一种可生存的数据库安全结构设计与实现*)

朱建明[1] 郭渊博[1] 马建峰[1,2]

(西安电子科技大学 计算机网络与信息安全教育部重点实验室 西安710071)[1]

(天津工业大学计算机与自动化学院 天津300160)[2]

摘 要 提出了一种基于容忍入侵的可生存的数据库安全结构。这种结构将冗余和多样性技术相结合,采用秘密共享方案,实现数据库系统的可生存性,以及关键数据的机密性。与其它容忍入侵的数据库系统相比,本文所提出的数据库安全结构,采用系统整体安全策略,综合多种安全措施,实现了系统关键功能的安全性和健壮性。

关键词 数据库,可生存性,容忍入侵,入侵检测

Design and Implement of Architecture for Survivable Database Systems

ZHU Jian-Ming[1] GUO Yuan-Bo[1] MA Jian-Feng[1,2]

(Key Laboratory of Computer Network and Information Security, Ministry of Education, Xidian University, Xi'an 710071)[1]

(School of Computer and Automatization, Tianjin 300160)[2]

Abstract This paper proposes survivable architecture for database system security based on intrusion-tolerant mechanisms. We utilize the techniques of both redundancy and diversity and threshold secret share schemes to implement the survivability of databases and to protect confidential data from compromised servers in the presence of intrusions. Comparing with the existing schemes, our approach has realized the security and robustness for the key functions of a database system by using the integration security strategy and multiple security measures.

Keywords Database, Survivability, Intrusion-tolerant, Intrusion-detection

1 引言

随着电子商务和电子政务的不断发展,Internet在线服务迅速增加,数据库系统的安全越来越重要。数据库安全研究的重点在于如何保护数据的机密性(confidentiality)、完整性(integrity)和可用性(availability)。所采用的技术主要有授权、访问控制、多级安全数据库、多级安全事务处理、加密和入侵检测等。这些安全技术的主要作用是防御攻击或入侵。然而事实上,正如文[1,2]中所指出的那样,在有些情况下,这些防御措施对于一些恶意攻击可能无效。因此,在防御失败的情况下,如何保障数据库系统的可生存性(survivability)就成为一个值得关注的重要问题。另一方面,即使入侵能够被检测出来,系统管理员仍要面临两大难题:一是如何确定入侵所引起的破坏,二是如何将系统恢复到安全状态。由于入侵者一般都会在入侵后修改系统日志文件,擦去入侵的痕迹,使得确定入侵所引起破坏的位置更为困难。而系统恢复需要有干净的备份,还要重新初始化系统、从备份中恢复信息等操作。此外,诊断的困难性也增加了恢复的困难性,而恢复本身通常需要较长的时间才能完成,这无疑也会降低原来系统的可用性,甚至可能引起安全备份与入侵发生期间所建立数据的不一致。

近年来,数据库系统的可生存性研究引起人们的关注[5,6]。在文[6]中,基于入侵检测,针对恶意事务处理,提出了四种容忍入侵数据库的体系结构,其研究的重点是确定破坏的位置、修复的方法、隔离攻击的措施以及数据恢复的机制。在文[5]中,基于秘密共享技术给出了一种可生存的存储系统。但是这些方法都仅仅是从数据库系统的一个方面来提高其可生存性的,没有从系统整体结构上综合考虑数据库系统的可生存性,而且所采用的方法都依赖于系统入侵检测的性能。本文基于容忍入侵的思想,将冗余和多样性技术相结合,同时使用门限密码技术,提出了一种可生存的数据库安全结构。

2 可生存的数据库安全结构

与入侵检测不同,容忍入侵主要考虑在入侵存在的情况下系统的生存能力,容忍入侵的系统具有

*)基金项目:国家自然科学基金重大计划(No. 90204012)、国家社会科学基金项目(NO. 03BJY089)、教育部优秀青年骨干教师资助计划、教育部科学技术重点研究项目。朱建明 博士生,主要研究方向为信息安全、电子商务;郭渊博 博士生,主要研究方向为信息安全、容忍入侵;马建峰 教授,博士生导师,特聘教授。

自诊断、修复和重构的能力。基于容忍入侵的思想，采用纵深防御策略，将冗余和多样性技术相结合，并利用门限秘密共享方案，我们提出了一种可生存的数据库系统安全结构，如图1所示。在该结构中，实现可生存性的主要策略是：在系统构件中引入一定的冗余度；不同的应用服务器运行于不同的操作系统环境中，应用程序采用多版本程序设计(N-version programming)；利用门限密码技术将信息分布于多个系统构件上，各个构件通过一定的通讯机制建立联系，以实现数据库系统的可生存性和机密数据的安全性；综合多种安全措施，在容忍入侵的基础上，提高入侵检测系统的性能和学习功能；采用多阶段数据库恢复技术，及时评估数据破坏，恢复系统状态。

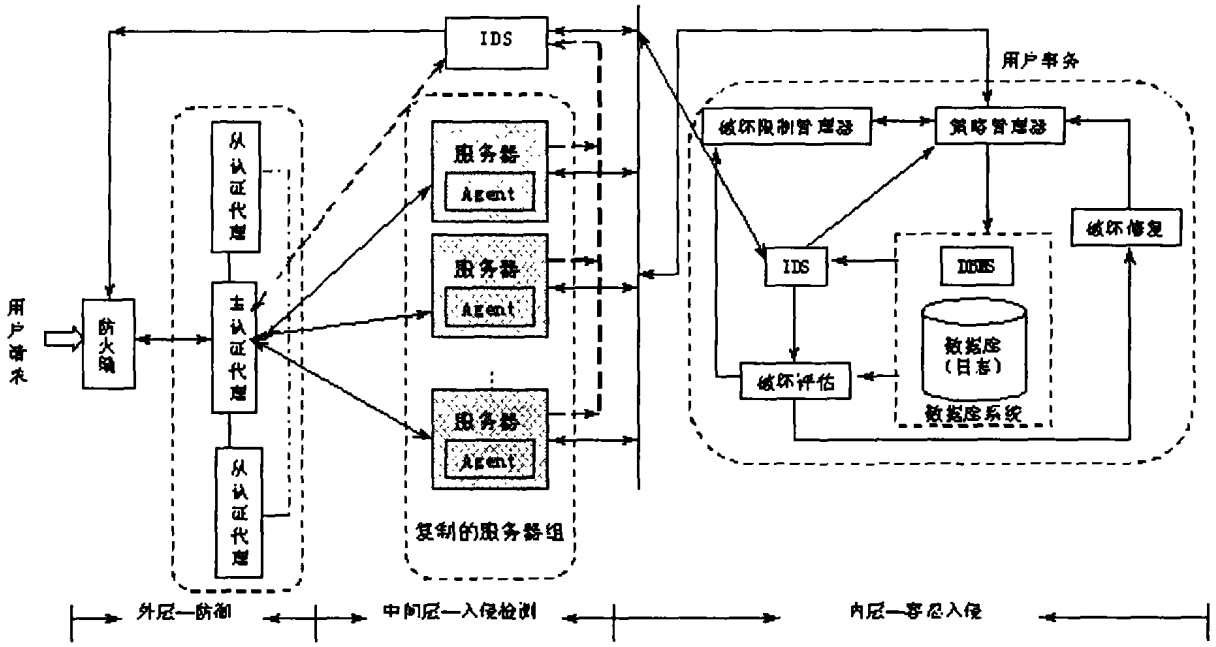

图1 自适应的容忍入侵的数据库安全体系结构

系统主要构件：防火墙、主/从认证代理、服务器组、入侵检测系统(IDS)、冗余的数据库存储节点等。

系统的运行过程：当用户对数据库进行访问时，运行过程如下：

(1)首先通过防火墙过滤；

(2)进行认证：在认证代理中有一个主认证代理，其他称为从认证代理。主认证代理负责过滤和净化客户要求，并将有效的客户请求传给服务器。服务器处理客户请求并将结果返回到主认证代理，主认证代理经过大数判决后，将结果提交给客户。当主认证代理出现故障时，其中一个从认证代理将成为新的主认证代理继续工作[3]。

(3)服务器组由在功能上具有一定冗余的多个COTS(Commercial-of-the-shelf)服务器构成，其主要功能是为客户提供应用服务。

(4)IDS 采用 Multi-agent 技术[4]。IDS 由多个 agent 组成，即在每个服务器上分布一个或多个 agent，负责监视服务器的运行状态以及其中关键数据的机密性和完整性。IDS 将 agent 从所在主机接收到的信息、认证代理处反馈的信息和存储节点处收集的信息进行分析汇总，进行入侵检测。特别是数据库系统中的 IDS，将基于数据库的存储特征，如数据和属性的修改、修改模式、内容完整性变化、对可疑内容的操作等，进一步提高入侵检测的性能。

IDS 根据所保存的历史日志来确定恶意事务，破坏评估确定恶意事务所引起的破坏，破坏修复通过 UNDO 事务来修复所确定的破坏，破坏限制管理器限制对已经由破坏评估所确定的破坏的数据项的访问，而不限制已经清除的数据项，策略管理器一方面作为正常用户事务和 UNDO 事务的代理，另一方面负责系统容忍入侵的策略。

(5)可生存的数据库存储结构。数据库数据根据其安全级别可以分为一般数据 G 和机密数据 S，每个存储节点存储着一般数据的全备份和机密数据的一个份额。采用(t,n)门限秘密共享方案，将数据库中的机密数据 S 分成 n 份：$S_1, S_2, \cdots, S_n$(n 是存储节点的个数)，分别存储于 n 个存储节点中。只有 t 个以上的机密数据份额才能重构数据，攻击者只有入侵 t 个以上的存储节点才有可能重构机密数据 S，从而使数据库数据具有更高的安全性。

3 系统各功能模块的设计

在所提出的结构中，综合使用了容忍入侵、入侵检测、破坏评估和状态恢复等多种手段实现系统的可生存性。

3.1 容忍入侵

在受到入侵时，系统继续提供基本服务的能力是通过系统容忍入侵的性能来体现的。这种思想贯穿于我们系统设计的各个方面：

(1)防火墙和认证代理：在图1的结构中，防火墙和认证代理对系统的访问控制起着关键的作用，因此也成为外部攻击的重点目标之一。在我们的设计中，为了增强认证代理的抗攻击能力，将主认证代理作为一个动态的虚拟服务器，不固定某一个代理为主认证代理。对外而言，主认证代理只表示认证代理中的一个 IP 地址。每个代理都设定一个动态的优先级，主认证代理定期(间隔几秒，称为广播间隔)向从认证代理发送信息。同时，从认证代理也要对主认证代理发出的信息即时做出回应，如果在一定时间内主认证代理接收不到某个从认证代理的应答，则应发出报警信息，显示该从认证代理出现故障。所有从认证代理都能收到主认证代理所发送的信息，如果在一定时间内没有收到主认证代理的信息，则优先级高的从认证代理成为新的主认证代理。

(2)服务器组：服务器组的容忍入侵特征通过以下两个方面来体现：

·不同的服务器运行于不同的操作系统平台，所运行的应用软件采用多版本程序设计技术，以防止一种攻击造成对所有服务器的破坏。

·通过主认证代理对服务器的认证，及时发现有问题的服务器。此外，还可以通过代理 IDS 发现可疑服务器，并暂时中断与它的联系，进行故障处理。

(3)安全存储结构：将数据库复制、分散存储于多个存储节点上，通过存储节点集的安全性和连续性实现系统的可生存性。因此，这里的可生存性并不依赖于某个特殊节点的安全性。机密数据的存储采用(t,n)门限密码方案，将其份额分别存储在 n 个存储节点上。数据存储节点独立于主机运行，不仅负责存储数据，而且还要保护数据。

3.2 入侵检测

IDS 的性能直接影响到整个系统的性能。当前，IDS 在性能上存在高误报率、漏检以及时间延迟等问题，而这些问题的解决在很大程度上依赖于所收集数据的及时性、完整性和准确性。如果收集到的数据有延迟，入侵检测就会因为太迟而没有意义；如果获取的数据不完整，入侵检测的性能就会降低而出现漏检；如果数据不正确，IDS 可能会产生误报。因此，在 IDS 中建立及时、完整和准确的数据收集机制非常重要。在我们的方案中，采用基于 Agent 的 IDS，通过分布在系统中的 Agent 及时收集入侵检测的数据[7]，提高 IDS 的性能。

基于多代理的入侵检测数据收集机制如图2所示。其中，元代理 MA 负责管理信息安全策略，并对每个 agent 的行为进行调整和协作，以确保满足系统整体安全的要求。后台处理代理 AD 负责监视和预处理通信数据。学习代理 LA 负责适应网络配置和新的攻击类型。密码代理 CSA 负责网络节点之间的通信渠道的安全。控制破坏代理 DCA 负责评估入侵对系统造成的破坏和信息恢复。鉴别和认证代理 IAA 负责对于信息源的鉴别和真实性的确认。访问控制代理 ACA 一方面负责对机密信息进行保护，不允许敏感信息通过非授权的访问渠道流出；另一方面按照严格的访问控制策略为合法用户提供对信息资源的访问。其中代理 DCA 与数据库的破坏评估和状态恢复相联系。

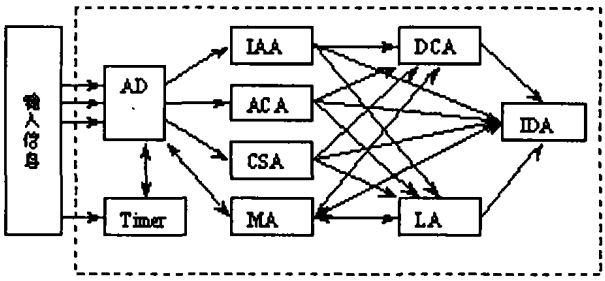

图2 基于多代理的入侵检测数据收集机制

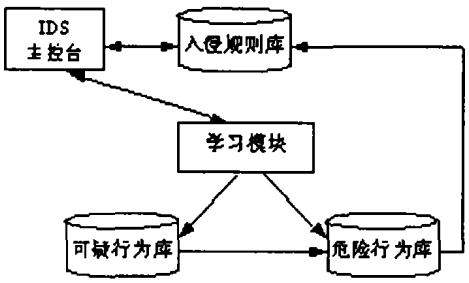

图3 IDS 示意图

通过这样的多代理数据收集机制，一方面丰富了入侵检测系统的数据资源，另一方面将系统各个环节的安全业务联系起来，使得能够从系统整体的角度去判断入侵事件，提高入侵检测的性能。

另一方面，由于系统具有容忍入侵的能力，使得 IDS 能够具有不断学习和适应的能力。根据收集的数据，不断充实可疑行为库、危险行为库，最终加入入侵规则库(如图3所示)，使 IDS 的检测性能不断提高。

此外，agent 收集基于数据库的操作特征，如数据和属性的修改、修改模式、内容完整性、对可疑内容的操作等[8]，能够更好地捕获入侵的特征，进一步提高入侵检测的性能。

3.3 数据库破坏评估与状态恢复

入侵者一旦获得访问系统的权限，可能有几种破坏的方法。大多数通过修改系统日志文件来破坏入侵的证据，也有的入侵者在系统中安装后门程序，

以便将来再次入侵，也有可能安装其他软件，读取或修改敏感文件，或者把系统作为一个平台进行攻击。由于入侵与检测之间存在延迟，在这期间，入侵者可能继续进行其恶意活动，同时其他用户也在使用系统，这样使合法的修改与入侵者的恶意操作交织在一起。这样一旦入侵被检测或停止，系统管理员将面临诊断和恢复两大难题。

我们将数据库看成是数据对象的集合，数据库的状态由这些对象的值所决定。事务是读与写操作的序列，其结果要么提交，要么中止。由于中止操作对数据没有什么影响，因此这里只考虑提交的情况。一个事务的执行将数据库从一个状态转换为另一个状态。当两个交易同时对一个对象进行操作，而且其中一个是写操作时，这两个事务发生冲突。一组事务的执行称为历史(history)。

按照数据库的原子特性，只有提交的事务才能真正改变数据库数据。那么如果在提交前能把恶意事务检测出来，通过回卷(roll back)就可以恢复到引起破坏之前的状态。这在理论上是可行的，但在实际中是不现实的。因为通常交易的执行要比检测快得多，而将交易执行放慢又会降低系统的可用性。另一方面，一些内部的恶意用户很难检测出来，只有通过事务的语义才有可能确定这种入侵。因此，可行的方法是：在发现数据库被破坏后，尽可能确定破坏的位置并修复，以便使数据库能够继续使用。

我们通过捕获事务的关系来跟踪并限制破坏的扩散[1]。为此给出下面的定义，其中 T_i、T_j、T_u、T_v 表示事务。

定义1(事务依赖关系) 如果存在一个对象 x，使得 T_i 在 T_j 修改 x 后读 x，而且在 T_j 修改 x 与 T_i 读 x 之间不存在其他事务修改 x，则称 T_i 依赖于 T_j，记为(T_i, T_j)。

事务依赖关系能够表示恶意事务破坏传播的路径。

定义2(破坏传播) 如果修改对象 x 的事务 T_j 依赖于一个修改对象 y 的一个恶意事务，则对 y 的破坏传播给了 x，称 x 被破坏了。如果修改对象 x 的事务对已破坏的对象 y 进行读操作，也称对 y 的破坏传播给了 x。

定义3(影响) 如果(T_v, T_u)属于事务依赖关系的传递闭包，则称事务 T_u 影响事务 T_v。

因此，如果一个恶意事务 B_i 影响了一个无辜事务 G_j，那么 B_i 的写集的破坏将传播到 G_j 的写集，G_j 的每一个写集都将被破坏。

定义4(依赖图) 定义历史中的事务集 S 的依赖图为：$DG(S)=(V,E)$，其中 V 是 S 与被 S 影响的事务集合的并集，在 E 中存在一条边，$T_i \rightarrow T_j$，如果 $T_i \in V, T_j \in (V-S)$，并且 T_j 依赖于 T_i。

这样，除了 S 外，DG(S)包含而且只包含了由事务 S 所影响的事务，通过计算 DG(S)就能够评估由事务 S 所引起的破坏。

当一个恶意事务被检测出来时，初始限制阶段，会估计一个可能受到破坏的对象集 S_E，经过一系列的破坏评估，得到准确的破坏集 S_D，S_E 与 S_D 之间的关系有四种：

(1)$S_E=S_D$，表示准确估计

(2)$S_E \supset S_D$，表示过度估计

(3)$S_E \subset S_D$，表示估计不足

(4)$S_E \cap S_D \neq S_E$ 且 $S_E \cap S_D \neq S_D$，表示近似估计

准确估计是我们所要求达到的目标，但在初始阶段实际上不太可能。在多数情况下，过度估计是合理的评估策略，因为它能够保证破坏不再扩散。而估计不足只能限制部分破坏，从而造成破坏扩散。近似估计则可能限制了没有破坏的数据，而将破坏的数据没有限制。

因此，在我们的评估策略中，采用过度估计的策略，实施多阶段限制过程，使限制的范围逐步达到准确。用 $S_E, S_2, S_3, \cdots, S_n, S_D$ 表示多阶段限制过程中评估得到的破坏集，那么有下列结论成立：

定理1 在对数据库进行多阶段破坏评估中，采用过度估计策略，所得到的破坏集序列 $S_E, S_2, S_3, \cdots, S_n$ 收敛于 S_D。其中 S_E 表示初始限制阶段得到的破坏集，S_i 表示第 i 个阶段得到的破坏集，$2 \leqslant i \leqslant n$，而且 $S_E \supset S_i$，当 $i<j$ 时 $S_i \supset S_j$，$S_n \supset S_D$。

状态恢复就是沿着被修改对象的踪迹恢复数据库数据。从状态转移的角度看，恢复状态就是恢复数据库到好的状态，即没有被恶意事务影响的状态。在破坏确定以后，通过回卷可以使数据库恢复到正常状态[9]。

实现算法：

设定一个恶意事务集 B 和一个记录所有读写操作的日志 DatabaseLog。通过扫描日志，确定每一个被 B 影响的纯净事务(没有受到恶意事务影响的事务)。当扫描到一个恶意事务时，将该事务写过的所有数据标识为 dirty；当扫描一个纯净事务时，如果该事务读取过 dirty 项，那么该事务写过的数据也将被标识为 dirty，同时合成并运行一个特定的 UNDO 事务，以消除该 dirty 项的影响。当通过 UNDO 事务对数据项进行了清理之后，该数据项就不再被标记为 dirty。UNDO 事务的合成过程如下：对于恶意事务更新过的每一个数据项，如果该项之前没有被任何 UNDO 事务清理过，则在 UNDO 事务中增加一个写操作，以便将该数据项恢复为恶意事务更新之前的值。

由于被破坏的数据项在被标识并清除之前，算法允许用户继续运行新的事务。如果新的事务读取

了已被破坏，但尚未被标识的数据项，则该事务可能会传播这种破坏。如果破坏传播的速度比修复的速度快，那么修复过程可能永远不会终止；如果修复的速度更快，则该进程才会终止。事实上，只有在满足以下条件时，我们才能够确保修复进程终止：(1)每个恶意事务都被修复；(2)没有数据项被标记为dirty；(3)进一步扫描将不再标识出任何新的破坏。

结束语 本文所提出的可生存的数据库安全结构，将冗余和多样性技术相结合，采用门限密码方案，实现数据库系统关键信息的完整性和有效性，以及机密数据的保密性。与授权、推理控制、多层安全数据库和多层安全事务处理等传统的防御型安全措施不同，可生存的数据库不仅考虑了对入侵与攻击的防范与检测，而且在入侵存在的情况下系统具有一定的可生存性和抗毁能力。与现有的数据库安全模型相比，我们所提出的方案采用系统整体安全策略，综合了多种安全措施，实现了系统关键功能的安全性和健壮性，满足了数据库系统的可生存性要求，对于电子商务和电子政务中数据库系统的设计和实现具有重要的参考意义。

参 考 文 献

1 Liu P, Jajodia S. Multi-phase damage con-nement in database systems for intrusion tolerance[A]. In: Proc. 14th IEEE Computer Security Foundations Workshop[C], 2001. 191～205

2 Ammann P, Jajodia S, McCollum C D, Blaustein B T. Surviving information warfare attacks on database[A]. In: proc. of the IEEE Symposium on Security and Privacy[C], Oakland, CA, May 1997. 164～174

3 朱建明，马建峰. 基于容忍入侵的数据库安全体系结构[J]. 西安电子科技大学学报，2003，1

4 Spafford E H, Zamboni D. Intrusion detection using autonomous Agents[J]. Computer Networks, 2000, 34(4): 547～570

5 Ranger G R, Khosla P K, Bakkaloglu M, Bigrigg M W, Goodson G R, Oguz S, Pandurangan V, Soules C A N, Strunk J D, Wylie J J. Survivable storage systems[A]. In DARPA Information Survivability Conference and Exposition II[C]. IEEE Computer Society, June 2001. 184～195

6 Liu P. Architecture for Intrusion Tolerant Database systems [A]. In: Proc. 18th Annual Computer Security Applications Con. [C], Dec. 2002

7 Gorodetski V, Kotenko I, Skormin V. Integrated Multi-Agent Approach to Network Security Assurance: Models of Agents' Community [A]. Information Security for Global Information Infrastructures, IFIP TC11 Sixteenth Annual Working Conference on Information Security [C], Qing, S., Eloff J. H. P, Beijing, China 2000. 291～300

8 Pennington A G, Strunk J D, Griffin J L, et al. Storage-based Intrusion Detection: Watching storage activity for suspicious behavior [A]. In: Proc. of the 12th USENIX Security Symposium Washington [C], DC. Aug. 2003

9 Liu P, Luenam P. ODAM: An on-the-fly samage assessment and repair system for commercial database applications [A]. In: Proc. 15th IFIP WG 11.3 Working Conf. on Database and application Security [C]

(上接第142页)

算出来的一级评价指标矩阵 A，则模糊评价结果矩阵

$$B=V\times A=(v_1\ v_3\ v_3\cdots\ v_k)\times\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1k} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2k} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3k} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{j1} & a_{j2} & a_{j3} & a_{j4} & a_{jk} \end{pmatrix}$$

最后可以得到模糊评价结果矩阵 $B=(b_1,b_2,\cdots b_k)$，然后利用最大隶属度原则确定 Web 信息系统最终评价结果，即 $b=\max(b_1,b_2,\cdots,b_k)$。

6 评价模型运行效果

运用建立的 Web 信息系统软件质量模糊评估模型对我院电子商务模拟 Web 软件系统进行质量评估。经过学生、老师以及专家对各指标进行评分，整理后计算得到一级指标评估矩阵 A 为：

$$A=\begin{pmatrix} 0.7 & 0.3 & 0 & 0 & 0 \\ 0.35 & 0.65 & 0 & 0 & 0 \\ 0.4 & 0.3 & 0.2 & 0.1 & 0 \\ 0.2 & 0.3 & 0.4 & 0.1 & 0 \\ 0.55 & 0.45 & 0 & 0 & 0 \end{pmatrix}$$

$$B=V\times A=(0.3\ 0.2\ 0.2\ 0.15\ 0.15)\times\begin{pmatrix} 0.7 & 0.3 & 0 & 0 & 0 \\ 0.35 & 0.65 & 0 & 0 & 0 \\ 0.4 & 0.3 & 0.2 & 0.1 & 0 \\ 0.2 & 0.3 & 0.4 & 0.1 & 0 \\ 0.55 & 0.45 & 0 & 0 & 0 \end{pmatrix}$$

经过计算得到 B=(0.4725 0.3925 0.10 0.035 0)，利用最大隶属度原则确定该 Web 软件质量最终评估结果为“优秀”。

结束语 在本文建立的模糊评价模型中，随着计算机 Web 技术不断发展，对于模型中知识库的更新和不断完善有待进一步加强，今后考虑将每个专家的评价不断添加到知识库中，不断更新知识库，从而增强知识库的经验值，有利于评价结果更加准确和合理。

参 考 文 献

1 陈明. 软件工程学[M]. 北京：科学出版社，2002

2 刘普寅，吴孟达. 模糊理论及其应用[M]. 长沙：国防科技大学出版社，2000

3 李良宝，韩喜双. 软件质量的多级模糊综合评价[J]. 哈尔滨工业大学学报，2003，35(7)：812～814

4 Jensen A L. Building a Web-based information system for variety selection in field crops----objectives and results. Computer and Electronics in Agriculture, 2001, 32: 195～211

5 王胜芝，鲜明，等. 软件质量综合评价方法研究[J]. 计算机工程与设计，2002，23(4)：16～18

6 Zadeh L A. Fuzzy Sets and systems. Information and Control, 8 (3): 29～37

7 陈建明，王海峰. 软件质量模型及其评价. 微电子学与计算机[J]. 2003，5：64～66

8 冯建湘，唐嵘，高利. 基于模糊逻辑的软件质量评价方法. 安徽理工大学学报，2003，22(4)：40～42