APPLYING USE CASES

SECOND EDITION

需求与设计系列

软件工程丛书

用例分析技术

（第2版）

姚淑珍 李巍 译

Gen Schneider Jason P. Winters 著

中信出版社

CITIC PUBLISHING HOUSE

APPLYING USE CASES

用例分析技术

软件工程丛书

姚淑珍 李巍 等译
Geri Schneider Jason P.Winters 著

中信出版社
CITIC PUBLISHING HOUSE

图书在版编目（CIP）数据

用例分析技术（第2版）/ 施奈德（Schneider G.）等编著；姚淑珍等译. -北京：中信出版社，2002.6
（软件工程丛书）
书名原文：Applying Use Cases, Second Edition
ISBN 7-80073-454-4

Ⅰ. 用… Ⅱ. ① 施… ② 姚… Ⅲ. 软件工程 Ⅳ. TP311.5

中国版本图书馆CIP数据核字（2002）第018972号

用例分析技术（第2版）

著　　者：Geri Schneider, Jason P.Winters
译　　者：姚淑珍　李　巍等
责任编辑：赵　平　车立红　**责任监制**：朱　磊　王祖力
出 版 者：中信出版社（北京朝阳区新源南路6号京城大厦　邮编　100004）
经 销 者：中信联合发行有限公司
承 印 者：北京忠信诚印刷厂
开　　本：787mm × 1092mm　1/16　　**印　　张**：12.5　　**字　　数**：289千字
版　　次：2002年6月第1版　　**印　　次**：2002年6月第1次印刷
京权图字：01-2001-5205
书　　号：ISBN 7-80073-454-4/TP・20
定　　价：35.00元

译 者 序

用例技术通过用例、执行者和用例以及用例之间的关系来描绘系统外在可见的需求情况，它是用户和开发者共同剖析系统功能需求的起点。随着用例作为UML的一种重要表示法，用例分析开始在软件开发中占据决定性地位，用例技术对于项目管理员、高级分析人员、设计人员、程序员、测试人员、用户都非常重要。在用例技术越来越得到重视的今天，非常需要一本新书以通俗易懂的方式来说明用例表示的语意及其应用技术。

本书以一个邮购公司的订单处理系统为实例，通过实际开发人员连续不断的讨论和工作，展现了整个系统的开发过程和技术应用成果。翻译过程中，这种讨论式的系统开发方式经常把我们带入场景，与作者共同去思考可选方案和可能犯的错误，找出有效的解决办法。相信你也会从实际系统开发过程中更进一步体会用例的含义和应用技巧，从而达到最终的目标。

《用例分析技术》一书从多个不同的角度观察用例，展示用例如何有助于项目的体系结构设计、进度安排、需求分析、测度和验证。特别从用户的角度观察整个系统，讨论诸如边界、接口和界定范围等问题，讲述如何将一个实际的大型系统划分为可管理的多个部分，即系统功能分解技术。本书适合各种层次的软件开发人员使用。

全书共分11章和5个附录。第1到第6章由姚淑珍负责翻译，第7到第10章由李巍负责翻译，第11章、附录由李鸣负责翻译。钏伟、张亮、张煜、孙志强、沈燕芳、李搏、牟胜梅、余文舟、张春晓翻译部分内容。姚淑珍完成全文的最后统校。李鸣做了最后的文字修正。由于知识局限和时间仓促，不免有翻译不到位甚至错误之处，敬请读者批评指正。我们会接受批评并向原书作者表示歉意。

序

当我在1967年根据从事大型电信交换系统及其系统设计工作的经验，首次提出一组新型建模概念的时候，将用例作为一种分析方法还只是很粗略的想法。伴随着面向对象思想的出现，以及我后来在80年代应用面向对象方法以及对Objectory规则进行规范化所做的一系列工作，用例分析开始有了显著的发展，并且开始在问题分析领域占据重要地位。今天，用例所体现出的思想已经成熟，并且这项技术已经成为分析员必不可少的重要工具。

随着用例与工业标准的建模语言UML的结合，需要一本新书以实用易懂的方式来阐明当前的表示法和语意。用例分析也已经在新的软件开发统一过程中占据重要地位。于是，对于经理、设计师、设计人员、分析员、专家、程序员以及测试人员来说，理解如何使用用例变得十分重要。

在《用例分析技术》这本书中，Geri Schneider和Jason Winters已经为介绍这一强有力的工具和演示它们在现实生活中如何实现做了大量出色的工作。不是预先给出完整答案，书中的例子与实际项目的进展状况一样，早期的模糊模型随着小组对项目理解的加深而变得精确起来。这种方式有利于对实际项目中发生的问题进行介绍。

本书清晰地反映出Geri作为Rational 软件公司的培训人员所具有的经验，她在指导和培训Wyyzzk Training and Consulting公司客户方面的大量倾注以及Jason使用该技术指导Lucent Technologies工程师的经验一并呈献在本书中。我极力推荐那些需要了解用例分析的人阅读本书。

伊瓦尔·雅各布森
(Ivar Jacobson)

前　言

开始一项新的项目，难如登月。但是，如果你组建了强大的小组，乘上需求的航船，就有希望到达成功的彼岸。成功的彼岸没有失败的项目，利润率很高，满地铺满了黄金。

但是，从项目开始到成功会遇到很多危险，很多项目半路就夭折了。有人说，差不多80%的项目从未获得成功。你可以询问一下那些曾经尝试过的人。他们总是会这样说："使用面向对象作为工具吧，Booch、OMT、OOSE和UML都是可供选择的良好模型。同时你也需要一幅图表来描绘开发过程中的风险和主要的体系结构。最后，还需要绘制一系列的用例图来达到你的目的。"

用例包含于统一建模语言（UML）中，应用于Rational Unified Process的整个过程。它们在不同的商业和工业领域获得了广泛的认可。通常，用例被用在软件项目和企业级的应用中。

本书为那些在项目开发中对用例技术感兴趣的人而写。但是我们不能保证使用用例开发的项目一定会获得成功。本书可以使你从不同的角度来观察正在开发的项目，并提供一些工具，以增加项目开发成功的可能性。

如果你有一些面向对象的基本概念和知识，你将在本书中获得更大的益处。我们使用统一建模语言作为表示法，并解释这些表示法。比较好的一本关于表示法的参考书是Fowler所写的UML Distilled，这是一本极好的书，通俗易懂。

本书使用Rational Unified Process作为框架来进行组织。在过程的各阶段中，我们将讨论阶段中的活动，重点放在基于用例的活动上。我们将简要谈论一些与用例相关的活动，例如软件体系结构、项目管理和面向对象的分析与设计。这些都是非常重要的活动，每一个话题都可用一本书的篇幅进行论述。因此，在附录A的资源列表中，你将发现关于这些话题的一些推荐书籍。

用一个为邮购公司设计的订单处理系统作为贯穿全书的例子。这将使我们的思维保持连贯，并组建合理而复杂的例子。这一解决方案的不同部分在不同的章节给出，来阐明用例的观点。

本书以一系列连贯的步骤进行介绍，尽管现实生活从未那么简单。每一部分都与其余的部分相连，直到整个系统完成为止。因此如果某一个章节说要创建一个体系结构，你只要应用你自己所掌握的知识，做当时能够做的就可以了。你将在工作过程中随着知识的增长对先前的项目进行补充与完善。

在开始使用用例之前，并不需要读完整本书。一到六章给出了使用用例的基本知识。我们推荐所有的读者首先阅读这些章节。第9章论述了体系结构和将用例映射到

体系结构之中的内容。第7章包含了归档用例的内容。第10章包含了使用用例进行项目规划的内容，第8章论述了如何评审用例文档。第11章讨论了从用例到OOAD过渡的有关内容。

最后，要用用例来归档系统。附录A提供了我们所参考的书目列表，同时还有一些在项目开发过程中十分有用的书籍。附录B给出了所使用的文档模板。这些为你自己的项目提供了范例和基点。你可以根据需要对它们进行修改，

1995年10月，Rational Software Corporation与Objective Systems合并，从而引入了Ivar Jacobson和他的用例。1996年2月，我为Rational公司编写了第一个用例课程，并且将其和Grady Booch以及Jim Rumbaugh的面向对象方法学结合在一起。从那时起，我开始为Rational公司的客户和我自己的咨询公司（Wyyzzk）的客户讲授有关用例的知识。在为他们讲述用例知识的同时，我也从这些客户身上学到了很多东西。这本书出自我在工作实践中的实际经历。

第2版前言

自从1998年9月第1版发行以来，我们和UML语言都发生了许多变化。本书将反映最新内容。

第2版包含了第1版的内容，只是章节不同。我们将面向工程的内容移到了本书的末尾，将面向商务的内容移到了本书的开头。这将使不同的读者更容易发现他们感兴趣的资料。

我们将本书更新到UML1.3。第3章和第4章发生了许多变化，因为我们在那里给出了绝大多数表示法。使用关系变成了UML1.3中的包含和泛化两个关系。多扩展关系变成了单扩展关系。在这两种情况下，表示法也发生了相应的变化。对特定场景的定义也有一点变化。我们过去称之为场景（scenario）的现在称之为路径（path）。

我们补充了一些十分有用和重要的材料。第6章是在用例中设置细节层次的新的一章，包含有关商业处理层次用例的信息和在不同层次细节上保持用例之间的可追溯性。第7章的归档用例包含了有关处理诸如登录、创建、读取、更新和删除等用例的思想。第8章评审增加了我们所遇到的常见错误及其更正方法。我们在第5章和第9章对于时序图加进了更多的信息。

我和贾森的情况也发生了变化。Jason离开了Octel，现在是Cadence Design Systems的助理工程师。我喜欢拥有自己的公司，但是不喜欢管理账目，因此，到Andrews Technology公司负责面向对象的部门。我们仍然拥有Wyyzzk公司，Jason为该公司提供周末业务咨询。出版商方面甚至也发生了变化。Addison-Wesley现在是Pearson Education的一部分，并且我们有一个全新的团队来管理对象技术系列丛书。他们很容易共同协作，使过渡尽可能的平滑。

我们被问及到的最多的一个问题是：脚印和对话图标代表什么意思？脚印图标主要是表示进入过程，对话图标表示进入故事场景。

感谢所有为此书而发E-mail的人。由于时间原因我们不能一一回复，但是我们阅读了所有来信，并且希望在第2版中回答了您的绝大多数问题。

非常感谢我们的著名评论家，他们与我们一样努力工作，使这本书得以出版：

- Lauren Thayer
- Venkat Narayanan
- Guy Rish
- Kelli A.Houston,Rational Software Corporation
- John Sunda Hsia

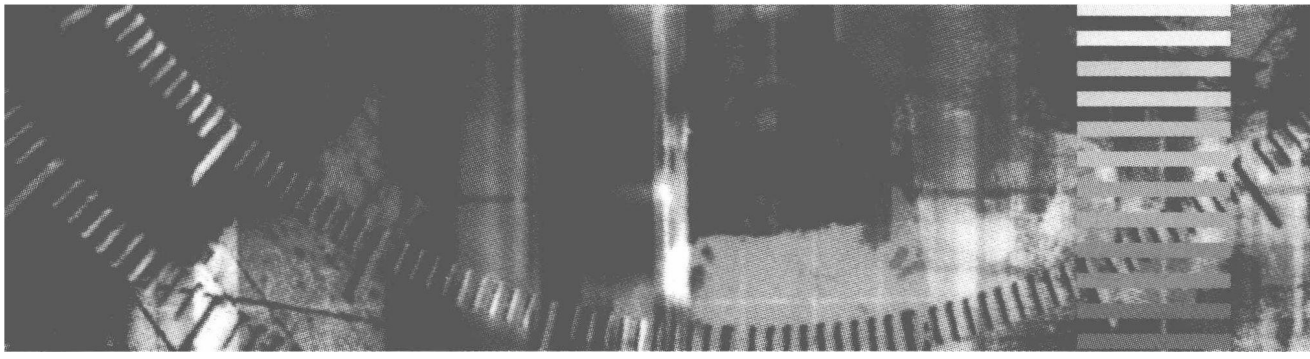
Paul Becker、Ross Venables、Tyrrell Albaugh和她在Addison-Wesley的生产小组

工作非常认真刻苦，很高兴能与他们一起工作。我们衷心地感谢他们的支持和耐心。为使本书得以出版，他们做了许多艰苦的工作。

最后，要提到几个特别的人。感谢Lauren Thayer和Kristy Hughes，他们是我近15年的老朋友。感谢我的两只小猫：Patches和Joker，是它们陪伴我度过了在计算机前的时光。感谢Jason Winters的爱、支持和鼓励。他们一直在帮助我前进。

Geri Schneider Winters
Santa Clara, California

目　录



第1章

开　始

用例被用来描绘一个系统外在可见的需求情况，常被用在项目的需求分析阶段，对项目的测试计划和用户指南也有用处。它们被用来创建和验证被提议的设计，并确保该设计满足所有的需求。用例也被用来在创建项目的计划时决定在每一个版本中应该加入什么内容。

本书将对在项目中如何使用用例给出实践性的指导。全书覆盖从最开始到实际建造一个系统之前的项目开发过程。我们也会涉及到在系统测试和创建用户手册的时候如何使用用例。

本书将从多个不同的角度观察用例，展示用例如何有助于项目的体系结构设计、进度安排、需求分析、测试和验证。我们将从用户的角度观察整个系统，讨论诸如边界、接口和界定范围等问题，了解如何将一个实际的大型系统划分为可管理的多个部分。我们也将着眼于谁会对你写的文档感兴趣以及在评审中应该寻找什么，如何使系统具有较好的适应性、如何制定构建与购买决策以及如何将文档变成面向对象设计等问题。

本书并未详细阐述有关软件体系结构、项目设计、测试、过程和方法学等方面的问题。附录A中有关于这些话题的书籍列表，那里有相当多的好书，该列表只是给你一个进一步学习的起点。

1.1　迭代式软件过程

用例可以用于多种过程之中。我们感兴趣的是迭代和风险驱动软件过程，通常情况下，它和用例及面向对象方法学一起使用。迭代式软件过程有助于在过程的早期对风险进行鉴别并加以解决，从而获得更加健壮和高质量的系统。最常使用的迭代和风险驱动过程的一个例子是Rational Unified Process(简称RUP)。这里，我们将简要介绍一下RUP，以展示用例用在了过程的哪个地方。后面的章节将会更详细地介绍在各个阶段如何使用用例。

RUP由四个主要的部分组成：初始、细化、构造和交付。

在初始阶段，应该了解项目范围，并且为其创建商业用例。在初始阶段结束时，

你应该能够回答：继续这个项目是否有益。

在细化阶段，将进行需求分析和风险分析，开发出基本的体系结构，并且为构造阶段创建计划。

在构造阶段，需要一系列的迭代。每一次迭代都包括分析、设计、实现和测试等工作。

在交付阶段，把已经开发出的项目完善成为产品，包括Beta测试、性能调整和创建诸如培训手册、用户指南以及销售工具等文档。还要为内部或者外部的用户团体制定相应的产品推出计划。

那么，用例应用到所有这些阶段的什么地方呢？在初始阶段，开发出高层用例，以帮助划定项目的范围，例如：我们应该在这个项目中包含什么功能？哪些应属于另一个项目？如何在规定的进度和预算范围内完成项目。

在细化阶段，需要开发更细节化的用例。这将有助于风险分析和体系结构的建立。用例将被用来创建构造阶段的计划。

在构造阶段，以用例作为设计和开发测试计划的起点。更加细节化的用例可以作为每一次迭代分析的一部分进行开发。用例提供在每次迭代中必须满足的一些需求。

在交付阶段，使用用例来开发用户指南和培训手册。

1.2 示例项目

我们使用一个贯穿全书的例子，把所有的技术用在这个例子上。并用UML作为表示语言，详见附录C。

这是一个为一家邮购公司设计的订单处理系统。让我们从最开始谈起，四个人饭后围坐在桌旁，其中有人有了一个想法。

“这很疯狂!”Dennis坐在Tara的对面，大声叫道，几乎将咖啡泼了出来。

“什么？”Lisa坐在Gus旁边，端着咖啡问道。

“事实上我没法找到可以为我们提供合理服务的一家供应商。我遇到过这样一个供应商，他可以提供出色的服务，并且我也喜欢和他打交道。但是即使是最简单的订单他也要花三周的时间给我答复。不仅如此，还有别的事情。我当然可以在三天之内获得订单，但这样的订单经常出现错误，并且当我把这种情况反馈给他们时，他们的答复让我感觉这是我自己的过错。看起来好像是他们在让我选择其他供应商。”

“我知道您的意思。”Lisa说道，“我自己与邮购公司也有这个问题。你一定想到他们应该更加注意客户的想法。”

我认为我要是做的话，可以做得更好，我很清楚什么不应该去做。”

这是Dennis几个月前经常抱怨的事情。现在大家已经习以为常了。Tara突然笑了笑，说道：“为什么你不开一个呢？”“开什么？”Dennis嘴里含着咖啡问道。

“开一个邮购公司啊！你会采取什么办法来解决遇到的问题呢？”

“是啊，看起来将订单处理自动化要好一些，这可以使我有机会自己开办公司。但是我对软件一无所知。”

就在此时，Gus加入了谈话，“你应该把它列一个计划。我在面向对象的课上曾经学过一种方法，可以在这里使用。并且我们还可以帮忙。我们可以把经验汇集起来，并且发挥我们各自的才能来解决那些我们不知道的部分。”

“面向对象？那是什么？你知道我不是程序员，我不懂任何编程语言。”

“不，面向对象并不是编程语言，而是一种思考问题的方法，是一种用于建模，将问题分解为可以区别的对象以使你能够使用它们来开展工作的方法。问题是什么并不重要，是否是一个编程问题也并不重要，开办公司的问题也不重要。它只是一种取决于你如何看待它的方法。”

“嗯……”Dennis沉思了一会儿，好像想到了更多的事情，“那么你们愿意帮忙吗？”

“当然了。”

“为什么不呢？”

“听起来挺有趣。”

“那好，Gus，我们应该从何处开始这一奇妙的面向对象过程呢？”

在开始写用例之前，应该搜集一些如何开始着手的信息。这也是RUP初始阶段的一部分。所搜集的信息应该包括：项目描述、影响项目的市场因素、项目的风险因素和所做出的假设。本章余下部分将讨论有关这些信息来源的问题。

1.3 项目描述

你已经对项目有了一个了解，下一步是把要做的事情描述出来。这听起来很简单，实际上也如此。但是进行项目描述的小组越大，所花费的时间就越长，并且也越复杂。最好的方法是由一两个人写出一个简短而完整的项目描述。

项目描述的大小应该介于为小型项目而写的简短的一段话和为实际大型项目而写的几页纸之间。这并不是对需求的描述，而是从总体上对项目进行描述。

在这一点上人们犯的最大错误是不写项目描述。这通常是因为每个人都觉得他们知道项目是什么，因而没必要把它写出来。我们已经花了几天来讨论怎样用一段话来描述整个项目。在将其写到纸上之前，你不能肯定小组中的每一个人都同意这个项目描述。

“那么就开始吧。我们应该用日常语言把这个订单处理系统描述出来。它将变成问题陈述，这是开始获得项目需求的一种途径。”

"为什么需要把它写下来？我们对从邮购公司定购商品都十分熟悉。对于这样简单的一个问题，似乎有点小题大做了。"

"是这样的，我们把它写下来是要确保所有的人看法一致。即使只是一个人工作，把它们写下来仍然是一个好方法，因为这样可以避免遗漏一些重要的东西。除此之外，它为我们提供了出发点，我们可以在以后的过程中对它进行添加和补充。现在，我要开始了。"

问题描述

我们正在为National Widgets邮递公司开发订单处理系统。这是一家转售各种商品的公司。

这家公司每年发布两次产品目录，并将其邮递给客户和其他感兴趣的人。

"你认为每年两次很好吗？如果我们的产品变化的更快些会怎样？"

"记住，这只是我们的开始。我们会在进展中对它进行补充和完善，而且对我们所做的一切会有更深刻的理解。让我们继续吧。"

更多的需求

客户以提交产品列表并向National Widgets公司付费的方式购买商品。National Widgets公司填写订单并将产品运送到客户的地址。

订单处理软件记录从收到订单直到商品被运送给客户的整个过程。

National Widgets公司将提供快捷的服务。他们应该能够以最快、最有效的方法来运送客户订购的产品。

"好极了！那看起来很像我们的想法。现在我们应该做些什么呢？"

我们的朋友在哪些方面做得不错呢？他们保持了描述的简洁，他们所谈论的是应该实现什么，而不是应该如何去实现。他们写下了对他们来说重要的元素：这是一家目录公司——一家经销商，而不是生产商；它可以提供快捷的服务，软件用于在整个过程中记录订单。这些是他们的项目中的关键特征。不必担心他们的描述不够完美。如果有不应该做的事情，他们也会把这些事情写下来。他们现在有一个达成共识的基本的描述，但以后可能需要修改。不管怎样，这些关键特征不会改变。

1.4 开始风险分析

既然已经有了对项目的描述，下一步就是把你所知道的有关项目的其他情况写下来。你在寻找影响项目的或好或坏的因素以及可能导致项目失败或者被客户拒绝的一切。我们将以问题陈述的方式使用这些成分来创建用例、其他需求和风险因素。从考虑市场因素开始：

- 竞争对手是谁，竞争点是什么?
- 你依赖的是何种技术，例如：
  - Web
  - 对象数据库(Object Databases)
  - Power PC芯片
- 影响项目的市场因素
- 你所依赖的未来趋势，例如：
  - 更多的家庭办公
  - 更多的小公司
- 情况是否会变成：
  - 不能快速适应市场
  - 对于市场来讲过快

对于我们的邮购公司，可以根据个人经验创建市场因素的列表。

邮购市场因素

> 在大多数家庭中，所有成年人都工作，至少是兼职工作。他们很少有时间购物，因此乐于采用邮购等便利方式。
>
> 网上购物和家庭购物网络很受欢迎，是邮购市场的竞争者。
>
> 其他的邮购公司提供24小时订单接收服务，交付时间从一天到两周不等；此外还有礼品打包服务，并提供总额折扣。

在编写这张表格的时候，应该有点创造性。要多动脑筋，要把你能想到的问题都记录下来，把所有可能发生甚至最不可能发生的事情也都记录下来。在这一阶段，应从多个观点来看待这个项目，这有助于坚定你的观点。要找一些有关市场发展趋势的书来看，要弄清竞争公司对在哪里，错在哪里。

在项目中也应该考虑风险因素，要考虑可能出错的事情。仅仅记下你所希望的事情发生必然导致失败的灾难。首先考虑事情会如何出错，并为其做好计划，这要比盲目前进，遇到问题时惊惶失措好得多。

也需要考虑取得惊人成功的可能性。有时你会发现，如果看上去过于成功，事情实际上在出错。例如，在第一个月里，如果National Widgets公司接到4 000个电话定购将会发生什么事情?他们将不得不处理大量的订单和数据。即便公司能够处理这种情况，处理软件是否能够应付?如果软件不能及时处理所有需求，公司就会失去客户，并有可能因此而声名狼藉，我们的朋友将把这作为其中的一个风险因素写下来：

- 人
  - 团队没有经验
  - 团队对于所使用的技术不熟悉
  - 不能雇佣到有合适背景的人
- 系统

- 系统每个时间段的交易量
- 预期的用户数
- 一些功能的预期寿命
- 必须与之接口的早期系统，如：
  - 软件
  - 商业过程
  - 数据存储和数据库

• 资源
- 周期太短
- 用户数过多
- 供应商不能提供我们信赖的商品

• 技术
- 所依赖的技术发生了变化

• 协作
- 缺乏用户的认可
- 公司增长过快

关于订单处理项目的风险因素，我们考虑了团队缺乏经验、系统错误和市场需求这几个因素。

National Widgets的风险因素

> • 有些设计软件的人缺乏经验。
> • 如何在系统出错时防止丢失订单？
> • 系统必须易于操作以使非专业人士可以使用。
> • 如果不支持Web接口是否会成功？
> • 如果系统突然收到大量订单会出现什么情况？
> • 如何处理公司不同部门的众多用户？
> • 如何应付数据库崩溃？

对于风险清单和市场因素清单，除去极端的情况，例如彗星撞击到你的公司使你的公司关闭，或者太阳不再升起，对其余因素进行优先划分。新的经过优先级划分的清单是你所创建的第一个风险分析，列出的所有的事情使你陷入了无法完成项目的危机之中。首先列出最重要的事项，在底部列出不太重要的事项。如果你希望自己的项目成功，就必须解决高风险性因素。

对于National Widgets公司的风险因素，如果我们相当确信如果不解决某些风险项目就会失败，并且我们肯定它们会发生，那么，这些就是高风险因素。对于这个项目，我们选择三个风险因素作为高风险因素：团队缺乏经验；非专业人员的易用性；对Web界面的支持。我们选择一种中等风险因素：众多用户在同一时刻访问系统的需求。