A framework for Internet service evolution based on active object

HU Hua†1, ZHANG Yang2

(1) Institute of Computer & Information Engineering, Zhejiang Gongshang University, Hangzhou 310035, China
(2) School of Computer Science & Engineering, South China University of Technology, Guangzhou 510640, China

†E-mail: huhua@mail.zjgsu.edu.cn

Received Jan. 19, 2006; revision accepted June 16, 2006

Abstract: The wide use of Internet Service in distributed computing and e-business has made the evolution of Internet Service to be one of the most prevalent research fields in software development domain. Traditional methods for software development cannot adapt to the challenge of Internet Service oriented software development. In this paper, we propose a new paradigm for the evolution of Internet Service with active objects from the characteristics of Internet Service and principles of active objects. The paradigm uses an automatic monitoring mechanism of active object to detect and process evolution requirement in system based on Internet Service.

Key words: Framework, Internet Service, Evolution, Active object

INTRODUCTION

The rapid development of Internet and E-business technology has obviously stimulated the wide use of online Internet Services whose increasing number has stimulated much research effort toward online Internet Services software development, and also attracted much attention on the evolution of Internet Service oriented systems. Traditional software developing paradigms, such as waterfall paradigm or prototype paradigm, have problems with adapting to the challenge of online software evolution. This generates a new requirement to develop new technologies for evolution of online Internet Service. Gadke and Graef (2000) proposed a Web Engineering paradigm for software development, where a model of Web composition is used to fulfill the development and evolution of Web software. Ingham et al. (1997) used the Web Engineering paradigm to develop Web Services. Chen et al. (2002) and Kicman and Fox (2004) proposed a feedback mechanism to help the evolution of Internet Service. Kirda et al. (2001) used engineering management and distributed methods to support application of Web services. The main focus of Web Engineering was the development of Web application based on software components, which made Web Engineering research stay in the fields of software component reusing and rapid composition of Web application. Chen’s feedback mechanism depended mainly on human beings and lacked a standard and automatic mechanism. While the current Internet Service software researches cannot adapt to the requirement from the huge number of Internet Services, this paper, based on the characteristics of Internet Services and principles of active objects, proposes a new paradigm for the evolution of Internet Service with active objects. This paradigm uses an automatic monitoring mechanism for active object to detect and process evolution requirements in an Internet Service system.

This paper is organized as follows. Section 2 introduces the basic theory of active objects. In Section 3, a new Internet Service Evolution (ISE) paradigm is proposed. Section 4 details the key
INTERNET SERVICE ORIENTED ACTIVE OBJECT

Event Condition Action (ECA) is a general rule model for researchers of active database to describe behaviors of active databases. In ECA model, “E” represents events that happen in the environment of the active database; “C” stands for the conditions for the active database to respond to the occurring event; “A” represents active database actions triggered by the events when conditions are true. By separating and abstracting the strategy and policy of a system from system data to ECA rules, an active database based on ECA model can realize the separation of management knowledge from normal application data. However, when we try to incorporate ECA into ISE, we found that ECA model cannot be directly applied to active objects for ISE, because an active object for ISE is a distributed active entity for system control and management. The attributes of a distributed object make the behaviors of active objects for ISE closely related with the changes of time, space and system internal states. It is the time and space problems of active objects that resulted in the many limitations ECA model has in describing behaviors of active objects in the Internet. These problems inspired us to extend the ECA model to the EECA (Extended ECA) model by adding time and space description ability to ECA. EECA model can be used to describe active object on the Internet due to its ability to describe time and space domain.

In the following text, we use $\alpha = \{e_0, e_1, e_2, \ldots, e_n\}$ to represent the set of all the events of a system, $\beta = \{\chi_0, \chi_1, \chi_2, \ldots\}$ to represent the attribute space of all the objects in the system, $\gamma = \{\xi_0, \xi_1, \xi_2, \ldots\}$ to represent the attribute space of the environment in which the objects are running.

Definition 1 An event for active object is a triple-tuple $\{e, t, p\}$, where $e \in \alpha$, $t$ are time stamps with structure of yyyyymmdhhmssms, and $p$ is a URL location.

Definition 2 The internal state of an active object is a five-tuple $\{T, \chi_0, \chi_2, t, p\}$, where $T$ is a global unique transaction identifier, and $\chi_0, \chi_2 \in \beta$ are static and dynamic attributes of an active object, respectively. The meanings of $t$ and $p$ are the same as those defined in Definition 1.

Definition 3 The external state of an active object is a five-tuple $\{T, \xi_0, \xi_2, t, p\}$, where $\xi_0, \xi_2 \in \gamma$ are static and dynamic attributes of an active object’s environment, respectively. The meanings of $T, t$ and $p$ are the same as those defined in Definition 2.

If we mark the event set for an active object as $E$, the set of internal state of an active object as $S_{in}$, then we can define an EECA rule as:

Definition 4 An EECA rule of an active object is a triple-tuple $\{e, f(s_1, s_2), a\}$, where $e \in E$, $s_1 \in S_{in}$, $s_2 \in S_{out}$, $f$ is an assertion on $s_1$ and $s_2$, and $a$ is an order set of the active object’s actions.

Now we define the active object’s semantic model as:

Definition 5 An active object class is a five-tuple $\{N, R, S_{in}, S_{out}, A\}$, where $N$ is the global name system of an active object, which distributes every object of this class with a global unique identifier, $R$ is the EECA set of the active object in this class, $S_{in}$ and $S_{out}$ are the sets of internal and external states of the active objects in this class respectively, and $A$ is the set of actions of that the active object in this class can take.

Definition 6 An active object, which is an instance of an active object class, is an eight-tuple $\{I, R, T, S_{in}, S_{out}, A, T_a, a\}$, where $I$ is global unique active object identifier, $R$ is the EECA set of the active object, $T$ is the trace of the active object’s passed states, $S_{in}$ and $S_{out}$ are the sets of internal and external states of the active object respectively, $A$ is the set of actions that the active object can take, $T_a$ is the trace of the active object’s passed actions, and $a$ is the current action the