

软件风险管理及优化控制

徐如志, 钱乐秋, 龚洪泉, 李维宏

(复旦大学计算机系软件工程实验室, 上海 200433)

摘 要: 描述并分析了基于 CMM 的软件风险管理特点, 给出一个软件风险优化控制模型及优化控制策略, 并设计一个动态规划的软件风险优化控制算法。最后给出使用上述方法解决问题的一个例子。该文的研究为有效地管理和定量地控制软件风险, 提高软件项目的成功率提供了有力支持, 因而可广泛应用于基于 CMM 的软件项目风险管理。

关键词: 软件过程; CMM; 风险; 风险管理

Software Risk Management and Control Optimization

XU Ruzhi, QIAN Leqiu, GONG Hongquan, LI Weihong

(Software Engineering Laboratory of Computer Dept., Fudan University, Shanghai 200433)

【Abstract】 This paper describes and analyzes the characteristics of CMM-based software risk management, and a software risk control optimization model is presented. Moreover, a software risk control strategy and a dynamic programming algorithm are put forward. Finally an example of using above method to solve a discrete problem is given. This research provides strong supports for effectively managing and quantitatively controlling software risk, and greatly promoting the possibility of the success of software projects. Therefore, it can be widely used in the application for CMM-based software project risk management.

【Key words】 Software process; Capability maturity model(CMM); Risk; Risk management

与一般项目相比, 软件项目具有更大的不确定性^[2], 在项目的实施中, 各任务的执行都存在与计划偏离的风险, 这些风险可以在软件项目各小组之间进行传递和累积, 并最终影响整个项目的风险水平。研究^[3]表明, 约有 1/3 的软件项目, 由于未能及时预见和有效控制软件风险, 导致软件过程失控而被迫取消; 另外 2/3 的软件项目, 由于同样的原因, 实际开发时间平均是计划的两倍。因此风险管理被认为是管理软件项目特别是管理大型软件项目的最佳实践中最重要的方面^[4]。

随着 CMM(Capability Maturity Model)在软件组织的推广应用, 软件过程能力的不断提高, 越来越多的软件过程数据及风险管理数据被收集和存放在过程数据库中^[5], 这些以往软件项目的历史过程数据和知识为评估和优化控制当前的软件项目过程奠定了基础。

1 相关研究

风险管理包括风险评估和风险控制, 目前软件风险管理主要基于 3 个基本的方法框架: (1)基于 COCOMO 模型的风险评估^[6]; (2)SEI 的基于 CMM 的持续软件风险管理^[7]; (3)根据实践经验而来的 Top10 风险分析控制方法^[8]。前一种方法着重于对软件风险要素的评估, 后一种方法则完全基于管理者的经验。这两种方法都是依赖于管理者的主观判断, 缺乏客观的数据支持。当前基于 CMM 的软件风险管理被越来越多的软件企业应用。其突出特点是风险管理与软件过程工程、过程管理整合在同一个框架。在 CMM2 的项目计划 KPA(Key Practice Area)、项目跟踪与控制 KPA, 以及在 CMM3 的集成软件管理 KPA 中, 对软件风险管理的目标、能力要求和活动作了详细约定。

在 SEI 后来推出的 CMMI(Capability Maturity Model

Integration)^[9], 风险管理作为一个独立的 PA(Process Area)得到进一步强化, 强调风险管理的持续性, 强调对风险管理过程数据的收集和利用, 而且过程数据库(含风险数据库)在风险管理中扮演重要角色^[5,9]。在风险评估方面, Elaine M^[10]研究并设计了用于识别和分析软件风险的提问单、风险列表。Alfred B 等人对基于过程的风险评估^[11]也作了深入研究。这些规定和研究成果显著地增强了软件风险预测的能力, 但风险管理是预测和决策相结合的管理技术, 有好的风险预测方法但没有好的风险控制决策支持工具, 仍然不能很好地解决风险管理的问题, 因此软件风险控制的定量支持工具成为当前软件风险管理的一个亟待解决的问题。

2 CMM 与软件风险管理

2.1 软件风险与软件风险管理过程

软件风险是指软件开发过程中存在大量的需求、技术、人员、过程、组织等方面的不确定性, 可能导致软件产品/服务的功能不能满足要求、费用超出预算、进度延迟或项目被迫取消等所不期望的后果。软件风险管理则是对可能导致上述不利影响的风险因素进行评估和控制, 将风险降至管理者可以接受的范围内^[7,10]。

软件风险评估包括风险识别和风险分析, 主要是识别并记录可能对项目造成不利影响的因素, 评估每一个已识别风险发生的概率与后果。在此基础上按风险影响的大小确定风

基金项目: 上海市科学技术委员会资助研究项目(035115026)

作者简介: 徐如志 (1966 -), 男, 博士、高工, 研究方向: 软件复用, 软件过程与项目管理; 钱乐秋, 教授、博导; 龚洪泉, 博士生; 李维宏, 博士生、讲师

定稿日期: 2004-03-16

E-mail: rzxu@fudan.edu.cn

险的优先级,最终形成一个软件风险列表作为软件风险控制的基础。软件风险控制包括风险计划、风险跟踪和风险处理,主要是制定风险控制的目标、策略、方法以及应对每个重要风险的方案,然后根据风险计划跟踪已识别风险的变化情况,及时调整风险应对计划或采取必要的风险处理措施,将风险控制在管理者可以接受的范围内。风险管理过程如图1所示。

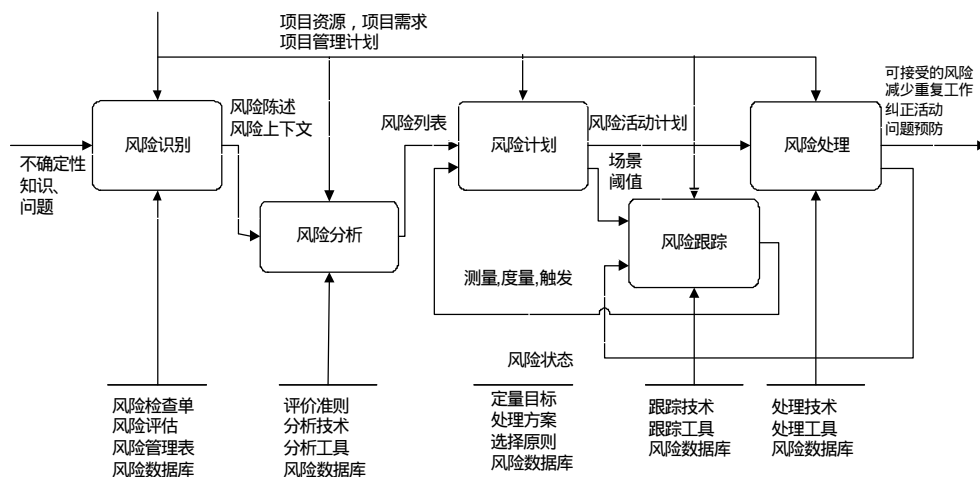


图1 软件风险管理过程

尽管引起软件项目风险的原因多种多样(如技术、过程、管理等因素),但对于一个软件项目,它的过程状态主要体现在与项目进度、成本(资源投入)和质量3个基本要素相关的参数。相应地,软件项目的风险也最终表现为与项目的进度、成本和质量相关的风险。本文重点研究在满足质量要求的前提下,如何通过合理地在项目执行的各个任务之间调配给定的资源投入,最大限度地降低整个项目的进度(工期)风险,以保证项目在合理的成本范围内按期交付。

2.2 基于CMM的软件风险管理特点

CMM^[11]将软件过程看作是一个可度量、可控制、不断改进的过程,它提供了一个框架,将软件过程改进的进化步骤组织成5个成熟等级,它的体系既包括软件工程过程本身,也包括对这一过程的管理。显然,软件过程的成熟度越高,过程定义的越清晰,过程越稳定,软件风险的可控性越强。

(1)CMM 将软件工程、过程管理及风险管理整合在一个框架。CMM3 的集成软件管理 KPA 要求项目组依据组织的标准软件过程经过适当裁减后形成项目定义的软件过程,根据项目定义的软件过程对项目进行计划和管理,并在具体的管理活动中,对软件风险识别、记录、评估和控制进行详细约定。一方面这些约定为实施软件风险管理奠定了规约基础,另一方面,软件过程在项目计划阶段事先定义,明确了软件风险传递的过程。

(2)基于CMM 的软件风险控制建立在软件过程跟踪与度量基础上。CMM2 要求在每个里程碑处测量项目关注的主要参数,CMM3 则要求获得更为详细的度量数据,测量项的粒度更细,软件过程的可视性更高,随着CMM 级别的提高,管理者对风险的识别、分析及控制能力不断提升,风险控制将具有更可靠的度量数据基础。

(3)基于CMM 的软件风险控制,可以复用组织的过程数据库中以往类似项目的知识、经验和数据。实施CMM2 的组织要求测量、记录和保存重要的项目过程信息,CMM3 至CMM 5 则要求所有项目的过程数据包括项目的风险评估及控制数据,作为组织的资产存放在过程数据库中(或单独的风险库中),并在整个组织内共享。因此在制

定风险控制计划时,可参考过程数据库中类似项目风险管理的历史数据和经验^[5,9]。

3 软件风险优化控制

3.1 优化控制模型

对于一个软件项目,从系统的分析、设计、编码实现、测试及交付的各个任务之间的关系都可以归结为串行、并行以及串行并行的混合共3种情况。利用文献[12] 给出的软件风险的传递算法,在明确了各个任务的风险是可以计算整个项目的风险。式(1)为n个串行任务的项目总风险。其中 r_i 为项目中第i个任务的风险,R为整个项目的风险。

一旦发现项目的风险超过可以接受的范围时,为了把项目的风险降至可以接受的范围内,管理者通常会首先采取在项目组内调整资源分配,或采取启用项目风险预备金的方式将项目的风险降至可以接受的范围内,但相同数量的资源投入在各个任务之间的不同分配方案会导致不同的软件项目风险。下面给出在资源投入总额确定的前提下对软件风险进行优化控制的模型。

$$R = 1 - \prod_{i=1}^n (1 - r_i) \quad (1)$$

通常,一个任务的风险与资源投入之间具有一定的负相关关系,即较多的投入意味着较小任务风险,即对某一任务增加投入或减少资源投入会导致该任务风险的增加或降低。假定任务k(1≤k≤n)在原计划基础上增加投入 u_k (u_k 为负值意味着减少投入)前后的风险分别是 $r_k(0)$ 和 $r_k(u_k)$,将 $r_k(0)$ 和 $r_k(u_k)$ 分别简记为 r_k 和 r_k' ,这时 r_k 和 r_k' 及 u_k 之间的关系用式(2)表示,其中 g_k 取决于任务k本身的特性和组织的过程能力。

$$r_k' = g_k(r_k, u_k) \quad (2)$$

任务k 的风险发生了变化,整个项目的风险将随之发生变化,可以通过增加项目中某个或某几个任务的投入来调整整个项目的风险。但是增加的投入与风险的下降之间呈非线性关系。对单个任务而言,其风险降到一定程度后,继续增加投入时风险一般不再明显下降。而且软件项目具有特殊性,不同阶段增加投入导致风险降低的效果是不一样的。因此仅靠对单个任务增加投入,可能难以将整个项目的风险降至管理者可以接受的范围。下面分析基于整个项目的多个任务风险优化控制问题。

假定项目在计划条件下追加一定数量的投入用于控制项目的风险(追加的这部分投入以下简称“风险控制投入”),追加的风险控制投入总量为 U_{\max} 并在风险传递路径上有n个任务。

若 U_{\max} 在n个任务之间共有m种不同的分配方案,对于其中的第k种分配方案,各个任务增加的风险控制投入分别为 $u_{1k}, u_{2k}, \dots, u_{ik}, \dots, u_{nk}$,相应地各个任务单元的工期风险分别变为 $r_{1k}, r_{2k}, \dots, r_{ik}, \dots, r_{nk}$,这时,求解整个项目的风险优化控制问题用下式表示:

$$\begin{aligned} \text{Min} R_k \\ R_k = 1 - \prod_{i=1}^n (1 - r_{ik}) \\ r_{ik} = g_i(r_{ik}^*, u_{ik}) \end{aligned} \quad (3)$$

$$\text{where } 0 \leq \sum_{i=1}^n u_{ik} \leq U_{\max}, 1 \leq k \leq m, 1 \leq i \leq n,$$

$$\{u_{ik} | 1 \leq i \leq n\} \in X, \{r_{ik} | 1 \leq i \leq n\} \in Y$$

R 代表项目的风险, r_{ik} 代表 k 中方案任务 i 的风险。 X 和 Y 分别表示管理者可以选用的风险控制方案集合及相应的风险状态集合。 $U_{\max}=0$ 意味着项目的风险控制投入为 0, 即在原计划投入条件下, 项目各个任务间进行资源调配和优化。

3.2 算法设计

式(3)属于离散优化问题, 目前离散优化的求解算法主要有穷尽搜索法和离散控制优化算法。 穷尽搜索的算法效率太低^[13], 只适用于很小的软件项目; 使用后一种算法, 则要求风险控制投入与风险状态之间具有精确的对应关系^[14], 只适用于软件过程非常稳定且完整定义的软件项目。 因此这两种算法目前都不能很好地适用于求解软件风险的优化控制问题。

下面我们设计用动态规划算法来求解上述软件风险优化控制问题。 假定通过风险分析已知项目当前各个任务的风险状态, 且风险控制投入总量为 U_{\max} 。 对于一个软件开发过程中的某个任务, 当风险控制投入即式(3)中的 u_i 已知时, 根据当时的风险状态(r_i), 可从过程数据库中得到相应的风险数据 r_i , 或由风险管理专家根据过程数据库中类似项目的风险数据给出 r_i 。

我们连续地列出函数 $f_0(h_0), f_1(h_1), \dots, f_j(h_j), \dots, f_n(h_n)$ 。 这里:

h_0 : 项目的所有任务都没有风险控制投入($h_0=0$), 这时项目的风险为项目初始状态下的风险, 为统一起见, 这时项目的风险用 $f_0(h_0)$ 表示。

h_1 : 项目的风险控制投入只分配给任务 1 且总投入为 h_1 ($0 \leq h_1 \leq U_{\max}$)。 除任务 1 的风险发生变化外, 其他任务由于没有风险控制投入而保持原风险不变。 这种情况下整个项目的风险用 $f_1(h_1)$ 表示。

h_j : 项目的风险控制投入只分配给任务 1 到 j 且总投入为 h_j ($0 \leq h_j \leq \sum_{i=1}^j u_{ik} \leq U_{\max}$)。 1 到 j 的各个任务的风险都发生了变化; j 之后的任务由于没有风险控制投入而保持原风险不变。 这种情况下项目的风险用 $f_j(h_j)$ 表示。

h_n : 项目的风险控制投入分配给所有任务(1 到 n)且总金额数为 h_n ($0 \leq h_n = \sum_{i=1}^n u_{ik} = U_{\max}$)。 此时整个项目所有任务的风险都发生了变化, 这种情况下整个项目的风险用 $f_n(h_n)$ 表示。

$f_j(h_j)$ 值通过以下的动态规划计算公式求出(公式推导过程略):

$$\begin{aligned} f_0(h_0) &= f_0(0) = 1 - \prod_{i=1}^n (1 - r_i(u_i^0)) = 1 - \prod_{i=1}^n (1 - r_i^0), \quad h_0^0 = 0, u_i^0 = 0, r_i^0 = r_i(u_i^0) \\ f_j(h_j) &= \min_{u_j} \{1 - (1 - f_{j-1}(h_{j-1} - u_j)) \times b_j(u_j)\}, \quad b_j(u_j) = (1 - r_j(u_j)) / (1 - r_j^0(u_j^0)) \\ f_j(h_j) &= \min_{u_j} \{1 - (1 - f_{j-1}(h_{j-1} - u_j)) \times b_j(u_j)\}, \quad b_j(u_j) = (1 - r_j(u_j)) / (1 - r_j^0(u_j^0)), \quad j = 2, 3, \dots, n \end{aligned}$$

4 应用示例

假设项目共有 3 个串行的任务, 根据过程数据库中的历史数据得出这 3 个将要执行的任务不能按进度计划完成的风险分别是 30%、20% 和 10%。 项目的风险预备金 U_{\max} 为 4 000 元。 假定项目风险小于 20% 为管理者可以接受的风险

范围。 管理者可供选择的风险控制方案及相应的风险数据见表 1(风险控制投入为 0 代表没有在原计划条件下增加或减少投入, 即初始状态)。 其中: 风险控制投入单位为千元, 负数代表减少投入, N/A(not available)代表不可用。

根据表 1, 在项目资源投入总量不变即项目风险控制投入总量为 0 时, 项目的过程资源分配共有 3 种情况: (1) 任务 1、任务 2 和任务 3 的风险控制投入皆为 0, 即项目的初始状态; (2) 任务 1、任务 2 和任务 3 的风险控制投入分别为 1、0 和 -1, 即在任务 1 和任务 3 之间调配资源; (3) 任务 1、任务 2 和任务 3 的风险控制投入分别为 0、1、-1, 即在任务 2 和任务 3 之间调配资源。 根据式 (1) 可以计算出上述 3 种情况下整个项目的风险 R 分别为 50%、40% 和 45%。

表 1 管理者可以采取的风险控制方案

任务序号 No.	1	2	3
风险控制投入(u_i)			
0	30%	20%	10%
-1	N/A	N/A	12%
1	15%	10%	5%
2	5%	5%	N/A
3	N/A	1%	N/A

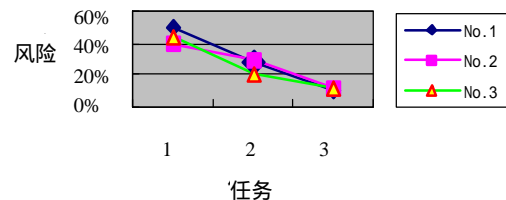


图 2 $U_{\max}=0$ 时 3 种情况下的项目风险分布

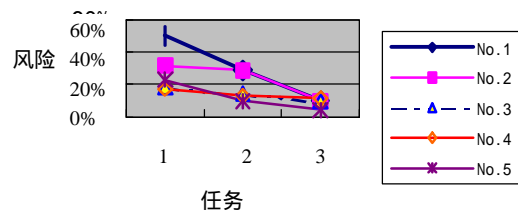


图 3 $U_{\max}=4 000$ 元经过优化控制的项目风险分布

显然, 在项目没有增加风险控制投入($U_{\max}=0$)的上述 3 种情况中, 软件项目的风险也不一样, 如图 2 所示。 尽管第 3 种情况下的项目风险最小, 但仍超过管理者可以接受的范围, 管理者可选择利用风险控制投入来降低整个项目的风险。 图 3 为利用本文给出的优化模型和算法对给定的 4 000 元对项目风险进行控制优化的结果。 其中 No.2 为增加的 4 000 元只投在任务 1 而其他任务投入不变的情况下经过风险优化控制后的项目风险分布; No.3 为增加的 4 000 元投在任务 1 和任务 2 而任务 3 的投入不变的情况下经过风险优化控制后的项目风险分布; No.4 为增加的 4 000 元投在 3 个任务的经过风险优化控制后的项目风险分布。 为了比较和说明问题, 图 4 中还增加了 No.1 和 No.5 两种情况。 其中 No.1 代表项目的原始状态下(没有风险控制投入)项目的风险分布, No.5 为增加的 4 000 元投入随意分配在 3 个任务即未经风险优化控制的项目风险分布。

显然在增加风险控制投入 4 000 元时, No.4 的项目风险最

小(17%),降至管理者可以承受的风险范围 20% 以下,因而在原计划基础上对任务 1、任务 2 分别增加 2 000 元、3 000 元同时任务 3 减少 1 000 元的资源配置方案最为合理。相反,未使用本文给出的优化方法随意分配风险控制投入如 No.5,虽然同样增加 4 000 元风险投入但项目的风险依然较高(23%)。

5 总结

本文阐述了基于 CMM 的软件风险管理过程及特点,提出一种优化软件风险控制的方法,为管理者进行风险控制决策提供决策依据。本文给出的风险优化控制模型基于 CMM 的过程框架,从而将风险控制与软件过程工程及过程管理统一在一起,具有明确统一的操作规范、广泛的适用性,而且可以复用过程数据库中以往类似项目风险管理的历史数据和经验。所设计的动态规划算法针对实际,且简单易用。

应用实例确认了上述方法的实际效用。随着组织的软件过程成熟度的提高,以及软件过程数据库的不断充实和完善,上述方法的准确性将随之提高。将上述方法和已有的一些项目管理工具集成起来是未来研究和发展的方向。

参考文献

- 1 Boehm B. A Spiral Model of Software Development and Enhancement. Tutorial: Software Engineering Project Management, IEEE Computer Society Press, Washington, DC, 1987:28-142
- 2 Charette R N. Large-scale Project Management Is Risk Management. IEEE Software, 1996-07:110-117
- 3 Brown W. AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis. New York, Wiley Computer Publishing, 1998

- 4 Brown N. Industrial-strength Management Strategies. IEEE Software, 1996-07: 94-103
- 5 Jalote P. CMM in Practice: Process for Executing Software Projects at Infosys. Addison-Wesley, 2000: 5-117
- 6 Madachy R. Heuristic Risk Assessment Using Cost Factors[J]. IEEE Software, 1997-05/06:51-59
- 7 Risk Management Overview. Copyright 2002 by Carnegie Mellon University, Last Modified: www.sei.cmu.edu, 20020-04-03
- 8 Williams R. Putting Risk Management into Practice[J]. IEEE Software, 1997-05/06:75-82
- 9 CMMI Distilled: A Practical Introduction to Integrated Process Improvement. Translated by Zhou B S. China Machine Press, 2002: 79-84
- 10 Elaine M. Managing Risk: Method for Software Systems Development. Addison-Wesley, 1998: 69-101
- 11 Alfred B. Process-based Software Risk Assessment. Proceedings of the 4th European Workshop on Software Process Technology (EWSPT' 95), Nordwijkerhout, Netherlands, 1995-04
- 12 Xu Ruzhi, Qian Leqiu. CMM-based Software Risk Control Optimization[C]. Proceedings of the 2003 IEEE International Conference on Information Reuse and Integration (IRI-2003), Las Vegas, 2003:499-503
- 13 Chang C K, Christensen M. A Net Practice for Software Project Management. IEEE Software, 1999,16(6): 80-88
- 14 Sukhodolsky. Object State Monitoring Optimization. In International Journal of Applied Science and Computations, 1997,3(3): 239-250

(上接第 55 页)

需通过控制控件版本号来进行更新升级,维护也很方便。实践表明,该组件能很好的应用到网络 GIS 系统当中,系统运行速度快,可扩充性强,整体性能良好。图 3 就是在该 GIS 组件的基础上用脚本、ASP 等开发集成的一个网络 GIS 运行效果。

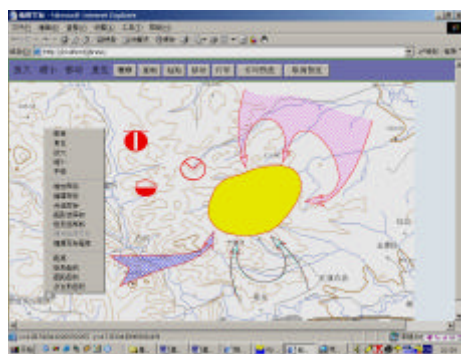


图 3 系统运行效果

3 结语与展望

按照 OpenGIS 规范设计的 GIS 组件能够实现一定的空间信息的互操作,其空间数据除了可以通过常用的 Arc/Info、AutoCAD、MapGIS、MapInfo 标准交换格式进行交换之外,还可以通过该规范提出的 wkt 和 wkb 格式进行交换与共享,也可以通过 GML 语言来实现空间数据的网络传输与共享,增加了数据运用的灵活性;同时其他系统可以按照规范定义

的标准接口来调用各种 GIS 操作,从而实现一定的 GIS 数据与操作的共享。同时由于组件技术的特点,在该 GIS 组件上开发各种应用系统方便灵活,开发效率高,支持各种开发方式与开发语言,提高了运用的灵活性。

在该地图组件的设计与实现中,由于考虑到组件在其他环境的通用性,所有的接口的基接口都由 Iunknown 改为 Idispatch 接口,同时考虑到组件的实用性,对其对象模型进行了一定扩充,增加了一些新的对象,对一些接口也适当增加了一些新的方法。

OpenGIS 规范现在还在不短的发展与成熟之中,SFO 实现规范只是对二维空间内简单几何对象的接口进行了统一,还未制定三维 GIS 空间实体的标准接口,随着 OGC 的不断发展,GIS 界的共同努力,相应的 OpenGIS 规范将得到不断的修改与完善,地理信息的真正共享与 GIS 系统之间的互操作必将得以实现。

参考文献

- 1 <http://www.opengis.org>
- 2 Platt D S. COM 精髓(第三版).北京:人民邮电出版社,2001,(1)
- 3 Open GIS Consortium. OpenGIS Simple Features Specification for OLE/COM Revision 1.1. 1999
- 4 林春,王波,贺贵明.利用 MapX 实现的组件式 GIS 技术.计算机工程,2002,28(5):217-218,271
- 5 Rogerson D. COM 技术内幕.北京:清华大学出版社,1998-1