# Library

## 一．该系统可以完成的操作：

1. 借出(或归还)一本书。

2. 获得一个特定作者或特定主题的书单。

3. 找出一个特定的借阅者目前借了什么书(用户只能自己查)。

4. 找出哪一个借阅者最后借了一本特定的书(仅限于员工)。

5. 记录从图书馆(仅限工作人员)增收(或移出)一本书。

## 二．系统的某些功能要求

该系统必须能够快速搜索和更新目录(以避免冗长的结帐队伍，并使在线图书搜索成为卡片目录的可行选择)，并容易处理工作人员用户对潜在的大量藏书的更新和更正。

## 三．系统的完整性约束

a.图书馆所有的书都必须可以被借出。

b.任何副本不得同时提供和借出。

c.借阅者一次借阅的图书数量不得超过预定数量。

d.借阅者一次借出的某本书不能超过一本。

## 四．图书馆系统问题的历史

这个问题的现有历史一直可以追溯到20世纪60年代美国国会图书馆开始其机读目录项目时。当时，人们就已经预见到了像上述那样的在线图书馆系统。对上述问题的简明陈述由Kemmerer完成，他在[Kemmerer 85]中首次将此问题作为规范练习发表。1986年，在第四届软件规范和设计国际研讨会上，提出了Kemmerer问题陈述的一个变体，12篇已发表的论文考虑了它。Wing在[Wing88]中总结了它们的规范和特点。The workshop specification对规范做了一些修改，比如员工用户也可以完成事务1(假设他们会代表普通借阅人去做)。

## 五．设计考虑

这个问题的焦点是在一个大的且大部分是静态的集合中搜索和更新关于单本书的信息，这个问题的明显架构选择是一个面向数据库的系统。

基于这一前提，我们可以考虑许多设计方面的变化:

1.应用程序应该如何与数据库交互？虽然有些结果必须实时生成，但有些事务可以捆绑在一起进行批处理。

2.系统应该有多集中？数据库和应用程序都可以分布在多台机器上。

3.数据库的类型会影响软件架构的选择吗？可以想象，与关系数据库相比，面向对象数据库系统偏向于不同的实现。

# Automated Teller Machine (ATM)

## 一．问题的提出

自动取款机的问题出现在几份报纸上。这个问题最初是由伦巴夫在他的《面向对象设计》一书中提出的，如[Lubars92]所述。

## 二．系统的功能

1.设计软件系统以支持计算机化的银行网络，包括由银行联盟共享的人工出纳机和自动柜员机。

2.每家银行都提供自己的计算机来维护自己的账户，并根据这些账户处理交易。

3.出纳站归个别银行所有，直接和自己银行的电脑通信。

4.收银员输入账户和交易数据。

5.自动柜员机与中央计算机通信，中央计算机与适当的银行结算交易。

6.自动柜员机接受现金卡，与用户交互，与中央系统通信以执行交易，分发现金，并打印收据。

7.该系统需要适当的记录保存和安全措施。

8.系统必须正确处理对同一帐户的并发访问。

9.银行会为自己的电脑提供自己的软件；你要为自动取款机和网络设计软件。

10.共享系统的费用将根据持有现金卡的客户数量分摊给银行。

## 三．体系结构设计需要解决的问题

1.安全机制应该放在哪里，自动取款机还是中央网络控制器？

2.异构系统(每个银行都有自己的软件)的存在应该如何管理？