Future Generation Computer Systems 28 (2012) 533–543

Contents lists available at SciVerse ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Morphological approach for autonomous and adaptive systems based on self-reconfigurable modular agents

Kiwon Yeom [b,a,*], Ji-Hyung Park [a]

[a] Center for Intelligence and Interaction, Korea Institute of Science and Technology, Seoul, Republic of Korea

[b] Human Systems Integration Division, NASA Ames Research Center, Moffett Field, CA, USA

ARTICLE INFO

Article history:
Received 6 May 2010
Received in revised form
25 February 2011
Accepted 1 March 2011
Available online 21 April 2011

Keywords:
Morphogenesis
Self-organization
Modular agents
Federation of agents

ABSTRACT

This paper describes a novel approach for managing self-organizing, distributed modular components in dynamically changing environments. The main concept is to fabricate a system which is composed of dynamically associated modular agents, that can migrate and reorganize by itself while the system is being executed. Association between modular agents can be varied and transmuted according to components' own migration schemes including deployment based on biological processes. This paper presents a self-organizable architecture, which can reorganize and reconfigure a system based on modular agents. It is contrived through observation of biological phenomena, and implements a platform to host the architecture in dynamically changing environments. We draw several key features of the modular agents, describe the principles of the modular agent based self-organizable framework, and depict how the proposed framework satisfies the functional requirements of network applications, which are made of several agents. We also demonstrate the efficiency and scalability of the framework through examining some simulation results.

1. Introduction

In the near future, it is easy to imagine that many people, regularly or irregularly, will connect to and disconnect from running applications via global networks as a part of their daily lives. At the least, future applications will need several properties. First, they will need an ability to adapt to dynamically changing user requests and network conditions. Second, they have to keep and handle tremendous demands from users. Finally, applications will have to recover, survive, and maintain their availability against partial failures [1–3].

Distributed computing systems often consist of a number of software components, which run on different computers and cooperate with each other via network on demand [4]. The complexity of distributed systems has already exceeded human's ability to manage components, which are deployed at several computers on networks, by traditional methodologies such as centralized or top-down techniques [5,1,6].

Over billions of years, large scale biological systems, such as ants and bees, have developed adaptation mechanisms to live and survive in changing environments. They are able to enlarge their scale without any centralized control because they are able to behave autonomously, and they are only influenced by regional conditions and individual interactions with their neighbors. They also adapt themselves to dynamic conditions [7,8]. These large scale biological systems can survive partial cave-ins of the nest or disasters resulting from predators because they do not depend on any master, not even the queen [9]. In terms of engineering, the suitable and desirable traits of these biological systems are adaptability, scalability, and survivability, which are not available to an individual. In the future, networked applications will need to be more autonomous, adaptive, scalable, and survivable to dynamically changing environments such as changing network traffic or available resources relative to the number of users [10]. It is difficult to cope with the changes in user requests, or to reflect the changes of execution environments during runtime such as adding and removing components, changing network topology, etc. These kinds of problems will be more critical in both ubiquitous computing environments and large scale distributed systems like grid computing environments [11]. Since their computational resources, such as CPU, memory, and I/O devices, are heterogeneous and restricted, an individual system is able to only support its own applications [12]. The proposed architecture is motivated by the observation of desirable properties in the future network applications, previously described, and by inspirations from biological systems [13–15].

In this paper, we describe a scalable system framework that allows for developing adaptive, autonomous, highly distributed,

* Corresponding address: NASA Ames Research Center, Mail Stop 262-4, Bldg. 262, Rm. 155, P.O. Box 1, Moffett Field, CA 94035-0001, USA.

E-mail addresses: kiwon.yeom@nasa.gov, pragman@gmail.com (K. Yeom).

0167-739X/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2011.03.002

and mobile agent based network applications. This paper presents a set of key features for the adaptive and autonomous component based agents in our architecture, and introduces functional requirements for building an application which is made up of a federation of agents. This paper presents how the agents can combine and reorganize a confederation of components, which may execute on heterogeneous computing environments, and modify their associated contexts with the user requirements (e.g., locations, tasks, etc.). This architecture also provides agents with seamless runtime services without any halting of the system. We describe the design principles of the framework, and demonstrate how the proposed framework gratifies the functional requirements of future networked applications. To this end, we present some experimental results.

2. Related works

In this section, we describe several biologically inspired approaches to distributed and multi-agent systems. Our platform is similar to existing mobile agent platforms, such as Aglets, Voyager, JumpingBeans, and Grasshopper [16–18,9], in the sense that it implements a weak migration mechanism for agents. In contrast, the proposed framework emphasizes the decentralized combination and reorganization of agents on distributed network environments based on the biological concepts.

Most existing agent platforms presuppose the existence of a host controller or of centralized agents. For instance, Hive deals with decentralized agents [16], but its current implementation is based on a central host and deployment of the remote message by Java RMI. In contrast, the proposed framework allows agents to organize a virtual network among agents using their functionalities and relationships. They also carry out distributed discoveries through interactions with each other, which result in social networking.

There are several platforms which provide social networking functionality such as OceanStore, Pole, and Co-Field [18,9,19]. The discovery mechanism for them is based on structured peer-to-peer communication with a distributed hash function on the whole overlay network. Such discovery mechanisms are very expensive, and it is hard to maintain their structure in dynamic network environments, where individuals frequently join and leave.

Alternatively, instead of depending on any distributed hash function, our framework is designed on a loosely coupled virtual network of agents in order to adapt to dynamically changing environments. It also provides a flexible discovery scheme that allows agents to detect and specify the corresponding cytokine/chemokine (pheromone), which affects the behavior of other agents of the same type (e.g. attracting them sexually).

In a few attempts, the architecture for real distributed systems have been presented. For example, the Anthill project established a bio-inspired middleware for peer-to-peer systems, which consists of an aggregation of interconnected dens [20]. Autonomous agents, called ants, can move around the network to satisfy user requests. The main difference between Anthill, including its applications, and our proposed framework is that agents are independent objects that can accomplish their tasks and are flexible components that can be combined with and segregated from each other through the proposed framework. The Co-Field project suggested the concept of general coordinator for the movements of agents and organization of a group of agents, which consist of mobile devices and mobile robots. However, it is only available within limited simulation environments.

Murata and Dressler suggested biologically inspired networking for communications between sensor nodes under mobile wireless sensor network environments. Their research has been focused on data transmission mechanism which can consume low power source [21,22]. However, they are limited to only conceptual proposal and sensor networks (Table 1).

Researchers have also explored the behavior of other biological systems such as cells and social insects as inspiration for emergent computing architectures for ubiquitous and pervasive computing.

One of the pioneering efforts was by [23]. It is called AntNet which has probably been the most influential prototype based on ACO (Ant Colony Optimization). AntNet introduced the concept of forward and backward ants. Like ABC (Ant Based Control), forward ants stochastically follow the pheromone from a source to a destination [24]. However, it requires a very long delay to propagate information, since routing tables are only able to be updated by backward ants. [25] also proposed a limited extension for networked machines. Against a new virus or malicious code as any machine learned how to defend itself the machine would tell all of its neighbors how to do so as well. However, in this proposal when a neighbor would broadcast corresponding information to other neighbors, it found that it was already infected with the same virus. In the paper, the authors mainly concentrated on the aspect of code reusability that was already in use in IBM's AntiVirus product [26].

In order to develop effective applications for ubiquitous or pervasive like computing environments, new abstractions for the application architecture are required [27,28]. Whether due to the increasing complexity of infrastructure based services and scalability issues, or the highly dynamic and unstructured nature of mobile networks, these computing paradigms demanded new functionality which applications or services are able to be self-organized and self-configured by themselves.

One of the research projects pursuing self-organization is performed by [29]. In this work, the authors introduced an alternative architecture for Internet based service applications based on the bio-inspired cyber-entities which can be managed by their host framework using life-cycle. Cyber-entities can replicate, reproduce, migrate across the network topology, and die if they do not have enough energy. The authors evaluated this bio-inspired architecture via empirical simulation and showed promising result, however, the results were slightly mixed and also seemed to suggest that the proposed architecture was somewhat sensitive to initial experimental conditions.

Their group expanded their initial achievements toward more distributed ways and showed their evaluation in [30]. More recent work by [31] showed cyber-entities the ability to mate, both sexually and asexually, and therefore evolve. Although they showed that evolution could help mitigate the sensitivity of the system to its initial conditions, they did not evaluate their model's performance in comparison to other, more traditional, distributed Internet service models.

The BIONETS research project, which has been accomplished by [32] and focusing on pervasive computing rather than Internet services, has similar goals and inspirations to the Bio-Networking Architecture project [33]. They proposed a two-tiered architecture: the first tier consists of user nodes, called as U-node, with abundant storage capacity and computing power, whereas the second tier is composed of T-nodes, very simple sensor nodes that cannot interact with each other, but U-nodes. No traditional routing protocol is used by either type of node and they do not even have a notion of address.

Recently, [34] have described a somewhat different approach to self-organizing services for mobile networks. The authors present a simulation model based on their design guidelines. The authors also provided little amount evaluation and showed that their guideline does indeed produce the desired result for the scenario considered.

In our framework, there are two key ideas. The first is to implement components as mobile agents that can travel between