基于 Petri 网的 Web 服务组合与分析*)

闫春钢　蒋昌俊　李启炎

(同济大学计算机科学与技术系　上海 200092)[1]

(国家高性能计算机工程技术研究中心同济分中心　上海 201804)[2]

摘　要　Web服务为互联网提供了一种新的应用环境。然而，Web服务还有许多需要进一步研究的问题。Web服务的组合及其验证就是需要深入研究的问题。本文针对通用构件描述语言(UCDL)提出一种 Petri 网模拟和验证方法，即对于 Web 服务的元活动和构件，提出相应的 Petri 网模型和建模方法。在此基础上进一步研究了 Web 服务系统 Petri 网的语言表达式生成算法，从而为 Web 服务系统的验证分析提供了有效工具。

关键词　Petri 网，化简技术，Web 服务组合，语言，表达式

The Composition and Analysis of Web Service Based on Petri Net

YAN Chun-Gang　JIANG Chang-Jun　LI Qi-Yan

(School of Electronics and Information Engineering, Tongji University, Shanghai 200092)[1]

(Tongji Branch, National Engineering & Technology Center of High Performance Computer, Shanghai 200092)[2]

Abstract　Web service provides a new application environment for Internet. However, Web service still has many problems need further study, e. g. Web service composition and verification. This paper presents a Petri Net modeling and verifying method for Universal Component Description Language (UCDL), that is to say, presents corresponding Petri net models and modeling method for Web service meta-activities and components. On the basic of above all, research of Petri net language expression generation algorithm for Web service system is further carried out. Thereby, an efficient tool for verification and analysis of Web service system is provided.

Keywords　Petri net, Reduction technique, Web service combination, Language, Expression

1　引言

Web服务组合的目的在于当现实环境所需求的功能变得复杂，单一 Web 服务所提供的功能无法满足使用者的需求时，就必须藉由 Web 服务的组合来建立新的服务，提供更复杂的功能。Chandrasekaran[1]指出，当个别的服务受限于它们所提供的功能时，我们必须以网络流程的形式去组合现存的网络服以建立新的功能。Snell[2]认为，Web 服务合成的目的是让跨企业流程的无缝整合和交易生命周期具有相同的型态，并且能让许多的 Web 服务使用。Snell 指出了 Web 服务组合的另一个目的就是：要让具有企业流程功能的 Web 服务能够通过发布的程序，让更多的应用程序开发者使用。Zeng[5]指出，以流程为基础的 Web 服务是一种新兴的组织内和跨组织的自动化企业流程方法。文[6]讨论了服务合成中的 QoS 问题。文[7] 研究了自服务环境中的服务合成方法。文[8]研究了服务网格的自适应组态问题。文[9]基于 WSDL 提出一种通用构件的描述语言，该语言具有统一规范的描述接口，能够屏蔽多种构件库之间的差异特性，将为 Web 服务的构件组织、发现、组合、存储和管理奠定了良好基础。这些研究无疑对 Web 服务的合成研究起到推进作用，但关于合成过程的验证分析尚需进一步研究。Petri 网是一种适合于分布式系统验证分析的形式化方法。其中 Petri 网化简是一种有效的分析技术。文[10]曾对 Petri 网的一类特定系统(T-图)研究过化简分析方法。文[11]针对自由选择的工作流网系统，讨论了化简技术对合理性的保持性问题。文[12]进一步扩充了化简规则，并讨论了化简过程对语义的保持关系。另一方面，研究 Petri 网系统动态行为的另一有效途径是 Petri 网语言的分析方法。文[13～19]在这方面开展过诸多研究。基于语言表达式可以在不生成 Petri 网状态可达图的情况直接分析 Petri 网的动态性质和性能，因此生成 Petri 网的语言表达式非常重要。但一般情况下 Petri 网的语言表达式并不容易，大多通过其状态可达图的化简获得，这往往会出现状态爆炸问题。考虑到工作流网系统是一类结构相对简单的特定 Petri 网系统，化简技术对于其语言表达式生成会带来很好的效果。本文针对 UCDL 构件描述语言，给出了 Web 服务的元活动和构件的 Petri 网模型和建模方法。在此基础上进一步研究了 Web 服务系统 Petri 网的语言表达式生成算法，从而为 Web 服务系统的验证分析提供了有效工具。

2　预备知识

文[9]给出了 Web 服务的活动元素及其模型、运行脚本描述表。本文在此基础上增加对应活动元素的 Petri 网模型的描述，形成如表 1。本文考虑的 Petri 网是 $\Sigma=(P,T;F,M_0)$。有关 Petri 网的基本概念可见文[19, 20]。

3　Web 服务构件的网模型组合

文[9]讨论了 Web 服务的组装问题，给出了相应的步骤。但过程中对于模型的正确性验证并没有考虑，这里在文[9]中

*)国家自然科学基金(60125205,60534060)资助。闫春钢　副教授，博士研究生，研究方向为 Petri 网理论、工作流建模方法、Web 服务。

方法的基础上，考虑了相应的 Petri 网建模型，并利用 Petri 网的相关分析方法可以对模型进行正确性验证，从而保证了在复杂组合过程中系统的正确性。不仅如此，我们还可以基于 Petri 网的技术对系统进行事务序列的代数刻画和性能分析，下一节将讨论这些内容。

算法 1 Web 服务组合算法

输入：Web 服务活动元素集、(元素)构件组合关系

输出：Web 服务构件组合 Petri 网模型及其脚本

Step 1. Web 服务活动元素的构件图形化建模；

Step 2. 根据表 1，由 Web 服务活动元素集和元素构件组合关系构造构件 Petri 网模型；

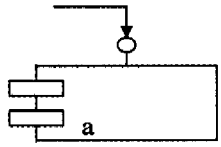
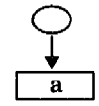
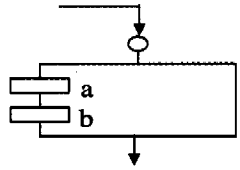
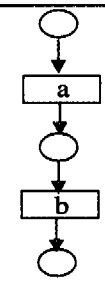
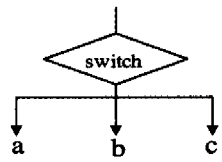
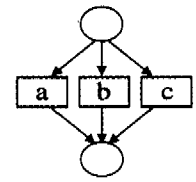
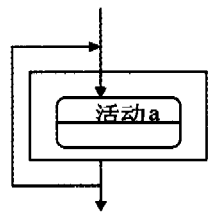
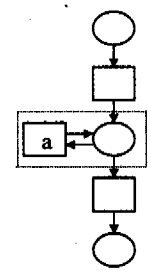
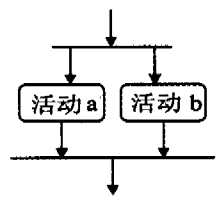
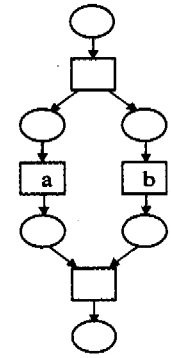
Step 3. Web 服务构件的图形组合；

Step 4. 基于构件的组合图形，对每一个构件及其组合关系按表 1 形成构件组合网模型；

Step 5. 验证构件组合网模型，若不正确，则修改上述图形模型及网模型，否则做 Step6；

Step 6. 根据表 1 和图形组合，生成运行脚本，结束。

表 1 Web 服务构件的活动元素及其模型、运行脚本描述表

名称	示例	Petri网模型	对应的运行脚本
调用			<invoke Interface ="interface" component="MP3Decoder"> <Targer linkName ="viewer-to-mp3"> </ invoke >
顺序			<sequence> < invoke a > ..... </ invoke a> < invoke b> ..... </ invoke b> </ sequence >
分支			<swich> <case condition ="case1"> < invoke a> ..... </ invoke a> </ case > <case condition ="case2"> < invoke b> ..... </ invoke b> </ case > </swich>
循环			<while condition ="bool-expr"> <sequence a > ..... </ sequence a > </ while >
并行			< flow > < invoke a > ..... </ invoke a > < invoke b > ..... </ invoke b > </ flow >

例 1 一个 ATM 机的操作构件包括：(1)转款构件：{转帐、转入帐号、转入数}；(2)取款构件：{取款、取出数}；(3)操作构件：{查款、取款、转款}；(4)重复操作构件：{操作、返回}；(5)过程构件：{开始、插卡、密码、重复操作、结束}。

首先，我们给出元活动对应的名称与符号表(如表 2)。

表 2 Web 服务元活动的对应表

元活动或构件	对应名称	对应符号
开始	Start	*a*
插卡	Insert-Card	*b*
密码	Password	*c*
查款	Check Account	*d*
取款	Drawing	*e*
取出数	Drawing-Number	*f*
转帐	Transfer	*g*
转入帐号	Transfer-in-Account	*h*
转入数	Transfer-in-Number	*i*
返回	Return	*j*
结束	Ene	*k*

产生的 Petri 网模型如图 1。

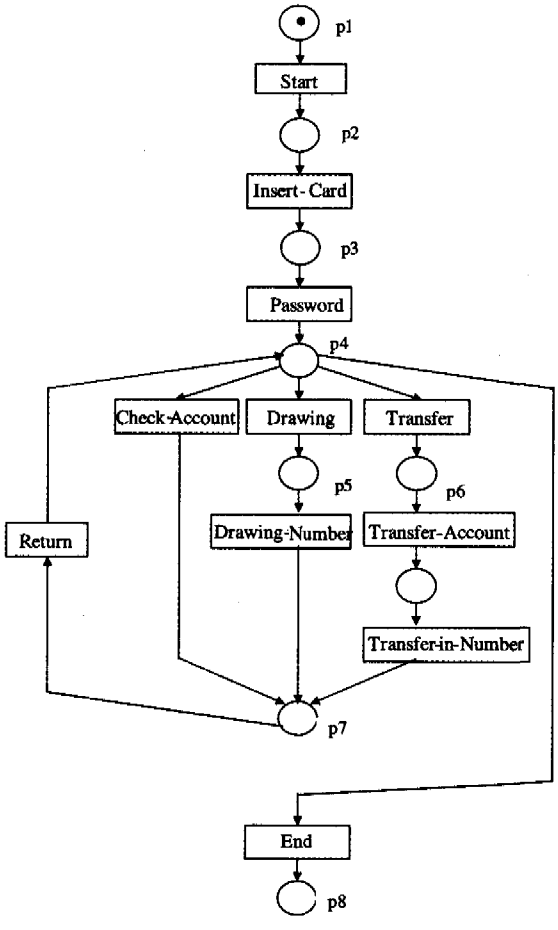


图 1 例 1 的 Petri 网模型

产生的 UCDL 脚本语言如下：

```
〈 process Interface="interfacename" 〉
                              component="ATM operation"
〈 target linkName="ATMgui" 〉
〈 flow name="DeadLockFlow" 〉
  〈 links 〉
    〈 link name="p1"/ 〉
    〈 link name="p2"/ 〉
    〈 link name="p3"/ 〉
    〈 link name="p4"/ 〉
    〈 link name="p5"/ 〉
    〈 link name="p6"/ 〉
    〈 link name="p7"/ 〉
    〈 link name="p8"/ 〉
    〈 link name="p9"/ 〉
  〈 /links 〉
〈 sequence 〉
  〈 invoke name="Start" 〉
    〈 source linkName="p1" 〉
    〈 target linkName="p2" 〉
  〈 / invoke 〉
  〈 invoke name="Insert-Card" 〉
    〈 source linkName="p2" 〉
    〈 target linkName="p3" 〉
  〈 / invoke 〉
  〈 invoke name="Password" 〉
    〈 source linkName="p3" 〉
    〈 target linkName="p4" 〉
  〈 / invoke 〉
〈while condition="No Return" 〉
 〈 switch 〉
   〈 case condtion="Check-Account"〉
     〈 invoke name="Check-Account" 〉
       〈 source linkName="p4" 〉
       〈 target linkName="p8" 〉
     〈 / invoke 〉
   〈 / case 〉
  〈 case condtion="Drawing"〉
    〈 invoke name="Drawing" 〉
      〈 source linkName="p4" 〉
      〈 target linkName="p5" 〉
    〈 / invoke 〉
  〈 invoke name="Drawing-Number" 〉
              〈 source linkName="p5" 〉
              〈 target linkName="p8" 〉
            〈 / invoke 〉
          〈 / case 〉
          〈 case condtion="Transfer"〉
          〈 invoke name="Transfer" 〉
            〈 source linkName="p4" 〉
            〈 target linkName="p6" 〉
          〈 / invoke 〉
          〈 invoke name="Transfer-in-Account" 〉
            〈 source linkName="p6" 〉
            〈 target linkName="p7" 〉
          〈 / invoke 〉
          〈 invoke name="Transfer-in-Number" 〉
            〈 source linkName="p7" 〉
            〈 target linkName="p8" 〉
          〈 / invoke 〉
        〈 / case 〉
      〈 / switch 〉
    〈 / while 〉
    〈 invoke name="End" 〉
      〈 source linkName="p8" 〉
      〈 target linkName="p9" 〉
    〈 / invoke 〉
  〈 / sequence 〉
〈/ process 〉
```

4 Web 服务系统网模型的化简分析

对于一个复杂的 Web 服务系统，其对应的 Petri 网模型的验证分析较为复杂，有的甚至难以实现。Petri 网的化简技术是一种有效途径，这方面已有诸多研究[2~4]，但大多是对 Petri 网化简过程的保持特性的研究。而基于服务序列的 Petri 网语言正好是反映 Web 服务系统的动态行为[13~19]，其语言表达式是以代数规则的形式化定义。在这一节中我们讨论 Web 服务系统的化简分析和语言表达式的生成。首先，给出如下化简规则表。

定义 1[19~21] 设 T 为有限字符集合，T^* 是 T 上的字符串集合，对 $\forall L \subseteq T^*$，则称 L 为 T 上的语言。

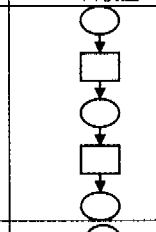
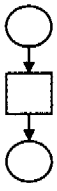
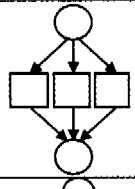
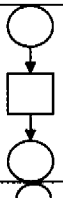
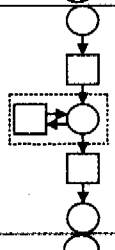
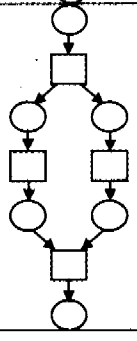
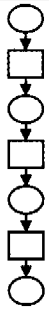
定义 2[19~21] 设 T 为有限字符集合，L_1，L_2 分别是 T 上的语言，我们定义如下的语言运算：

(1)串运算“$\circ$”，$L_1 \circ L_2 = \{\alpha_1 \circ \alpha_2 \mid \alpha_1 \in L_1 \wedge \alpha_2 \in L_2\}$；

(2)选运算“$+$”，$L_1 + L_2 = \{\alpha \mid \alpha \in L_1 \vee \alpha \in L_2\}$；

(3)并运算“$//$”，$L_1 // L_2 = \{\alpha_1 // \alpha_2 \mid \alpha_1 \in L_1 \wedge \alpha_2 \in L_2\}$，其中对于 $a, b \in T$ 和 $\alpha_1, \alpha_2 \in T^*$ 及空字符 ε，则有 $a\alpha_1 // b\alpha_2 = a(\alpha_1 // b\alpha_2) + b(a\alpha_1 // \alpha_2)$ 且 $a // \varepsilon = \varepsilon // a = a$；

表 3　Petri 网化简规则表

名称	Petri 网模型	化简的 Petri 网模型
顺序 R1		
分支 R2		
循环 R3		
并行 R4		

(4)环运算“$*$”，$L_1^* = \bigcup_{x=1}^{\infty} L_1^x$，其中

$L_1^0 = \{\varepsilon\}$，$L_1^x = L_1 \circ L_1^{x-1}$，$x \geqslant 1$。

定义 3[19~21]　设 T 为有限字符集合，T 上的语言表达式及所代表的集合递归地定义如下：

(1)ϕ 是一个语言表达式，它代表空集合；

(2)ε 是一个语言表达式，它代表集合$\{\varepsilon\}$；

(3)对于 $\forall a \in T$，a 是一个语言表达式，它代表集合$\{a\}$；

(4)如果 α_1，α_2 分别代表语言 L_1，L_2 的语言表达式，则 $\alpha_1 \circ \alpha_2$，$\alpha_1 + \alpha_2$，$\alpha_1 // \alpha_2$ 和 α_1 分别代表集合 $L_1 \circ L_2$，$L_1 + L_2$，$L_1 // L_2$ 和 L_1^* 的语言表达式。

对于 Petri 网 $\Sigma = (P, T; F, M_0)$，我们用 $RSS(\Sigma)$ 表示 Σ 的可达状态集合，Σ 的任何一个可达状态用袋[20]表示。如 $s \in RSS(\Sigma)$，$s = (p_3, 2p_5, p_6)$。

定义 4　设 Web 服务的 Petri 网模型 $\Sigma = (P, T; F, M_0)$，$RSS(\Sigma)$ 为 Σ 的可达状态集合，令

$L(\Sigma) = \{\sigma | \sigma \in T^* \wedge M_0[\sigma > M \wedge \forall M \in RSS(\Sigma)\}$，

则称 $L(\Sigma)$ 为 Web 服务的 Petri 网模型 Σ 的语言。

算法 2　Web 服务系统的 Petri 网模型语言表达式生成算法

输入：$\Sigma = (P, T; F, M_0)$

输出：$\alpha(\Sigma)$

Step1. 逐步遍历 Web 服务的 Petri 网模型 Σ，依此检查规则 R1～R4 的条件，对符合条件的，实施 Web 服务的 Petri 网模型化简，直到不能化简为止，此时化简的 Petri 网模型记作 $R(\Sigma)$；

Step2. 若 $R(\Sigma)$ 为最简 Web 服务的 Petri 网模型，则算法结束，Web 服务的 Petri 网模型 Σ 的语言表达式为最简 Petri 网模型的变迁 $\alpha(\Sigma)$；否则进行 Step3；

Step3. 利用有界 Petri 网系统的状态可达图生成算法生成化简 Petri 网模型 $R(\Sigma)$ 的状态可达图 $RMG(R(\Sigma))$；

Step4. 利用有限状态自动机的化简规则对 $RMG(R(\Sigma))$ 进行化简，最终生成 Web 服务的 Petri 网模型 Σ 的语言表达式 $\alpha(\Sigma)$。

对于例 1，我们根据上述化简规则得到 ATM 机操作过程的化简 Petri 网模型，如图 2。

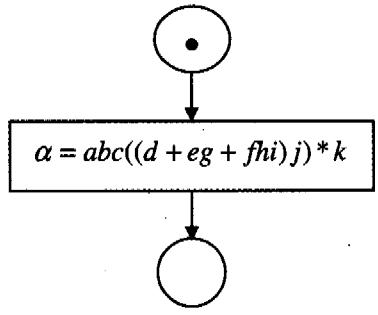


图 2　ATM 机操作的化简 Petri 网模型

其语言表达式如下：

$\alpha = abc((d + eg + fhi)j) * k$，

显然，我们有$\{p_1\}[\sigma > \{p_9\}$，$\sigma \in \| \alpha(\Sigma) \|$，$\| \alpha(\Sigma) \| = \{t \in T | t$ 为 $\alpha(\Sigma)$ 中出现的任何字符$\}$，而且 Σ 是安全的。

结论　本文研究了 Web 服务的元活动模拟和服务构件的组合集成问题，给出了相应的算法。同时，从 Petri 网化简技术出发，研究了 Petri 网化简过程的语言表达式的生成问题，提出了语言表达式的生成算法，本文结果为 Web 服务组合过程的动态特性分析和性能评价提供有力支持。进一步的研究是考虑语言表达式生成的其它有效途径，以及基于这一工具的有效分析方法。

参 考 文 献

1　Chandrasekaran S, Miller J A, Silver G S, et al. Performance Analysis and Simulation of Composite Web Services. The International Journal of Electronic Commerce and Business Media (EM), 2003, 13(2):120～132

2　Snell J. The Web services insider, Part4: Introducing the Web Services Flow Language. http://www-106. ibm. com/developerworks/Webservices/library/ws-ref4/, Jun 2001

3　Lehmann M. Web Services Composition. http://www. isys. uniklu. ac. at/ISYS/Courses/03SS/S...DKE/lehmann. ppt, 2003

4　VanderMeer D, Datta A D K, et al. FUSION: A System Allowing Dynamic Web Services Composition and Automatic Execution. In: IEEE International Conference on E-Commerce (CEC), 2003. 399～404

5　Zeng Liangzhao, Benatallah B, Lei Hui. Flexible Composition of Enterprise Web Services. Electronic Markets-The International Journal of Electronic Commerce and Business Media, 2003b, 13 (2):141～152

6　Zeng Liangzhao, Benatallah B, Ngu A H, et al. QoS-Aware Middleware for Web Services Composition. IEEE Transactions on Software Engineering, 2004, 30(5):311～327

7　Benatallah B, Sheng Quan-Zheng, Dumas M. The Self-Serv Environment for Web Services Composition. IEEE Internet Computing, 2003a, 7(1):40～48

(下转第 124 页)

执行引擎内部转化后，相当于没有策略时插入的元组：

(101，'Tim'，'1970-03-12'，CINFO_ENCRYPT('334-25-9023')，'DF425'，…)

DBA既不能直接检索到'334-25-9023'，也不能通过数据导出来获得原始信息。至于破解密钥，那是加密的问题。

限于篇幅，这里不再列举各种操作的具体实现方法。

3 基于代价的查询重写/优化

支持安全策略，数据库执行引擎不仅扩展语法(另外一种不修改语法的实现是系统将安全策略作为抽象数据类型来维护，我们在KingbaseES中就是采用类似技术支持B1级安全性)，同时还要优化策略影响的执行计划，特别在结果集元组很多的时候。

例4 根据SSN确认客户的保单数的查询Q1：

```
SELECT COUNT(*) FROM CUSTOMER
  WHERE CSSN='334-25-9023';
```

因为CSSN上有数据加密，相应的有安全策略对操作REL_SELECT和COL_SELECT进行解密。假设过滤函数是CINFO_DECRYPT，则最简单的查询执行过程是：

1)抽取所有元组形成快照；

2)策略执行CINFO_DECRYPT，如果匹配则计数。

该方法相当于对加密数据没有策略时的等价查询Q2：

```
SELECT COUNT(*) FROM CUSTOMER
  WHERE CINFO_DECRYPT(CSSN)='334-25-9023';
```

假设有N个客户记录，每页M个元组，估略代价为：

$Cost_{i/o}(N/M)+N*Cost(CINFO\_DECRYPT)$

观察Q2，很容易得到一个等价查询Q3：

```
SELECT COUNT(*) FROM CUSTOMER
  WHERE CSSN=CINFO_DECRYPT('334-25-
        9023');
```

检索函数附加代价为1！这时我们有下面的优化建议：

O1、将过滤函数应用到常量但保持查询等价；

O2、常量上的应用结果要缓存；

对于例4，如果CSSN经常被检索，可以建立索引，由于该列需要加密，索引键值也需要安全策略，这时有：

O3、索引键值直接使用执行策略后的元组列值。

这时的检索代价大略是：

$Cost_{i/o}(IdxPg+RelPg)+RelPg*p*Cost(CINFO\_DECRYPT)$

$Cost_{i/o}(IdxPg+RelPg)$是读入的索引和关系的页面I/O，而p是页面中有效元组的选择率。

此外，各种通用的数据库查询重写和优化方法仍然适用。

4 相关工作

视图是最常见的安全保护机制，通过选择输出列达到屏蔽信息的目的[6]，但视图不能影响对基本表的访问。

触发器主要作用在操纵表的DML语句，如PG[7]、SQL Server 2000[8]、MySQL 5.0[9]等。它能控制元组操作但不能预防例1的问题。类似有PG的规则系统，在执行SQL前根据规则重写查询。触发器和规则系统都可以增强以支持安全策略。SQL Server 2005[10]、Oracle 9i/10g[11,12]都提供了更强的触发器技术，监测服务器、数据库级的事件，但效率需改进。9i/10g[13]支持返回动态谓词的安全策略，但不支持在安全策略中使用任意过滤函数，包括数据加密。

结论和未来工作 数据库受到来自内外的安全威胁，DBA也不可靠。安全策略面向应用，对任何数据库用户都有效。本文的安全策略模型有6个基本原则：4个强制性、1个可配置和1个推荐原则，它们或控制操作，或加工数据。

完全支持安全策略基本规则的系统不要求应用修改原来的SQL语句。按照建议模型，查询引擎能够自动根据策略定义对各种SQL语句进行基于代价的重写和优化。

我们提出的安全策略模型只是一种防卫手段，随着面向服务和自定义能力的增强，数据库系统会受到更多安全危险。我们的策略模型要继续完善，还要考虑意外补救措施和多类管理员权限划分等。

参考文献

1 CCIMB-99-031& 032 & 033. Common Criteria for Information Technology Security Evaluation, Version 2.1, August 1999

2 GB/T 18336-2001. 信息技术/安全技术/信息技术安全性评估准则. 2001

3 Department Of Defense. Trusted Database Management System Of TCSEC, 1991

4 Elliott B D, LaPadula Leonard J. Secure computer systems: unified exposition and MULTICS interpretation, 1976

5 张孝. 可信COBASE的系统强制存取控制的设计与实现:[中国人民大学硕士学位论文]. 1998

6 萨师煊，王珊. 数据库系统概论(第三版). 高教出版社，2000

7 PostgreSQL8.2. http://developer.postgresql.org/docs/postgres/

8 Microsoft Corp. SQL Server 2000, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/tsqlref/ts_create2_7eeq.asp

9 MySQL 5.0. http://dev.mysql.com/doc/refman/5.0/en

10 Microsoft Corp. SQL Server 2005. http://msdn2.microsoft.com/en-us/library/ms189799.aspx

11 Oracle Corp. Oracle9i Application Developer's Guide, 2002

12 Oracle Corp. Database SQL Reference 10g, 2003

13 Oracle Corp. Label Security Administrator's Guide 10g, 2003

(上接第103页)

8 Han Yan-Bo, et al. CAFISE: An Approach to Enabling Adaptive Cofiguration of Service Grid Applications. Journal of Computer Science & Technology, 2003, 18(4):484～494

9 张尧学，方存好. 主动服务——概念、结构与实现. 北京：科学出版社，2005

10 李建强，范玉顺. 基于Petri网模型化简的工作流模型验证. 信息与控制，2001, 30(6):492～497

11 孙瑞志，史美林. 一种基于Petri网化简的工作流过程语义验证方法. 软件学报，2005, 16(7):1242～1251

12 许安国，蒋昌俊. P/T网的化简运算及其性质研究. 软件学报，1997,8(7): 493～504

13 Jiang C J. A Polynomial-time Algorithm for the Legal Firing Sequences Problem of A Type of Synchronous Composition Petri Nets. Science in China (Series E), 2001,44(3):226～233

14 Jiang C J. Testing of Functions of Complex Systems Based on Synchronous Composition Nets. Int J of Studied on Information Control, 2000, 9(4):321～328

15 Jiang C J, et al. Behaviour Relations in Synthesis Process of Petri Net Models. IEEE Trans on RA, 2000, 16(4):400～407

16 Jiang C J, et al. Urban Traffic Information Service Application Grid. J of Comp Sci & Tech, 2005, 20(1):134～140

17 Jiang C J, Wang H Q, Liao S Y. Behavior Relativity of Petri Nets. J of Comp Sci & Tech, 2002,17(6):770～780

18 Du Y Y, Jiang C J. Formal Representation and Analysis of Batch Stocks Trading Systems Through Logical Petri Net Workflows. In:Proc. of 4th Int Conf. on Formal Engineering Methods, Lecture Notes in Computer Science, No. 2495, 2002. 221～225

19 蒋昌俊. 离散并发系统的PN机理论. 北京：科学出版社，2000

20 Peterson L L. Petri网理论与系统模拟. 吴哲辉译. 中国矿业大学出版社，1981

21 Hopcroft J E, et al. 自动机理论、语言和计算. 徐美瑞译. 北京：科学出版社，1990