A Nature Inspired Multi-Agent Framework for Autonomic Service Management in Pervasive Computing Environments

Frank Chiang and Robin Braun
Faculty of Engineering, University of Technology Sydney,
Broadway, NSW 2007, Australia

***Abstract*— This paper describes the design of a scalable biomimetic framework in the management domain of complex *Ubiquitous Service-Oriented Networks*. An autonomous network service management platform - SwarmingNet is proposed. In this SwarmingNet architecture, the required network service processes are implemented by a group of highly diverse and autonomic objects. These objects are called TeleService Solons as elements of TeleService Holons, analogue to individual insects as particles of the whole colony. A group of TSSs have the capabilities of fulfilling the complex tasks relating to service discovery and service activation. We simulate a service configuration process for Multimedia Messaging Service, and a performance comparison is made between the bio-agents scheme and normal multi-agents scheme.**

I. Introduction

The operational management of Next Generation Network (NGN) services is expected to be autonomous, scalable, interoperable and adaptable to the diverse, large-scale, highly distributed, and dynamically ever-changing network environment in the future. The functional management for them are also desired to be as simple as possible from the perspective of both designs and implementations. Current network management infrastructure is struggling to cope with these challenges.

In contrast, social insects and biological organisms have developed relatively easy and efficient mechanisms to thrive in hostile, dynamic and uncertain environments after many years' evolution and natural selection. Hence, taking advantages of the synthesis on full-scaled biological societies is of vital importance in achieving autonomic management in the future Ubiquitous Service-Oriented Network (USON) [2], which will dynamically connect human beings and home/office electronic appliances via distributed devices (e.g., cell phones, notebooks, PDAs) and applications running on these devices to flexibly enable services at any time, any places without *constraints in quantity or frequency* [3].

We believe a better designed bio-functional framework exhibiting self-organization, adaptation, scalability and mobility is necessary for the availability of seamless services exempted from potential failures and attacks. This research is motivated by the observations from swarm intelligence in biological systems which are based on the principles underlying the behaviors of those system (e.g. Termite, Ant colonies, Bees, Fish schooling [4]; See Figure 1 for details) consisting of swarms of many agents. Those biological systems have developed instinctive mechanisms to achieve the desirable characteristics for network management paradigms such as autonomy, robustness, scalability, adaptability and even simple individual actions for design and implementation.

Instances of Colonies	Swarming Behaviors
Ants	Items sorting in nests; Food foraging; Path selection
Bees/Ants/Termite	Hive construction or nest building
Fish	Schooling
Birds	Flocking
Spiders	Web weaving
Bacteria/Slime mold	Aggregation
FireFlies	Synchronization blinks
Locusts	Group hunting

Fig. 1. Swarming Behaviors in the Nature World

The aim of this paper is to propose a bio-swarming framework — *SwarmingNet* for network service managements. The biological platforms proposed by Suda [6] and Suzuki [7] are more emphasizing *evolutionary* behaviors of agents. The status of network services (mutation, clone, reproduction and replication) depends on multi-agents' internal states. These platforms are particularly complicated in terms of rapid practical application. We distinguish our framework from theirs by applying TeleSolon hierarchy[1] concepts into hierarchical network management system which is analogue to the *ecosystem* in nature and *colony* structure in ants. We apply stigmergic ant-foraging behaviors into the threshold-based self-organization algorithm. This is much easier to be applied into practical embedded systems with industries.

This paper is organized as follows: Section II oulines the current development of bio-swarming intelligence. The design principles and our self-organized provisioning algorithm for *event-based* autonomic management architecture are mainly specified in section III. The system-level architecture are

[1]Due to the limitation of pages, details of THSs and TSSs can be referred to in the full version under requests

included in section III which section also covers the definitions and theorem for the threshold-based self-organized algorithm. Service configuration process of MMS is simulated as an illustrative application, simulation results are shown in Section IV. Finally we conclude with performance comparisons and future work trends in section V.

II. Related Work

The relative research on *bio-swarming intelligence* started around 30 years ago in the field of cellular robotic systems, Wolfram [8] outlines swarming intelligence as a new emerging science comparing with dynamic systems and chaotic systems where most of the work are on dynamics and complex interactions. The researches on large-scale biological systems and the simple entities consisting of such systems in the filed of Artificial Life (AL) and Complex Adaptive Systems (CASs) help solve the various engineering problems, control problems and computational problems in the way of analoging or adapting from biology. Outstanding achievements even include a more complex swarm explorations in NASA's intelligent rovers and spacecraft which mimick the societal behaviors of swarms, flocks by offering great potential implementation power in fulfilling NASA space mission in the future. Swarm intelligence offers an alternative way of designing "intelligent" systems in which autonomy [9],[10], emergence, and distributiveness replace control, preprogramming, and centralization [11]. Our research is motivated by previous research work on the ant-based algorithms, models, theories are mainly carried out by Di caro [12], Agassounon and Gambardella [13].

III. Design Principles

Manual operation of the network configuration in decentralized environment becomes more and more unrealistic, meanwhile, the high dynamics in UOSN networks, variations of network conditions (such as bandwidth, traffic congestion) and users' requests require services configuration process to be autonomically functioning. In addition, autonomicity provides great chances for the human system administrators and technicians to focus on higher technical issues without intervention into tasks from huge amount of operations for service configuration, provisioning and assurance. Hence, there is a need to introducing intelligent ability into devices (programmable) and new policies/ new rules (autonomically updated) to achieve desirable performance. The incorporation of social insects paradigm into autonomic service configuration is believed to be the right way to meet these requirements. This can be achieved by modelling networks as a distributed aggregation of self-organized autonomous TSS[2] solons in our approach. This is similar to the social insects colony (networks) consisting of large amount of individual insect (solon).

A. Self-Organized Service Provisioning & Algorithm

Future networks should not only provide basic connectivity but also intelligently and immediately enable on-demand services in *pervasive computing* environments at anywhere, anytime. Those services must be provisioned in a flexible and distributed way in highly dynamic runtime infrastructure. Thus, service deployment and management for devices in USON are extremely difficult since a provisioning infrastructure ought to cope with the high level of heterogeneity, degree of mobility, and care about limited device resources.

In this context of self-organization, we describe the service provisioning as the ability to *create, remove, reproduce, reconfigure* the instances of services at runtime. Moreover, the bio-mimetic agents run at particular network nodes, (1) *measure* the local demands for network services autonomically beyond other nodes; (2) *reconfigure/reproduce* local services when demands are detected; (3) *remove* the services when there are no demands of quantity.

a) Self-organization Algorithm —: This section defines the algorithm to enable self-organized bio-networks. This algorithm also considers the scalable design principle, such as adaptability and robustness in distributed environment; localized decision making process based on neighborhood information. Hence, it is also practical principle to consider only *local customers* who are not too far from the service center. The *content-based event messaging* system categorize and store the messages for different services into different space in information store (e.g., database). The following definitions will together define the problem domain in our service provisioning process.

Definition 1: A dynamic threshold θ is configured for each requested service respectively; A parameter η is used to evaluate the keen degree for customers who require certain services; A parameter d represents Euclidean distance between available server of services and customers; The localized service zone ω is designated as $\omega(d) \leq 10\% \cdot D$, where D is the diameter value of the whole service area

Definition 2: A parameter τ evaluates the intensity of digital pheromone which are placed along the traces to servers in dispersed area by previous bio-agents. The intensity of digital pheromone measures how easy the service is available in the particular server the path gets to. Moreover, τ is related to the Euclidean distance d

Lemma 1: (Time) When customer requests are accumulated to the θ value, the service provisioning starts. This is a dynamic value varying in accordance with essence of specific services. For some reason, some services should be activated as soon as there is a demand for it. While some services can be activated only when there are enough number of requests

Lemma 2: (Space) When the customer and services are both inside localized service zone, the service could be provisioned

Theorem 1: The self-organized process for service provisioning is activated successfully iff Lemma 1 and Lemma 2 are satisfied simultaneously[3]

[2]Due to the limitation of pages, the details of THSs and TSSs can be referred to the full version under requests

[3]Due to the limitation of page lengths, the proof part can be referred to under requests. And the full pseudocodes on our simulation for ant-inspired learning and adapation can be provided under any request.

B. System-Level Architecture

This architecture we propose here is partly depicted in Figure 1 of our paper [4]. We have to omit it due to page limitations. This systems-level architecture illustrates the combinatorial links among our three indiscerptible parts for autonomic service activation process: 1) Users; 2) Instrumentation support and measurement, monitoring; 3) Enhanced 4-layers TMF management model. Analogue to the biological society, we introduce the concept of *ecosystem* into the whole system which acts as the environment where agents create, live and die. We designate the *energy exchange* is the "currency" between ecosystem components (e.g. swarm agents) and eco-environment. This is a layered structures with 4 layers of the TMF model with 2 extra management layer for the achieving of autonomic service assurances.

IV. Simulation and Experimental Measurement

In this section, we present our numerical experiment results to evaluate the performance of the operation and efficiency of our proposed architecture by applying biological pheromone. We choose the Multimedia Messaging Service (MMS) services as an evaluation application. As for the system-level architecture, the *managed object* in this context is MM_Box (MultimediaMessaging_Box); The product components are 1) Gold_MM_Box (capacity=1000MB), 2) Silver_MM_Box (capacity=100MB), 3) Bronze_MM_Box (capacity=10MB).

The events messages contain on-demanding service provisioning requests from clients, these messages include information on: 1) *create/delete* users' MM account in the product or 2) *migration* Multimedia Messaging (MM) account among the products – *Gold, Silver and Bronze boxes*. The service provisioning results indicate bio-inspired network management paradigm and can maintain SLA compliance as well as efficient transaction time.

Digital pheromone evaluates the degree of difficulty in activating or migrating MM accounts which are stored in MM servers in this context (The large intensity of digital pheromone means MM_boxes are easier to be migrated from silver to gold, or from bronze to silver, etc.). The effectiveness of digital pheromone in MMS server configuration process in the framework has been tested. The service-configuration performance comparison between the bio-agents and normal agents are analyzed. Java classes are built on the modelling platform AnyLogic® . Specifically through the hierarchical decomposition structure of the object classes, the system scalability is easily achieved in this swarm-based object-oriented simulation paradigm.

The experiment scenario is summarized here: The event messages with service requests from clients PC in our testbed will trigger the service configuration process whenever the service requests approach a service threshold θ_{ij} where i represents the service ID, j represents the clients' ID.

We argue this is an *autonomic* process instead of an *automatic* paradigm because the θ_{ij} value changes according to the requested service profiles, autonomic agents learn and decide the best threshold. Our adaptation strategies are not fixed to be a set of rules like that in automatic system, on the contrary, the autonomy is achieved by goal setting and suggestion through learning and modification of the existing adaptation strategy.

The multiplication ω of digital pheromone intensity and customer keen index will be considered as an important index in an exponential formula (e.g., $exp(\omega)$), which determines which MM account will be configured to activate in certain MM_servers. Service lifetime is calculated by multiplication of these two factors. Moreover, network vendor agents will view into caché database for updated information being synchronized with our 4-layer structure which cover the *specification* files for products and services, and all the configuration files for resources (e.g., devices, equipment, etc.). Specifically, by taking into fact that service requests usually are provisioned by local servers, the factor d, an Euclidean distance measures our virtual distance between service requests to MM_server in the coordinate plane. If there are servers meeting the requirement simultaneously, we will randomly pick up one of them.

Figure 2 describes the overall simulation configurations in details. Figure 3 shows the parameters of 3 experimental scenarios which test the service provisioning. As shown in Figure 3 for Experiment 3, the number of MM_servers is decreased till 100 while other parameters remain the same as above.

Topology	❑ 1000 virtual MM servers are uniformly distributed into an area [0, 280]
Event Parameters	❑ Requests for on-demanding service are randomly generated by clients with fixed seed=1 over a time interval of 20 days ❑ Services lifetime are not permanent, they will deceased whenever there are no needs or termination willing from customers. {We give maximum_ service_lifetime = 1 day}
Space/Time Dynamics	❑ Transaction time for each service requests dynamically change according to A = Customer_Keen_Index and B = Digital_Pheronmone_Intensity; ❑ In order to simplify the simulation factors, we set $A=0.5$; $B=\gamma\times d$, (γ=0.5 or 0.6) where $d=\sqrt{(x-x_1)^2+(y-y_1)^2}$, d represents the Euclidean Distance between one particular service request and one particular MM server; ❑ Maximum duration for any certain service provisioning $\lambda=0.4$ days ❑ Services configuration happen on those servers which are close to customers like in real world. We only provision the service distance $d\leq30$

Fig. 2. Experiment Description

	Num_of_MM_Servers	Initial_Value_Provisioned_Servers	Digital_Pheromone_Intensity	Customer_Service_Keen_Index
Test 1	1000	0	0	0.5
Test 2	1000	0	0.6	0.5
Test 3	100	0	0.6	0.5

Fig. 3. Parameters for Experimental Test 1, 2 and 3

A. Experimental Results

Based on the three experimental test scenarios described in the previous subsection, the performance comparison between bio-agents on biologically inspired framework and normal agents without biological behaviors are presented in Figure 4. As we can see, the number of servers in service provisioning process is bigger for bio-inspired agent framework when the same quantity of service requests (=11692) arrives. Service configuration tasks, or workload are distributed uniformly into servers with a shorter response time, and performance for load balancing is also optimized by this bio-inspired framework.

Fig. 4. Performance Comparison for Number of Servers Configured between Biological Ant-based Agents Scheme and Normal Agents Scheme

For both bio-agent scheme and normal agent scheme, we calculate the service configured percentage for the total number of services which are finally provisioned respectively in the low client node density (=100 in the fixed area) and high client node density (=1000 in the fixed area). In low client requests environment, service configured percentage is 72%, higher than high client requests environment. However, our biological agents scheme results to better service configured percentages in the same environment as those normal agent schemes. Details are illustrated in Figure 5.

V. Conclusion and Future Work

According to our simulation results, firstly, we conclude that our bio-inspired multi-agent framework provides a solution to envision future *antonomic* service management system. This framework outperforms the current normal multi-agent based system in terms of service discovery and service assurance for future IP networks. In addition, our architecture can work flexibly as an universal framework in multi-agent based autonomic communications. Secondly, this framework does not rely on particular types of insects societies or colonies, i.e. agents could be entities in USON ranging from any hardware devices to robotic agents, or biologically-inspired software elements. Our future work will specifically focus on the performance comparison among Particle Swarm Optimization

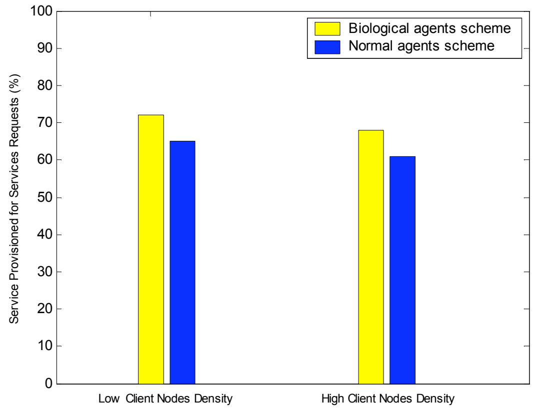

Fig. 5. Percentage of Configured Services in Heterogeneous and Dynamic Network Environment with Different Clients Density

(PSO), Ant Colony Optimization (ACO), Genetic Algorithm (GA) or Probability Based Incremental Learning (PBIL) with regards to efficient service configuration issues on the basis of this framework.

References

[1] M. Takemoto, T. Oh-ishi, T. Iwata, Y. Yamato, Y. Tanaka, K. Shinno, S. Tokumoto, and N. Shimamoto, "A service-composition and service-emergence framework for ubiquitous-computing environments," in *International Symposium on Applications and the Internet Workshops, 2004*, pp. 313–318, 2004.

[2] K. Yamazaki, "Research directions for ubiquitous services," in *Proceedings of International Symposium on Applications and the Internet, 2004.*, p. 12, 2004.

[3] K. Suzuki, "Ubiquitous services and networking: monitoring the real world," in *Proceedings of International Symposium on Applications and the Internet, 2004*, p. 11, 2004.

[4] F. Chiang, R. Braun, S. Magrath, and S. Markovits, "Autonomic service configuration in telecommunication mass with extended role-based gaia and jadex," in *Proceeding of IEEE International Conference on Service Systems and Service Management,* vol. 2, pp. 1319–1324, 2005.

[5] T. Ono, K. Ragab, N. Kaji, and K. Mori, "Service oriented communication technology for achieving assurance," in *Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops*, pp. 69–74, 2002.

[6] T. Nakano and T. Suda, "Self-organizing network services with evolutionary adaptation," *IEEE Transactions on Neural Networks,*, vol. 16, no. 5, pp. 1269–1278, 2005.

[7] N. Makinae, H. Suzuki, and S. Minamoto, "Communication platform for service operations systems in advanced intelligent network," in *1997 IEEE International Conference on Communications, 1997*, vol. 2, pp. 872–877, 1997.

[8] S. Wolfram, *A new kind of Science*. 2002.

[9] IBM, "The redbook of autonomic computing," tech. rep., April 2003.

[10] IBM, "An architectural blueprint for autonomic computing," tech. rep., April 2003.

[11] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. Santa Fe Institute Studies in the Sciences of Complexity, Oxford University Press, 1999.

[12] G. Di Caro, F. Ducatelle, and L. Gambardella, "Swarm intelligence for routing in mobile ad hoc networks," in *Proceedings of Swarm Intelligence Symposium*, pp. 76–83, 2005.

[13] R. Montemanni and L. Gambardella, "Swarm approach for a connectivity problem in wireless networks," in *Proceedings of IEEE Swarm Intelligence Symposium*, pp. 265–272, 2005.