be constant only during one iteration search and vary independently from one iteration to another. We denote the followings:

$$\mathbf{P}_{\{1, \dots, n_p\}}^1, \dots, \mathbf{P}_{\{1, \dots, n_p\}}^{c_p}; \mathbf{C}_{\{1, \dots, n_c\}}^1, \dots, \mathbf{C}_{\{1, \dots, n_c\}}^{c_c}; \\ \mathbf{S}_{\{1, \dots, n_s\}}^1, \dots, \mathbf{S}_{\{1, \dots, n_s\}}^{c_s}; \mathbf{R}_{\{1, \dots, n_r\}}^1, \dots, \mathbf{R}_{\{1, \dots, n_r\}}^{c_r}.$$

where \mathbf{C} can be regarded as the "terminal" of "concentrator" \mathbf{P} ; \mathbf{S} is the "terminal" of "Concentrator" \mathbf{C} ; and \mathbf{R} is the "terminal" of "Concentrator" \mathbf{S} , and $n_p, n_c, n_s, n_r \in \mathbb{R}^+$, and *subscript* numbers $\{1, \dots, n_p\}, \{1, \dots, n_c\}, \{1, \dots, n_s\}$ and $\{1, \dots, n_r\}$ represent various instances of network components belonging to the particular class out of c_p, c_c, c_s, c_r respectively.

Let $\text{Cost}(\cdot)$ be *total costs*, which are associated with three main costs: (1) the cost via the link (e.g., transmission cost via wireless channel; traffic condition influenced cost due to finite link capacity etc) and (2) the Total Cost of Ownership (TCO). TCO includes tangible base cost of CO (Cost of Ownership) and intangible costs. And (3) *Cost of goal-driven Service-Composition (CSC)*: the activation of a SLA-defined service usually involves many decomposed sub-services to work together. Sub-services that may need to use the services of others are integrated and assembled together. Goal-driven autonomic element-based service activation requires the component-based service to be able to be self-assembled.

$$\text{Cost}(\cdot) \triangleq \begin{cases} \begin{bmatrix} f(\cdot) - CSC + BC + VC \\ \varphi(\omega_{n,i,k}(t), R_{n,i,k}(t), C_{n,i}(t), \lambda_{j,k}) \\ \text{Components logically connected} \end{bmatrix} \\ \begin{bmatrix} \propto \\ \text{Not logically connected} \end{bmatrix} \end{cases} \quad (1)$$

Where $\varphi(\omega_{n,i,k}(t), R_{n,i,k}(t), C_{n,i}(t), \lambda_{j,k}) \in \mathbb{R}^n$ are the link costs relating only to the components in the resource layer. CPU/Memory usage, bandwidth, capacity are all factors required in the calculation of cost. CSC is the cost for service composition.

The cost values are determined in three parts. The first part considers mainly link costs consisting of the parameters in Table 1. The second part is determined by TCO, which is

Table 1. Costs and Determined Parameters

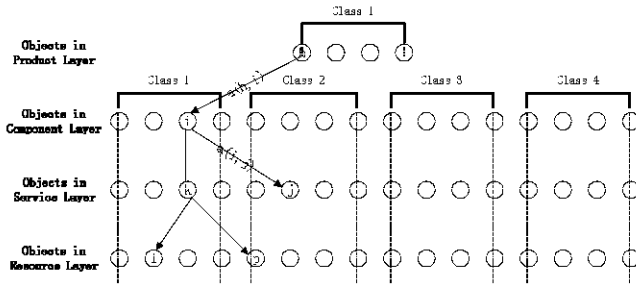
Parameters	Cost
Traffic Intensity Condition \uparrow	\uparrow
Node Capacity Level and Link Capacity Level \downarrow	\uparrow
Delay Time \uparrow	\uparrow

a function of BC and VC as shown in equation 1. The third part is determined by CSC which depends on the integration costs.

3.2 Benchmarking structure and its cost model

This benchmark structure has a strong link with our architecture described in previous publications. Figure 1 depicts the constrained benchmark structure containing the object nodes as the instantiation from classes. Each node in this figure represents one managed element (including managed services and managed physical resources) in the four layer model. The edge weights $a(i, j)$ between them denote the Effective Cost (EC) that the configuration process needed.

We describe it as a *constrained* structure. This *candidate list* as suggested similarly by Dorigo [6] provides possible paths as roughly-known directions for agents. Agents behave randomly within those possible candidate clusters, in so doing, the dimensions of the search space are further reduced as well as the computational time being kept within reasonable limits. The candidate list is determined by the following three preliminary parameters — 1) Dependency String (DependsOn): Denoted as D , is a binary string. 2) Connectivity Binary String: shows connection status between individual objects. 3) Cost of Usage: considers the sum of integrated service costs defined in Equation 1.

**Figure 1. Managed Elements**

The service configuration process needs the information of the *Effective Costs* (EC) instead of cost information. EC is a function of dependency and costs, and is stored into *alocal information centre*. The calculation EC is illustrated in equation 2:

$$EC(i) = D \times Cost(i) \quad (2)$$

How the AEs get the cost values via external environments or by coordination behaviors are not in the scope of this paper. We assume these information is provided in the local information center and is stored into hierarchical XML structure for our calculation purpose.

4 A PBIL implementation of the benchmark

The simulation model evaluates how the Population Based Incremental Learning (PBIL), as a special type of genetic algorithm (GA), can be adaptable to the dynamic environment with its "learning" (via probability vector) and adaptation strategy in order to fulfill our configuration task. PBIL searching strategy has been applied in many fields since initially being proposed by Baluja in the year 1995. In accordance with our architecture, we take email accounts configuration as a testing scenario.

The following shows the email configuration process with regards to our analysis in the previous section. We assume the same number of objects instantiated of each class. That is,

1. In the product layer, there is a class of product Email (User), under this class, there are 4 objects P_0, P_1, P_2, P_3 which are instantiations of a Golden Email Account.
2. In the component layer, there are assumed to be four classes of components. Some components could be:
 - a) C_0 - Basic_Email_Box(users); b) C_1 - Dial-up Internet; c) C_2 - Premium_Email_Box(User); and d) C_3 - Broadband Connection.
- Each one of the classes contain 4 objects and are denoted as $C_{00}, C_{01}, C_{02}, C_{03}$ and so do service and resource objects.
- In the service layer, there are assumed to be 4 classes of services. Some services could be -- a) S_0 - Transport (its objects are, for example, POP/IMAP, TCP/IP, SMTP, DNS); b) S_1 - Authentication (e.g., SpamFiltering et al.); c) S_2 - Anti-Virus (e.g., VirusFiltering); d) S_3 - Billing Service.
- In the resource layer, there are assumed to be 4 classes of resource. Some resources could be -- a) R_0 - Router; b) R_1 - Switch; c) R_2 - Backoffice Storage Servers; d) R_3 - Bandwidth.

To simplify the computational complexity in the simulation, we assume each class has only 4 instantiated objects with regards to different users' SLA. Therefore, the total number of objects is 52. The data used to calculate *effective cost* are derived from our university campus network based on monthly throughput. Figure 2 describes the algorithmic steps towards minimum cost calculation. The detailed illustration can be found in the following pseudocode presentation regarding 1) how to update probability vector, and 2) how to get the minimum cost value, etc.

5 Simulation Results

The paths discovered by the centralized PBIL algorithm formulate a best configuration solution on the basis of *cost* criteria described by the objective function. The nodes along this configuration path represent the components to

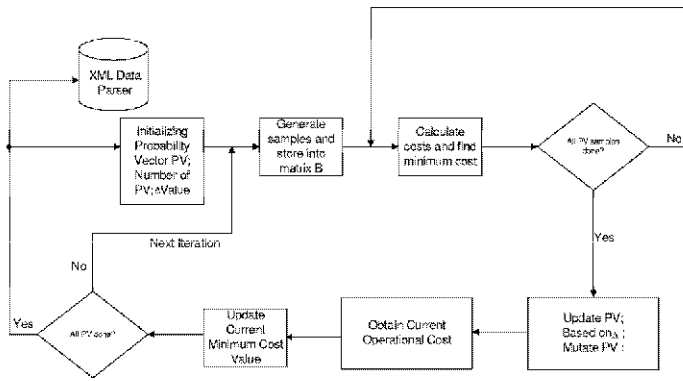


Figure 2. PBIL algorithm for network optimization

be necessarily included. A Java-based PBIL application for this configuration process is designed, and the simulation GUI is constructed. The path discovered by the PBIL algorithm is encoded in the probability vector in Figure 3.

Our particular configuration problem requires (1) a 36-pair of binary string (=72 bits) to describe the edges between 52 nodes; (2) n (e.g., 100) trial sample vectors which are generated according to the Probability Vector (PV). After each generation, the PV will be adjusted incrementally to the effect that the best solution sets are to be enhanced and the bad solutions are to be diminished; (3) 500 loops of iterations which corresponding to the the number of generations. (Actually, 500 is larger than we require. Generally, 100 will suffice.) We noted that the discovered path strongly depends on the cost values.

Figure 3 shows the performance test on PBIL adaptation strategy with regards to achieving minimum cost in terms of instantiating a service or a product. Around 100 iterations are sufficient to find a configuration path in a converged telecommunication network. The binary string of final probability vector indicates the subscript of network components in need of being involved into this configuration process given to the known system objectives.

6 Conclusion

The purpose of this paper is to describe a notional management structure that would lend itself to the selective introduction of autonomic behavior into those parts of the OSS where it would be appropriate. The validation of this architecture is done via one stochastic searching-based genetic algorithm - PBIL, which has been applied to service configuration issues by incorporating this notional management structure.

The main benefit of the model is that it clearly indicates:

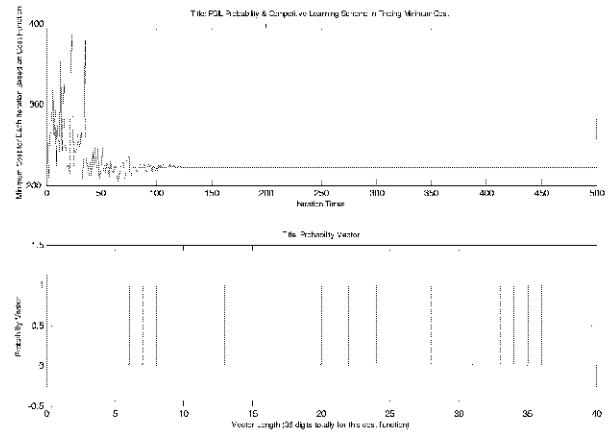


Figure 3. Performance of PBIL adaptation strategy applied into minimum cost evaluation

- 1) How to position autonomic behavior and how to set it in the context with the OSS systems; 2) How it might be simulated and how it might be implemented in real applications

The simulation results show that this proposed architecture and benchmark issues can be well fitted into the autonomic communication networks in an ever-changing complex network environment as long as the eligible self-learning and adaptation strategy or corresponding algorithms are carefully designed and implemented. Although PBIL is essentially a centralized scheme, good performance is still achieved for the given configuration problem.

References

- [1] TMF, "The ngoss technology neutral architecture specification v3.0," tech. rep., Tech. Rep. TMF053, TMF, 2003.
- [2] F. Chiang, R. Braun, and J. Hughes, "A nature-inspired multi-agent architecture for self-configuration of network services in pervasive communications," *in print for publication with Journal of Pervasive Computing and Communications*, 2006.
- [3] F. Chiang, "Self-configuration of network services with nature-inspired learning and adaptation," *Accepted for publication by Journal of Network and Systems Management*, 2006.
- [4] F. Chiang, R. Braun, S. Magrath, and S. Markovits, "Autonomic service configuration in telecommunication mass with extended role-based gaia and jadex," vol. 2 of *Proceeding of 2005 IEEE International Conference on Service Systems and Service Management (ICSSSM2005)*, pp. 1319–1324, 2005.
- [5] S. Magrath, R. Braun, F. Chiang, S. Markovits, and F. Cuervo, "Autonomic telecommunications service activation," in *Workshop on Autonomic Communication for Evolvable Next Generation Networks, 7th International Symposium on Autonomous Decentralized Systems*, pp. 731–736, April 4 - 8, 2005.
- [6] M. Dorigo and T. Stützle, *Ant Colony Optimization*. MIT Press, 2004.