Chapter 10
Role-Based Autonomic Systems

Haibin Zhu
Nipissing University, Canada

ABSTRACT

Autonomic Computing is an emerging computing paradigm used to create computer systems capable of self-management in order to overcome the rapidly growing complexity of computing systems management. To possess self- properties, there must be mechanisms to support self-awareness, that is an autonomic system should be able to perceive the abnormality of its components. After abnormality is checked, processes of self-healing, self-configuration, self-optimization, and self-protection must be completed to guarantee the system works correctly and continuously. In role-based collaboration (RBC), roles are the major media for interaction, coordination, and collaboration. A role can be used to check if a player behaves well or not. This paper investigates the possibility of using roles and their related mechanisms to diagnose the behavior of agents, and facilitate self-* properties of a system.*

INTRODUCTION

Cognitive Informatics (CI) is an interdisciplinary study into the internal information processing mechanisms and processes of the Natural Intelligence (NI) (Wang, 2007a). Autonomic computing is one of the fundamental technologies of for intelligent agents, which are a simulation of the natural intelligence possessed by human brains with computers (Wang, 2009).

DOI: 10.4018/978-1-4666-0264-9.ch010

Traditional computer systems management is often centralized and hierarchical. Current computing systems are physically scattered in different places, and functionally distributed with increasingly complex connections and interactions. This situation makes centralized management schemes much more difficult and even infeasible (Samimi et al., 2006; Tesauro et al., 2004). The autonomic computing paradigm is proposed to deal with the increasing complexity in computer-based systems, and the shortage of skilled human manpower to advance the related technological developments and innovations (Wang, 2007b). Compared with traditional systems, autonomic and autonomous management is required to guarantee that the overall computing system operates properly. Autonomic systems should be capable of managing themselves. These properties make autonomic computing more advanced and challenging.

Autonomic computing is to simulate the natural neural system of human bodies to possess autonomic properties, i.e., self-* properties (Horn 2001; Kephart, 2005; Muller et al., 2006): **self-awareness:** an autonomic system should "know itself" and be aware of its state and behaviors; **self-configuring:** an autonomic application should be able to configure and reconfigure itself under varying and unpredictable conditions; **self-optimizing:** an autonomic system should be able to detect suboptimal behaviors and optimize itself to improve its execution; **self-healing:** an autonomic system should be able to detect and recover from potential problems and continue to function smoothly; and **self-protection:** an autonomic system should be capable of detecting and protecting its resources from both internal and external attack and maintaining overall system security and integrity.

Three basic ways are used to make a system autonomic (Lapouchnian, 2006): building an isomorphic space of possible system configurations; designing planned capabilities and social skills to delegate tasks to components; and evolutionary approaches such as making components as biological cells. Role-based approaches belong to the second category.

In fact, a human body is composed of numerous cells and cell groups. All the cells and cell groups collaborate and accomplish different tasks to provide the functions of their host - the human body (Wang, 2003). Because a human body is in a micro-world, it is easier to understand the autonomic properties of a system with a human society in a macro-world.

A person is social. Without a society, a person never shows and requires his or her sociability. Without the requirement of sociability, one does not need to present their autonomy and autonomicity, because they coexist with the sociability. In a society, people work, interact, and collaborate with each other directly and roles are abstract regulations for them to collaborate and coordinate. In such situations, roles are abstract and implicit tools to evaluate people's performance in completing tasks. This kind of evaluation is ambiguous and qualitative most of the time. Therefore, it is difficult to operate role-based evaluations in the social world. However, roles can be made into easily managed mechanisms to evaluate agents' performances in computer-based systems (Zhu, 2007, 2008; Zhu & Zhu, 2003, 2006, 2008a, 2008b, 2008c).

In a multi-agent system, there are two kinds of "selves": one is the system itself and the other is an agent itself. To efficiently present the autonomic properties of the system self, we must fully develop the autonomic properties of the agent self. To present the system self, there must be task distributions, coordination and collaboration among agents. To make agents autonomic, they must be made be aware of their environments, respond to these environments, and be self-regulating.

The self-management of a society comes from a good central government and the active participation of their members. The self-management of a networked computer system also requires a central management and the participations of their component computers. From the viewpoint of multi-agent systems, an autonomic system is composed of a central manager and many autonomous and autonomic agents.

Role-Based Collaboration (RBC) is a computational thinking methodology that mainly uses roles as underlying mechanisms to facilitate abstraction, classification, separation of concern, dynamics, interactions and collaboration. Based on roles, RBC is such an emerging methodology designed to facilitate an organizational structure, provide orderly system behavior, and consolidate system security for both human and non-human entities