可配置网络式软件系统的可用性预计研究

杨格兰[1] 孟令中[2]
(湖南城市学院信息科学与工程学院 益阳 413000)[1]
(中国科学院软件研究所基础软件测评实验室 北京 100190)[2]

摘 要 在复杂系统的建模与仿真研究的基础上,提出了一种基于多Agent的可配置网络式软件系统的可用性预计方法。首先介绍了多Agent系统建模与仿真方法;其次分析了可配置网络式软件系统的特点;然后在研究基于多Agent的网络式软件系统建模与仿真的基础上,研究可配置的行为模型,并建立了基于多Agent的可配置网络式软件系统可用性仿真方法;最后利用Netlogo仿真平台,结合实例对可配置的作用进行了可用性预计,并验证了本方法的有效性。

关键词 多Agent,可配置,网络式软件系统,可用性预计
中图法分类号 TP311.5 **文献标识码** A

Research on Reconfiguration Networked Software System Availability Prediction

YANG Ge-lan[1] MENG Ling-zhong[2]
(School of Information Science and Engineering, Hunan City University, Yiyang 413000, China)[1]
(Laboratory of Fundamental Software Testing and Evaluation, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)[2]

Abstract Based on modeling and simulation for complex systems, a new approach to predict networked software system availability was proposed by multi-agent system modeling and simulation. Firstly, the method of multi-agent system modeling and simulation was introduced. Secondly, the characteristics of reconfiguration networked software system were analyzed. And then after researching on the approach to multi-agent based modeling and simulation for networked software system and behavioral models of reconfiguration, the new strategy which can be used for availability prediction was addressed. Finally, in order to verify the effectiveness, a case study was taken based on the approach, which was realized on the Netlogo simulation platform.

Keywords Multi-agent, Reconfiguration, Networked software system, Availability prediction

1 引言

以网络为基础的软件系统的应用与服务已成为国民经济可持续发展、社会生活与国家安全保障的重要支柱,同时随着网络的快速发展,网络化软件系统也被广泛地应用于各个领域,例如网上银行、电子政务、ERP类软件以及军用的C4I类软件等。

网络化软件系统是部署在网络环境中的复杂软件系统,其拓扑结构和软件行为可以动态演化[1]。其中单个子系统具有自主性和反应性,即具有相对独立性、主动性、自适应性以及感知外部运行和使用环境并对系统演化提供有用信息的能力;多个子系统具有协同性和演化性,即多个子系统的互连、互通、协作和联盟以及根据需求动态调节功能、结构和行为的能力。

DESEREC是第六届欧盟项目框架下的信息社会技术下优先、重点考虑的集成项目,目的是建一个"面向全球的可信性和安全性框架"。其中一项目标就是"允许服务可以动态配置",从而建立安全、可信任的网络和服务平台。

可信性dependability作为集合性术语,首先由法国学者laprie于1985年提出。他认为可信性是计算机系统的一种性质,使得它所提供的服务有足够理由认为是可信赖的。这为构建可信的计算机系统提供了一个概念框架和明确的开发模式[2]。Laprie于2004年明确了软件的可信性包含可用性、可靠性、安全性、保密性、完整性、可维护性[3]。其中可用性指一个系统在需要使用的时候可以运行和可以进入的程度。

对于部分单机软件,在软件部分功能发生失效的情况下,需要人工修复或者在一定时间内自恢复。但是对于用在关键领域的网络式软件系统,例如航空航天软件、通信软件或军用指控软件系统等,如果失效时无法及时地修复,会引起严重的后果。

可配置软件系统是指该软件系统能在构件失效时,根据不同的设计原理进行再配置,实现快速再配置,从而完成需要的功能需求。这样利用可配置系统的作用,提高网络式软件系统的可用性。

到稿日期:2012-01-03 返修日期:2012-04-12 本文受国家科技支撑计划课题(2012BAD35B07),湖南省教育厅优秀青年项目(12B023),湖南省科技计划课题(2011FJ3022)资助。

杨格兰(1975—),男,硕士,副教授,主要研究方向为软件可信性,E-mail:glyang@mail.ustc.edu.cn;孟令中(1981—),男,博士,主要研究方向为软件可信性、软件测试。

可配置网络式软件系统由多个子系统构成，子系统的组成元素之间相互调用，以完成用户所需要的任务，同时对外从整体上涌现出软件系统的可靠性、可用性等。传统方法有利于基于功能部件、服务进行可靠性的研究[4,5]，利用 Bayesian 网络对软件工作状态建模，从而进行可靠性和可用性研究[6,7]，利用基于组件的方法对软件可靠性进行预计和度量[8]以及可靠性分析[9,10]。但是对于网络式软件系统而言，由于其规模庞大、功能交互繁多、组成的构件之间的相互调用关系复杂以及动态再配置的过程，以上传统方法无法适用于可配置网络式软件系统的可用性预计。

本文将可配置网络式软件系统看作一类复杂系统，借鉴相关利用多 Agent 建模与仿真对可靠性的研究[11]，对可配置网络式软件系统的可用性进行研究。首先，介绍了 Agent 的基本概念及多 Agent 建模与仿真的主要思路；其次，在分析可配置网络式软件系统特点的基础上，分析了基于多 Agent 的可配置过程，从而进行可用性建模与仿真方法研究；第三，结合实际的可配置网络式软件系统进行实例应用，并对仿真结果进行分析；最后，对全文工作进行总结和展望。

2 多 Agent 系统的建模与仿真

Agent 是指驻留在某一环境下能够自主、灵活地执行动作以满足设计目标的行为主体，具有自治性、社会性、反应性和能动性等特征[12]。Agent 的结构包括 Agent 的各个抽象成分、每个抽象成分在 Agent 运作过程中的地位和作用以及这些抽象成分之间的相互关系。目前 Agent 的结构主要包括 3 类：反应式结构、思考型结构和混合型结构[13]。

多 Agent 系统(Multi-Agent System，MAS)是由两个或更多的 Agent 构成，其中每个 Agent 是独立的行为实体，并封装了状态和行为，同时不同的 Agent 通过通信进行相互交互。MAS 的体系结构可以分为集中式、分布式和混合式。

MAS 的基本思想是将许多子系统看作是由多个自治的 Agent 组成，Agent 之间的相互作用是系统宏观特性出现的根源，通过建立多 Agent 模型，可以更好地理解和分析这些系统。

多 Agent 仿真方法的本质特征是采用多 Agent 视角建立实际系统的概念模型。其主要流程是实际系统——概念模型——仿真模型——仿真结果——结论。主要的反馈环节是校核验证，通过将仿真运行结果与实际数据进行对比分析，促进对概念模型和仿真模型的改进，直到满足相似性要求。

3 基于多 Agent 的可配置网络式软件系统可用性研究

传统的复杂系统研究方法往往采用某些数学手段，如微分方程等，来宏观地刻画该类系统，这类自上而下的方法对于复杂系统初期的研究做出了很大的贡献。但是随着对复杂系统的深入研究，发现这类自上而下的方法将复杂系统中所有个体都看成同类，因此忽略了个体的差异与特征，故这类方法无法刻画局部细节行为以及局部变化对整体的影响。为此，一类自底向上的方法随之产生，其中多 Agent 的复杂系统研究方法成为一种重要方法。

3.1 可配置的网络式软件系统特点研究

可配置的网络式软件系统的可用性应该以服务为本，而不是以整个系统为本。如果一个可配置的网络式软件系统在某个子系统的构件发生故障后，通过再配置的方法仍能保证所有执行的服务按期完成，那么就能确定其可配置策略是成功的。

因此，本文对可配置的网络式软件系统的特点以及对可用性进行分析，为便于阐述，对文中用到的概念定义如下。

定义 1(软件服务) 软件服务是指网络式软件系统提供给用户使用的一组软件功能。

定义 2(软件业务) 软件业务是指用户对软件服务的使用，包括软件业务涉及的软件服务集合、对服务的使用方式和对业务性能和质量方面的要求。

可配置软件系统的基本特征包括：网络化、可靠性、可恢复性、连续性。其中网络化指系统能够提供性能优良的服务，而且性能的提高不以停机时间的增加为代价，一般通过联机进行规模的扩大、调整和重组来实现。可靠性指系统出现故障的次数很少，平均无故障时间足够长。可恢复性指具有迅速从系统故障或灾难中恢复的能力。连续性是指软件系统可以提供不间断的软件业务，当系统中某一结点出现故障时，该结点上的任务可以立刻转移到系统中其它的正常结点上，而且数据库的管理与维护是联机进行，一般不会停机。

3.2 基于多 Agent 的网络式软件系统建模与仿真研究

借鉴软件的体系结构，将整个可配置网络式软件系统的静态组成利用分层模式、面向对象模式和事件驱动模式混合描述。其中分层模式是整个软件系统的基本结构模拟，面向对象模式是对每一层的具体构件模拟，事件驱动模式用来模拟构件之间的相互作用与通信方式。

软件体系结构的模型，可以看成是由很多个构件和连接件构成。因此在建模过程中，把构件抽象成节点，连接件抽象成边，就能构成一个拓扑网络。多个子系统之间利用连接件相互连接，就构成了复杂网络。同时把构件和连接件抽象成不同结构类型的 Agent 进行表示，使复杂网络的节点和边具有智能性，从而进行多 Agent 宏观模式的有效分析。建模仿真框架如图 1 所示。

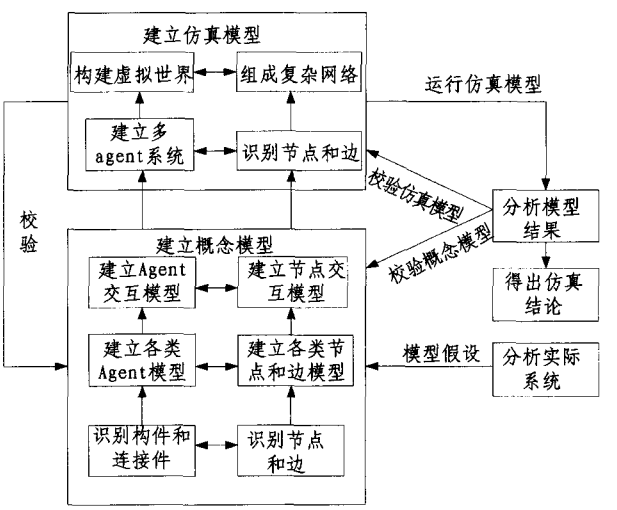

图 1 基于多 Agent 的可配置软件系统建模与仿真框架

其中，节点 Agent 利用认知型 Agent 进行建模，如图 1 所示，构件 Agent 从网络环境中接收需要的信息，根据内部状态进行信息整合或信息判断，之后匹配知识库中的信息制定规划，最后根据目标产生一系列的动作，从而对外部的网络环境

产生影响。

基于图 2 所示的认知型构件 Agent 结构模型，给出构件 Agent 的形式化描述，用一个多元组表示：

Com_Agent::=〈Comi, Com_set, Com_KB, Com_PS, Com_GS, Com_AS〉

式中，Comi 指构件 Agent 的序号；Com_set 指构件 Agent 的状态集合，包括构件的可靠度、状态等信息；Com_KB 指构件 Agent 的知识库；Com_PS 指构件 Agent 的规划集合；Com_GS 指构件 Agent 的目标集合；Com_AS 指构件 Agent 的动作集合。

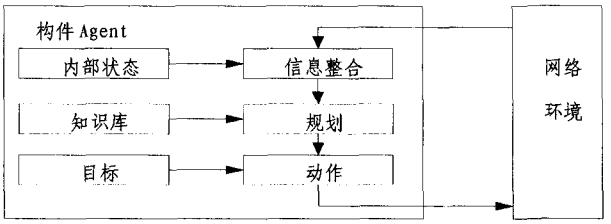

图 2　认知型构件 Agent 结构建模

连接件 Agent 利用反应型 Agent 进行建模，如图 3 所示。连接件 Agent 从网络环境中接收需要的信息，进行信息的整合，根据内部的规则库进行动作反应，最后将处理后的信息反馈给网络环境。

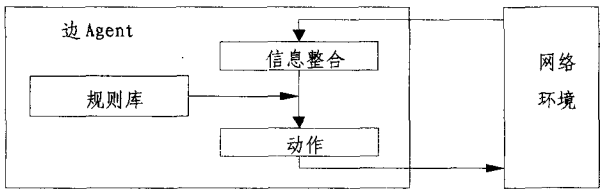

图 3　反应型连接件 Agent 结构建模

基于上述的反应型连接件 Agent 结构模型，给出连接件 Agent 的形式化描述，用一个多元组表示：

Con_Agent::=〈Coni, Con_set, Con_PS, Con_AS〉

式中，Coni 指连接件 Agent 的序号；Con_set 指连接件 Agent 的状态集合，包括连接件的可靠度、状态等信息；Con_PS 指连接件 Agent 的规划集合；Con_AS 指连接件 Agent 的动作集合。

3.3　基于多 Agent 的网络式软件系统可配置研究

网络式软件系统由多个子系统组成，每个子系统又有多个构件和连接件组成，同时各个子系统也通过不同的连接件进行数据或控制传输。这样在完成用户的软件业务要求时，需要调用多个软件服务，每个服务又调用了多个构件和连接件，即模拟完成每一个用户的业务的使用，需要多个构件 Agent 和连接件 Agent 共同完成。因此，多 Agent 的协调与交互的建模与控制变得十分重要。基于此，提出了一个适合于网络式软件系统的多 Agent 协调与交互模型，其模型利用多种 Agent 进行辅助运行，如图 4 所示。

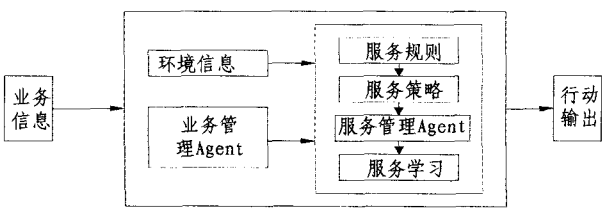

图 4　多 Agent 协调与交互模型

建立业务管理 Agent(Bus_Agent)和服务管理 Agent(Ser_Agent)用于运行多 Agent 的协调与交互模型，其中，业务管理 Agent 指管理服务使用的 Agent，服务管理 Agent 设定了网络软件系统需要为用户完成的服务，下面给出它们的形式化描述：

Bus_Agent::=〈Busi, Bus_set, Bus_ser_set, Bus_state〉

式中，Busi 指业务 Agent 的序号；Bus_set 指业务 Agent 的组成集合，包括业务的剖面、业务分支的转移概率等；Bus_ser_set 指业务管理包含的服务集合；Bus_state 指业务管理 Agent 的状态。

Ser_Agent::=〈Seri, Ser_S, Ser_P, Ser_Per, Ser_Trans〉

式中，Seri 指服务 Agent 的序号；Ser_S 表示状态集合；Ser_P 表示感知集合；Ser_Per 表示 E→Ser_P 的感知函数，将环境状态映射为感知输入，E 为环境状态集；Ser_Trans 表示 Ser_P * Ser_S→Ser_S 的决策函数，根据感知输入和当前状态实现服务的变化。

但是，在发生软件失效的情况下，并不是所有的构件和连接件都可以通过可配置的过程来满足用户的软件业务需求。在进行设计时，一般对关键软件业务进行服务可配置设计。

同时，在前文基础上，将多 Agent 的可配置过程利用形式化建模如下：

Re_Model::=〈Re_seti, 〈Re _act_set〉, Re _PS, Re _Cor〉

式中，Re_seti 指参与可配置过程的多种 Agent 的集合的序号；〈Re_act_set〉指每个 Agent 的动作集合，在一次可配置过程中，每个 Agent 的有效动作的组合构成了动作集合；Re _PS 指有关可配置的服务规则库；Re_Cor 指服务策略，包括多个 Agent 之间的协调、协作和协商等。

Re _act_set::=〈Re _act_seti, 〈Re _act _com〉, 〈Re _act _con〉〉, Re _act _com::=〈comj, comj_set〉, Re _act _con::=〈conj, conj_set〉

式中，Re _act_seti 指可配置的动作序号；〈Re _act _com〉指需要可配置的构件集；〈Re _act _con〉指需要可配置的连接件集；comj 指发生失效的构件；comj_set 是指通过可配置动作可以完成发生失效的构件 comj 功能的一组构件的集合；conj 指发生失效的连接件；conj_set 是指通过可配置动作可以完成发生失效的连接件 conj 功能的一组连接件的集合。

3.4　基于多 Agent 的可配置网络式软件系统可用性仿真研究

在对可配置的网络式软件系统的可用性进行建模与仿真时，不仅需要所包含的构件 Agent、连接件 Agent、业务管理 Agent 和服务管理 Agent，还需要一些功能性 Agent，它们用来实现特定的功能，从而保证仿真的顺利运行以及仿真数据的收集。其中包括调度 Agent(Sch_Agent)、监测 Agent(Mon_Agent)、通讯 Agent(Cor_Agent)、数据处理 Agent(Dat_Agent)和人机接口 Agent(Gui_Agent)等。其中调度 Agent 设定仿真时间以及启动仿真系统运行，并当条件满足时终止仿真；监测 Agent 在仿真过程中收集数据，以便实时显示数据；通讯 Agent 用来提供多个 Agent 之间的通讯方法；数据处理 Agent 将仿真运行中的各种数据进行处理，用来描述系统的宏观表现；人机接口 Agent 用来接收输入数据和输出数据。该建模与仿真平台如图 5 所示。

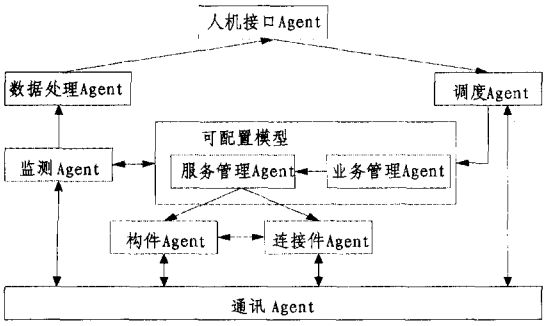

图 5　基于 MAS 的网路化软件系统连续性建模与仿真平台

各类 Agent 的属性与行为模型描述如下：

Sch_Agent=〈Schi,Sch_clock,Sch_Cori〉,其中 Schi 指调度 Agent 的序号;Sch_clock 指仿真钟;Sch_Cori 指连出的通讯 Agent;

Mon_Agent=〈Moni,Mon_type,Mon_ Cori 〉,其中 Moni 指监测 Agent 的序号;Mon_type 指监测 Agent 的类型,Mon_Cori 指连出的 Agent;

Cor_Agent=〈Cori,Cor_type,in_agents,out_agents〉,其中 Cori 指通讯 Agent 的序号;Cor_type 指通讯 Agent 的类型,分为监测 Agent 通讯、构件 Agent 通讯、连接件 Agent 通讯等;in_agents 指连入的 Agent;out_agents 指连出的 Agent;

Dat_Agent=〈Dati,Data_type〉,其中 Dati 指数据处理 Agent 的序号;Data_type 指数据处理 Agent 的类型;

Gui_Agent=〈Guii,Gui_type〉,其中 Guii 指人机接口 Agent 的序号;Gui_type 指人机 Agent 的类型。

网络式软件系统的可用性利用可用性 A_0 进行度量,指网络式软件系统能够提供软件业务的总时间,其中包括使用时间(OT)、待机时间(ST)、修护性维修时间(TCM)、预防性维修时间(TPM)等,其度量模型如式(1)所示。

$$A_0=\frac{MUT}{MUT+MDT} \tag{1}$$

式中,MUT 指平均能工作时间,即 $OT+ST$;MDT 指平均不能工作时间,即 $TCM+TPM$。

在仿真平台中,通过调度 Agent 协调仿真系统中各种 Agent 的行为,来保证仿真系统的进行,并通过仿真钟采用等步长推进方式,根据时间节拍推进服务 Agent、作用 Agent 和构件 Agent 等,这样可以保证各个 Agent 的时间同步,并对其进行状态更新。同时利用仿真钟对可用性度量模型中的各种时间进行统计,从而计算出可配置网络式软件系统的可用性。

4　仿真研究应用

为验证本文提出的基于多 Agent 的可配置网络式软件系统可用性预计方法的可行性和关键技术的有效性,选取某大型可配置网络式软件系统,搭建其可用性建模与仿真系统,针对其可用性进行了仿真,并对仿真结果进行了分析。

利用前文的研究内容在 NetLogo 平台下进行编程,对某大型可配置网络式软件系统的可用性进行仿真。NetLogo 是一个可编程的、能够模拟各类复杂对象行为演化规律的建模仿真平台。它由美国西北大学连接学习与计算机建模中心开发,特别适合对随时间演化的复杂系统进行建模与仿真。

为了简化仿真模型,对网络式软件系统的软件失效做出如下假设：

1)构件和连接件只存在失效和正常运行这两种状态;

2)构件和连接件发生的失效之间相互独立;

3)构件和连接件的失效率都服从特定的分布;

4)对于组成关键业务的构件发生失效时,其可配置行为可以瞬时完成。

在此基础上,利用前文介绍的基于多 Agent 的方法对可配置网络式软件系统的可用性进行建模,并运行仿真系统,系统的变化状态如图 6 所示。

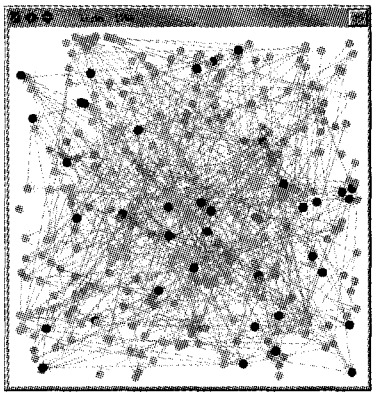

图 6　基于多 Agent 的可用性仿真结果

图 6 中,节点表示抽象出的构件 Agent,共计 306 个,其中度为 0 的点表示这个构件 Agent 可以单独提供某些软件业务;有向边表示连接件 Agent,共计 459 条;黑色的节点表示发生失效的构件,属于组成非关键软件业务的构件,失效后无法通过可配置过程恢复到可以运行的状态;深灰色的节点表示发生失效后,通过可配置过程恢复到可以运行状态的构件,这类构件属于组成关键软件业务的构件。

在仿真系统运行一段时间内,其可用性变化曲线如图 7 所示。其中,横轴表示仿真运行的时间;纵轴表示系统的可用性,为使图像便于显示,区间设为[50%,100%];黑色曲线表示不可配置系统的可用性变化趋势,灰色曲线表示可配置系统的可用性变化趋势。

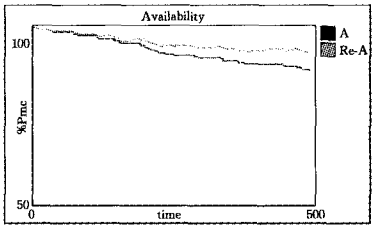

图 7　可配置系统与不可配置系统的可用性比较

分析以上结果,在网络式软件系统运行初期,两种系统的可用性差不多。但是随着仿真时间的推进,可配置系统的可用性明显高于不可配置系统的可用性,并且高出的幅度逐渐增大。这说明可配置的行为对于网络式软件系统的可用性有着明显的作用。因此,在一些关键系统中,通过可配置的方式,可以有效地提高该类系统的可用性。

结束语　本文以多 Agent 技术为支撑,从复杂系统的角度分析和解决了可配置网络式软件系统的可用性预计的问题。利用基于多 Agent 的建模仿真方法,验证了随着时间的运行,可配置系统比不可配置系统的可用性会有明显的提高。在今后的研究工作中,利用该方法将进一步分析以下方面:组成关键软件业务的构件确定方法,具有可配置行为的构件的数量优化方法等。

参 考 文 献

[1] Feng Zai-wen, He Ke-qing, Ma Yu-tao, et al. Towards individualized requirements specification evolution for networked software based on aspect[C]// Making Globally Distributed Software Development a Success Story. Leipzig, Germany, 2008

[2] Laprie J. Dependable computing and fault tolerance: concepts and terminology[C]// IEEE the 15th International symposium on fault-tolerant computing. Michigan, USA, 1985

[3] Avizienis A, Laprie J C, Randell B, et al. Basic Concept and Taxonomy of Depenable and Secure Computing [J]. Dependable and Secure Computing, 2004, 1(1): 11-33.

[4] Caban D. Dependability of Large Reconfigurable Information Systems [C]// IEEE DepCos-RELCOMEX. Brunow, Poland, 2009

[5] Boon Y O, Huah Y C, Cheah Y N. Dynamic service placement and redundancy to ensure service availability during resource failures [C]// 2010 International Symposium in Information Technology (ITSim). Kuala Lumpur, Malaysia, 2010

[6] Droguett E L, Moura M D C, Jacinto C M, et al. A semi-Markov model with Bayesian belief network based human error probability for availability assessment of downhole optical monitoring systems [J]. Simulation Modelling Practice and Theory, 2008, 16(10): 1713-1727

[7] Kang C W, Golay M W. A Bayesian belief network-based advisory system for operational availability focused diagnosis of complex nuclear power systems [J]. Expert Systems with Applications, 1999, 17(1): 21-32

[8] Marko P, Antti E, Eila O. The Reliability Estimation, Prediction and Measuring of Component-Based Software [J]. The Journal of Systems and Software, 2011, 84(6): 1054-1070

[9] Chao J H, Chin Y H. An Adaptive Reliability Analysis Using Path Testing for Complex Component-Based Software Systems [J]. IEEE Transactions on Reliability, 2011, 60(1): 158-170

[10] Haiyang H. Reliability Analysis for Component-based Software System in Open Distributed Environments [J]. International Journal of Computer Science and Network Security, 2007, 7(5): 193-202

[11] Meng Ling-zhong, Lu Min-yan, Lai Jing, et al. The Multi-Agent Based Reliability Simulation Approach for Networked Software System [J]. International Journal of Advancements in Computing Technology, 2011, 3(10): 197-207

[12] Wooldridge M, Jennings N R. Agent Theories, Architectures, and langeuages: A Survey [C]// Workshop on Agent Theories Architectures and Languages. Montreal, Canada, 1995

[13] Wooldridge M J, Jennings N R. Intelligent Agent: theory and practice [J]. Knowledge Engineering Review, 1995, 10(2): 115-152

(上接第 75 页)

参 考 文 献

[1] Shamir A. Identity based cryptosystems and signature scheme [C]// Crypto 1984, LNCS 196. Springer-Verlag, 1984: 47-53

[2] Al-Riyami S S, Paterson K G. Certificateless public key cryptography[C]// Advance in Cryptology-ASIACRYPT 2003. 2003: 452-473

[3] Baek J, Safavi-Naini R, Susilo W. Certificateless public key encryption without paring[C]// Information Security Conference (ISC). 2005: 134-148

[4] Dent A W, Libert B, Paterson K G. Certificateless encryption schemes[Z]. Strongly cryptography, 2008: 344-359

[5] Huang Xin-yi, Mu Yi, Susilo W, et al. Certificateless signature revisited[C]// ACISP. 2007: 308-322

[6] Liu J K, Au M H, Susilo W. Self-generated-certificate public key cryptography and certificateless signature/encryption scheme in the standard model: extended abstract[C]// ACM Symposium on Information, Computer and Communications Security (ASIACCS). 2007: 273-283

[7] Huang Qiong, Wong D S. Generic certificateless encryption in the standard model[C]// IWSEC. 2007: 278-291

[8] Huang Qiong, Wong D S. Generic certificateless key encapsulation mechanism[C]// ACISP. 2007: 215-229

[9] Dent A W. A survey of certificateless encryption schemes and security models[J]. Int. J. Inf. Sec., 2008, 7(5): 349-377

[10] Bentahar K, Farshim P, Malone-LEE J, et al. Generic constructions of identity-based and certificateless KEMs[J]. J. Cryptology, 2008, 21(2): 178-199

[11] Fiore D, Gennaro R, Smart N P. Constructing certificateless encryption and id-based encryption from id-based key agreement [C]// Pairing-Based Cryptography. 2010: 167-186

[12] Mandt T K, Tan C H. Certificateless authenticated two-party key agreement protocols[C]// Proc. ASIAN 2006. 2006: 37-44

[13] Shao Zu-hua. Efficient authenticated key agreement protocol using self-certified public keys from pairings[J]. Wuhan University Journal of Natural Sciences, 2005, 10(1): 267-270

[14] Wang Sheng-bao, Cao Zhen-fu, Wang Li-cheng. Efficient certificateless authenticated key agreement protocol from pairings[J]. Wuhan University Journal of Natural Science, 2006, 11(5): 1278-1282

[15] Swanson C M. Security in key agreement: Two-party certificateless scheme[D]. University of Waterloo, 2008

[16] Geng Man-man, Zhang Fu-tai. Provably secure certificateless two-party authenticated key agreement protocol without pairing [C]// International Conference on Computational Intelligence and Security. 2009: 208-212

[17] Hou Meng-bo, Xu Qiu-liang. A two-party certificateless authenticated key agreement protocol without paring[C]// 2nd IEEE International Conference on Computer Science and Information Technology. 2009: 412-416

[18] He De-biao, Chen Yi-tao, Chen Jian-hua, et al. A new two-round certificateless authenticated key agreement protocol without bilinear prirings [J]. Mathematical and Computer Modelling, 2011, 54(11/12): 3143-3152

[19] Yang Guo-min, Tan C-H. Strongly secure certificateless key exchange without pairing[C]// ASIACCS'11. 2011: 22-24

[20] Goldwasser S, Micali S, Rivest R. A digital signature scheme secure against adaptive chosen-message attack[J]. Siam J. Computing, 1988, 17(2): 281-308