Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Automated assembly of Internet-scale software systems involving autonomous agents

Wenpin Jiao[a,b], Yanchun Sun[a,*], Hong Mei[a]

[a] School of Electronics Engineering and Computer Science, Peking University, China
[b] Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education, China

ARTICLE INFO

Article history:
Received 14 October 2009
Received in revised form 22 March 2010
Accepted 30 April 2010
Available online 15 June 2010

Keywords:
Autonomous agent
Software assembly
Software quality
Internet-scale software systems

ABSTRACT

On the Internet, there is a great amount of distributed software entities deployed independently and behaving autonomously. This paper describes an automated approach to constructing Internet-scale software systems based on autonomous software agents. In the approach, the systems are modeled by interconnected divisions and cooperative roles. The approach adopts a dynamic trial-and-evaluation strategy to select high quality autonomous agents to undertake the responsibilities of roles, and implements a special mobile agent, called delegate, carrying the interaction information specified for responsibilities of roles to facilitate the interoperations among autonomous agents. The experiments show that the approach is highly scalable and improves the overall qualities of systems remarkably.

1. Introduction

As the Internet has been transforming from the provider of content services to the platform of computation services, the mode of constructing software systems has changed dramatically.

On the one hand, software entities are dynamically and independently deployed and located at different nodes on the Internet, they are decidedly autonomous and long-lived and they may not be invoked in the sense as traditional components (Garlan, 2002). They have many new characteristics (Garlan, 2002; Shaw, 1999), for example, they can autonomously enter or leave the Internet environment and decide whether to provide services or not. Usually, people understand autonomy of software entities as the taking of independent action. Software entities are autonomous implies that they interact without any intervention from users or other controlling entities (Suzuki and Suda, 2005) and take actions upon their own decisions. Internet-scale software systems (referred to as *Internetware* in (Mei et al., 2006)) have to be dynamically formed as loosely coupled software alliance (Shaw, 1999) by incorporating independent and autonomous software entities spanning different organizations (Wijngaards et al., 2002). In this paper, software entities with autonomy are referred to as *autonomous agent* (*Abbr.* as *AG*).

On the other hand, there are a large number of *AG*s offering the same or similar services with variety of QoS (quality of service) on the Internet. Developers have more opportunities for selecting services providers to construct their software systems according to their functional and non-functional requirements.

Therefore, how to construct high quality internet-scale software systems via automatically selecting and assembling *AG*s has become one of the grand challenges in the era of the Internet.

For approaches to automated Internet-scale software assembly, the following properties are desired at least.

- *Convergence/guaranteed success*: Software systems are required to meet the requirements of users and the assembly approach should be feasible to select appropriate *AG*s to form the systems that achieve the specified objectives.
- *Distribution and local autonomy preservation*: Running on a true distributed environment, Internet-scale software systems are naturally decentralized, *i.e.*, there is not a centralized role controlling or coordinating the cooperation of involved *AG*s. In addition, when participating in the systems, *AG*s are not obliged to provide services and they should also be prevented from becoming the communication bottlenecks due to over-engagement.
- *Optimizing system QoS*: A large number of existing *AG*s on the Internet offer an opportunity for *adaptively* selecting appropriate candidates with desired QoS in the assembly of software systems.
- *Simplicity and efficiency*: While selecting and assembling *AG*s, the approach should support a quick and simple way for constructing stable software systems.

* Corresponding author. Tel.: +86 10 62757670; fax: +86 10 62751792.
E-mail addresses: jwp@sei.pku.edu.cn (W. Jiao), sunyc@sei.pku.edu.cn (Y. Sun), meih@sei.pku.edu.cn (H. Mei).

0164-1212/$ – see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2010.04.074

Currently, in the research areas such as multiagent systems and Web services, much work has coped with the automated integration of software systems via composing software entities (*e.g.*, Egyed and Balzer, 2006; Hu et al., 2008; Huhns et al., 2005; Jureta et al., 2007; Maamar et al., 2005; Motahari Nezhad et al., 2007; Santofimia et al., 2008). However, most of the existing approaches in the literature are centralized and rare of them takes into consideration the quality requirements of the systems and the autonomy of constituent elements.

In principle, the process of assembling Internet-scale software systems by using *AG*s can be divided into three phases (Medjahed et al., 2003; Rao and Su, 2005).

- Plan or specify the business processes for achieving the system objectives.
- Select appropriate *AG*s to undertake the activities specified in the processes.
- Make the selected *AG*s interoperable.

Among these phases, the order of the first two phases is exchangeable, which will result in different assembly strategies. When there are multiple candidates (*i.e.*, *AG*s) to perform an activity in the system, the business process can be specified in advance so that candidates could be selected based on their properties (especially their QoS) to improve the construction of the software system. On the other side, when every *AG* can only undertake a special activity totally different from one another, planning the business process ahead will offer opportunities for accomplishing the system objectives in different ways.

However, in the latter case, planning is generally realized by using AI technologies (Rao and Su, 2005). The low efficiency nature of automatic planning technologies makes it infeasible and unpractical to adopt planning technologies to generate the business processes for application systems.

In our opinions, a reasonable and practical solution to the assembly of Internet-scale software systems has to make compromises on the following aspects.

- *Generation of business processes*: For real-world applications, the positions and interaction relationships of participants usually have been specified in advance so that participants could take actions and cooperate with one another in the desired way to achieve the system goals. It is often not a necessity that there is an automatic reasoning mechanism in the systems to generate the cooperation processes for *AG*s.
- *Selection of AGs*: *AG*s on the Internet are autonomously evolving and they may quit the environment without notifying others. It cannot be assured that *AG*s always reply to the requests for services, so selecting *AG*s once and requesting for services hereafter cannot guarantee the achievement of the systems objectives. Therefore, the selections may have to be carried out all along with the run of the systems and *AG*s had better be re-selected in time according to their QoS so that the success probabilities and overall qualities of the systems could be improved.
- *Interoperation of AGs*: *AG*s are expected to be incorporated into different cooperation applications. They are usually focused on the realizations of their computation logic and contrarily they are not hard-coded with application specific cooperation capabilities. When *AG*s enter application systems to undertake cooperation tasks, they must be informed with application specific cooperation specifications so that they could cooperate with one another in the desired ways.

In this paper, we put forward a decentralized approach to automatic assembly of software systems by using *AG*s distributed on the Internet. In the approach, the software system is organized as a collection of distributed and interconnected *divisions* (similar to *groups* in organizations (Ferber and Gutknecht, 1998) but with more concrete functionalities), in which *roles* take positions to perform activities of the system and they are concerned about their own businesses as well as communications with others. The overall behaviors of the software system are synthesized from the interactions among the roles. The divisions are responsible for *recruiting AGs* to undertake the duties of the roles and *conducting the interactions* of *AG*s to cooperate in pursuit of the achievements of the system's objectives. Meanwhile, the divisions *evaluate the QoS* of the *AG*s according to the returns of the *AG*s and *select* appropriate *AG*s to participate in the activities of the system based on the evaluations.

The main contributions of this work include:

- A new automated approach is proposed for assembling Internet-scale software systems, in which the constituent elements are autonomous.
- A new selection strategy is implemented, by which *AG*s are selected according to their QoS besides their functionalities.
- A new mechanism supporting the interoperation of *AG*s is implemented, in which a special type of mobile agents, called delegate, are used to carry the cooperation information specified by roles and facilitate the interactions of *AG*s.

The paper is organized as follows: Section 2 describes an example system. Section 3 defines the model for Internet-scale software systems and proposes the process of automated assembly of Internet-scale software systems. Section 4 describes the implementation of the approach and experiments for testing the example system. Section 5 compares our approach with some related work. Finally, Section 6 concludes our work and points out some future research directions.

2. An example system

To illustrate how our approach is used to assemble Internet-scale software systems, we consider a system that acts as running example throughout this paper.

Considering a university human resources MIS (Management Information System), we are required to assemble it by using *AG*s on the Internet.

In the department of human resources, there are three divisions:

- Talent management division, with responsibilities for the employment, dismissal/resignation, intra-university transfer, promotion, and retirement of staffs.
- Salary/allowance division, in charge of the determination and adjustment of the salaries and allowances of staffs.
- Personnel files management division, responsible for the maintenance and track of the personnel and payment information of staffs.

In the divisions, every responsibility is assigned to a personnel officer (or a role) specially. For instance, the officers and their responsibilities in the divisions are listed in Table 1.

Besides fulfilling their daily work, the personnel officers often communicate and collaborate with others (including the ones in other divisions) while carrying out their work. For an interaction between persons, it is mostly concerned with when or under what situations officers will communicate with one another. For instance, when a faculty member is promoted, the officer taking charge of promotion will notify the promotion information to the ones responsible for adjusting the promotee's salary and allowance and meanwhile ask the files manager to update the related infor-