**经典软件体系结构案例**

http://www.cs.cmu.edu/~ModProb/index.html

**Keyword in Context (KWIC)**

给定一组文本行，创建这些行旋转后的按字母顺序排列的列表。

关于“Keyword In Context”（KWIC）问题，Parnas在他的论文[Parnas72]中给出了一个简明的定义：KWIC索引系统接受一个有序的文本行集合，其中每一行都是一个有序的单词集合，而每个单词都是一个有序的字符集合。任何一行都可以进行“循环移位”，即反复地将第一个单词移除并将其追加到该行的末尾。KWIC索引系统输出按字母顺序排列的所有文本行的所有循环移位的列表。简而言之，KWIC问题要求对输入的文本行进行循环移位，然后按照字母顺序对所有循环移位后的行进行排序。

**Sea Buoy**

自动和按需收集和传输天气数据；允许紧急服务的优先权。

海上浮标支持海上航行。以下是[Boch86]的问题陈述：有一组自由漂浮的浮标，为海上的空中和船舶交通提供导航和天气数据。浮标通过各种传感器收集空气和水温、风速和位置数据。每个浮标可以具有不同数量的风和温度传感器，并且可以被修改以在未来支持其他类型的传感器。每个浮标还配备了一个无线电发射器（用于广播天气和位置信息以及SOS信息）和一个无线电接收器（用于接收过往船只的请求）。一些浮标配备了红灯，在海上搜索行动中，过往船只可能会激活红灯。如果水手能够到达浮标，他或她可以拨动浮标侧面的开关来启动SOS广播。每个浮标的软件必须：

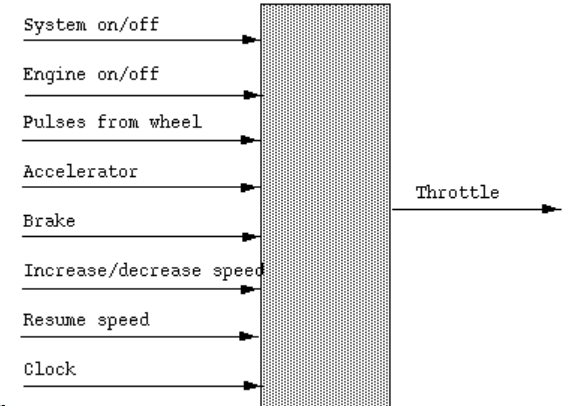
1. 保持当前的风、温度和位置信息；风速读数每30秒一次，温度读数每10秒一次和位置读数每10秒钟一次；风和温度值被保持为运行平均值。
2. 每60秒广播一次当前风、温度和位置信息。
3. 广播过去24小时的风、温度和位置信息，以响应过往船只的请求；这优先于定期广播
4. 基于来自过往船只的请求来激活或去激活所述红灯。
5. 在水手接通紧急开关后连续广播SOS信号；该信号优先于所有其他广播，并且持续直到被经过的船只重置。

**Cruise Control**

维持车辆速度。

巡航控制已经被许多作者用来说明软件设计方法。这个问题陈述源于Booch用来描述面向对象编程[Boch86]的陈述，以及Birchenough和Cameron后来用来比较JSD和OOD的陈述：

当驾驶员打开巡航控制系统时，即使在不同的地形上，巡航控制系统也能保持汽车的速度。当应用制动器时，系统必须放弃速度控制，直到被告知恢复。当驾驶员指示时，系统还必须稳定地增加或降低速度以达到新的维护速度。



这是用于这种系统的硬件的框图。有几个输入：

系统打开/关闭：如果打开，表示巡航控制系统应保持车速。

发动机打开/关闭：如果打开，表示汽车发动机已打开；巡航控制系统只有在发动机打开时才激活。

来自轮子的脉冲：轮子每转一圈就会发出一个脉冲。

加速器：指示加速器被按下了多远。

制动器：按下制动器时打开；如果踩下制动器，巡航控制系统将临时恢复到手动控制。

增加/减少速度：增加或减少保持的速度；仅在巡航控制系统打开时适用。

恢复：恢复上次保持的速度；仅在巡航控制系统打开时适用。

时钟：每毫秒定时脉冲。

系统有一个输出：

油门：工程师油门设置的数字值

**Conference Refreeeing**

征求、审阅并选择会议论文。

举行专业会议是为了宣布和讨论新的成果。组织会议的核心活动是选择要发表的论文。通常情况下，这是通过发出公开邀请，要求提交论文，将提交的论文分发给（地理分布的）评审小组，然后选择最好的论文出现在节目中。一个会议裁判自动化系统应该做到以下几点：

1. 节目委员会宣布“征集论文”。
2. 作者收到论文征集，决定提交论文。他们撰写论文并将其发送给项目委员会。一篇论文可能有几个作者，但只有一个回复地址。

3. 项目委员会收到投稿文件后进行登记。

4. 在某个时间点，项目委员会将论文分发给裁判小组。每一篇论文都会发给三位不同的审稿人，他们都不是论文的作者。

5. 项目委员会不断收集裁判的报告。

6. 在某个时间点，项目委员会会选择论文纳入项目，并将选择情况通知作者。这可能涉及到从裁判那里获得额外的意见。

7. 项目委员会就评选结果向作者提供建议。

**Mailing List Handler**

合并多个来源的地址信息，消除重复并遵守读者偏好。

我们都被目录和其他群发邮件的多份或不需要的副本所困扰。这些主要源于合并多个邮件列表、数据收集中的文书错误以及个人在不同时间以不同形式生成的原始信息。理想情况下，邮件列表系统将收集（甚至传播）更正，合并变体表单，并识别读者对收据的偏好。

邮件列表处理程序接受地址条目、更正和首选项以创建一个或多个邮件列表。它从列表中生成邮件标签。

地址条目包含名称、邮寄地址和读者/供应商信息。更正包括对单个地址条目的更新以及关于合并变体的指导。首选项更新读卡器/供应商信息。邮件列表是地址条目加上控制信息的集合。

地址条目可能是原始的（从原始来源收集，如读者请求），也可能来自其他邮件列表。地址条目也可以作为外部邮件列表（不一定是期望的格式）来接收。更正可能来自内部一致性检查、邮局更正程序、读者信息或其他来源。偏好信息可能来自读者、供应商或其他来源（例如，来自直销协会或USPS不良邮件程序的抑制信息）。

邮件列表处理程序必须维护一组邮件列表。它应该消除重复条目并更正错误。在生成邮件标签时，必须将读者/供应商信息考虑在内。

**Printer Spooler**

管理打印机网络中的打印作业。

局域网为用户提供服务。通常为了吞吐量、可靠性或物理便利性而复制服务。可以以多种不同的方式提供对这些服务的访问，这些方式在队列所在的位置、每个用户需要如何明确地指定服务以及本地故障的后果等细节上有所不同。

网络连接多台计算机和打印机。每台打印机由其中一台计算机驱动，为整个网络提供服务，并配有多个纸盘。在任何计算机上运行的程序都可以为其打印请求指定任何打印机上的任何纸盘。

**Library**

自动化传统图书馆任务，如图书的借入和归还。

库问题很好地服务于正式规范社区[Wing88]。要将其用作软件体系结构问题，我们将关注解决方案的可能结构，而不是功能的规范。

图书馆需要一个信息系统，为图书馆用户和工作人员提供以下在线操作：

1.借阅（或归还）一本书。

2.获取特定作者或特定主题的书籍列表。

3.了解特定借款人当前已退房的书籍（用户只能自己查找）。

4.找出哪位借款人最后一次借出某本书（仅限员工）。

5.记录向图书馆（仅限工作人员）增加（或删除）一本书的情况。

该系统必须能够快速搜索和更新目录（以避免漫长的结账队伍，并使在线图书搜索成为卡片目录的可行替代品），并轻松处理工作人员用户对潜在大型收藏的更新和更正。

系统还必须强制执行以下完整性约束：

a.图书馆中的所有副本都必须可供签出或被签出。

b.不得同时提供和检出任何副本。

c.借阅者一次借阅的书籍不能超过预定数量。

d.一本书的借阅者不能同时借阅一本以上。

**Automated Teller Machine (ATM)**

在远程位置提供常规银行功能。

ATM（自动柜员机）的问题已经出现在几篇论文中。这是Rumbaugh在其关于面向对象设计的书[Rumbaugh91]中提出的问题，如[Lubars92]中所述。

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。每家银行都提供自己的计算机来维护自己的账户并处理针对这些账户的交易。收银台由各个银行所有，并直接与自己银行的计算机通信。人工出纳输入账户和交易数据。自动柜员机与中央计算机通信，中央计算机与相应的银行结算交易。自动柜员机接受现金卡，与用户交互，与中央系统通信以进行交易，发放现金并打印收据。该系统需要适当的记录保存和安全规定。系统必须正确处理对同一帐户的并发访问。银行将为自己的电脑提供自己的软件；您将为ATM和网络设计软件。共享系统的成本将根据持有现金卡的客户数量分摊给银行。

在这种情况下，体系结构必须解决以下问题：

ATM或中央网络控制器中的安全机制应位于何处？

应该如何管理异构系统（每家银行都有自己的软件）？

**Calendar Scheduler**

组织会议日程。

日历管理是计算机中最棘手的问题之一。许多人对此进行了抨击，但迄今为止还没有出现完全令人满意的解决方案。

日历计划程序为许多人维护一致的会议计划。这些时间表至少记录了每次会议的时间、持续时间和参与者。某些会议可能包括日程安排不由日历计划程序维护的人员。会议可以随时添加或删除（直到会议发生时为止），会议参与者也可以添加或删除。会议可安排在所有（或足够）与会者召集的任何时间举行，但某些会议可能需要按特定顺序举行的情况除外。调度器可以维护关于它所服务的人的调度偏好的信息。

**Compiler**

将编程语言的源代码翻译为可执行形式。

编译器将编程语言翻译成机器语言。它们还与其他编程工具（如交互式编辑器和调试器）进行交互。

编译器将程序设计语言中的源代码转换为可以与其他目标代码链接并在计算机上执行的目标代码。

**Mobile Robot**

设计一个能够在监视环境的同时执行任务的移动机器人，例如避开障碍物。

这个问题集中在嵌入式实时系统上。这些系统必须处理外部传感器和致动器，并且它们必须在与其环境中的系统活动相称的时间内做出响应。

考虑移动机器人通常必须完成的以下活动：

获取传感器提供的输入。

控制其轮子和其他可移动部件的运动，

规划其未来的道路。

许多因素使任务复杂化：

障碍物可能会挡住机器人的去路。

传感器输入可能不完美。

机器人可能没电了。

机械限制可能会限制机器人移动的准确性。

机器人可能操作危险材料。

不可预测的事件可能几乎没有时间做出响应。