## 经典软件体系结构案例

## 1、上下文中的关键字 （KWIC）

KWIC 索引系统接受一组有序的行，每行是有序的一组单词，每个单词是一组有序的字符。任何行都可以通过反复删除第一个单词并在行尾附加来"循环转移"。KWIC 指数系统按字母顺序列出所有行的所有圆形移位。

## 2、海浮标

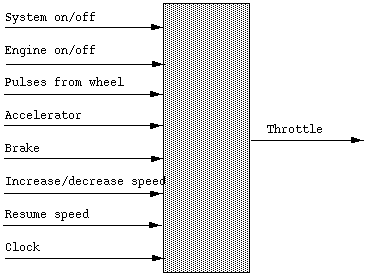
海上浮标支持海上航行。有一个自由浮动浮标的集合，提供导航和天气数据的空中和船舶交通在海上。浮标通过各种传感器收集空气和水温、风速和位置数据。每个浮标可能具有不同数量的风和温度传感器，将来可能会进行修改以支持其他类型的传感器。每个浮标还配备了无线电发射机（广播天气和位置信息以及 SOS 消息）和无线电接收器（接收过往船只的请求）。有些浮标装有红灯，在海上搜索作业中，路过的船只可能会激活红灯。如果水手能够到达浮标，他或她可能会翻转浮标侧面的开关，启动 SOS 广播。每个浮标的软件必须：

* 保持当前风、温度和位置信息;风速读数为每30秒，温度读数每10秒，位置每10秒：风和温度值保持为运行平均值。
* 每 60 秒广播一次当前风、温度和位置信息。
* 应过往船只的要求，广播过去24小时内的风、温度和位置信息：这优先于定期广播
* 根据过往船只的要求激活或停用红灯。
* 水手接合紧急开关后，连续广播 SOS 信号;此信号优先于所有其他广播，并持续到通过的船只重置。

## 3、巡航控制

许多作者已经使用巡航控制来说明软件设计方法。此问题陈述来自用于描述面向对象的编程 [Booch86] 的 Booch，以及伯切诺和卡梅隆后来用来将 JSD 与 OOD 进行比较的一个问题陈述：

当驾驶员打开时，存在巡航控制系统来保持汽车的速度，即使在不同的地形上也是如此。踩刹车后，系统必须放弃速度控制，直到被告知恢复。当驾驶员指示系统时，系统还必须稳步增加或降低速度以达到新的维护速度。



这是此类系统的硬件块  
图。有几个输入：

* 系统开/关：如果打开，表示巡航控制系统应保持车速。
* 发动机开/关：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会处于活动状态。
* 车轮脉冲：车轮的每一次旋转都会发出脉搏。
* 加速器：指示加速器被按下多远。
* 制动器：按下制动器时打开;如果按下制动器，巡航控制系统将暂时恢复为手动控制。
* 增加/降低速度：增加或降低维护速度：仅适用于巡航控制系统是否处于打开时。
* 简历：恢复最后保持的速度：仅适用于巡航控制系统是否处于打开时。
* 时钟：正时脉冲每毫秒。

系统中有一个输出：

* 油门：工程师油门设置的数字值。

## 4、会议裁判

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提交的论文。通常，这样做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）审查小组，然后选择在程序上出现的最佳论文。自动会议裁判的系统应执行以下工作：

1. 项目委员会宣布"要求文件"。

2. 作者收到论文要求，并决定将提交有关其作品的论文。他们写论文并寄给项目委员会。给定论文可能有几个作者，但只有一个答复地址。

3. 方案委员会收到后登记所投稿文件。

4. 在某个时间点，方案委员会向裁判小组分发文件。每篇论文都发给三个不同的裁判，他们不是论文的作者。

5. 方案委员会不断收集裁判的报告。

6. 在某个时间点，方案委员会选择论文列入程序，并通知作者有关选择的信息。这可能涉及从裁判那里获得其他意见。

7. 方案委员会向甄选结果的作者提供建议。

## 5、邮寄列表处理程序

我们都被目录和其他群发邮件的多个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不同时间以不同形式生成的个人原始信息。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。

地址条目包含姓名、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和有关合并变体的指导。首选项更新读卡器/供应商信息。邮件列表是地址条目的集合，或者可能是控制信息。

地址条目可能是原始的（从原始来源（如读者请求）收集），也可以从其他邮件列表中提取。地址条目也可以作为外部邮件列表接收（不一定以所需的格式）。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 令人反感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。它应消除重复的条目并纠正错误。生成邮件标签时，必须考虑读者/供应商信息。

## 6、打印机滑阀

局域网为用户提供服务。通常，这些服务会复制以达到吞吐量、可靠性或物理便利性。这些服务的访问方式多种不同，在诸如队列所在的位置、每个用户需要明确指定服务以及本地故障的后果等详细信息上有所不同。

网络连接多台计算机和打印机。每台打印机均由其中一台计算机驱动，为整个网络提供服务，并配有多个纸托盘。在任何计算机上运行的程序可以指定任何打印机上的任何纸盘以供其打印请求使用。

## 7、图书馆图书管理系统

图书馆问题很好地服务于正式规范社区[Wing88]。要将其用作软件架构问题，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. 查看（或退回）一本书的副本。

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签出的书籍（用户只能查找自己）。

4. 找出哪个借款人上次签出一本书的特定副本（仅限工作人员）。

5. 记录将一本书的添加（或删除）到（从）图书馆（仅限工作人员）。

系统必须能够快速搜索和更新目录（以避免长时间的退房行，使在线图书搜索成为卡片目录的可行替代方案），并轻松处理工作人员用户对潜在大型集合的更新和更正。

系统还必须执行以下完整性限制：

* 图书馆中的所有副本都必须可供结账或退房。
* 不得同时提供和签出任何副本。
* 借款人不能同时结账超过预定的书籍数量。
* 借款人不能同时检查给定书籍的一个副本。

## 8、自动取款机（ATM）

ATM问题在几篇论文中出现。这是问题所在，因为它最初是由伦博在他的关于对象导向设计[伦博91]的书中提出的，如[Lubars92]所述。

设计软件系统，支持计算机化银行网络，包括人力出纳员和自动柜员机（ATM），由银行财团共享。每家银行都提供自己的计算机来维护自己的账户，并处理针对他们的交易。收银站归各家银行所有，并与本行的计算机直接通信。人工出纳员输入帐户和交易数据。自动柜员机与中央计算机通信，中央计算机可以清除与相应银行的交易。自动柜员机接受现金卡、与用户交互、与中央系统通信以执行交易、分发现金和打印收据。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。银行将为自己的计算机提供自己的软件：您将为ATM和网络设计软件。共享系统的成本将根据使用现金卡的客户数量分配给银行。

本案例中的架构必须解决：

* ATM 或中央网络控制器应位于安全机制的位置
* 应如何管理异质系统（每家银行都有自己的软件）的存在

## 9、日历调度器

日历管理是计算的一大难题。迄今尚未出现完全令人满意的解决办法。

日历调度器为许多人保持一致的会议日程。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。有些会议可能包括那些日程安排没有由日历调度员维护的人。会议可随时添加或丢弃（直至会议发生时），会议参与者可添加或删除。会议可安排在为所有（或足够）与会者召开的任何时间举行，但某些会议可能需要按特定顺序举行。调度员可以保留有关其服务人员的日程安排偏好的信息。

## 10、编译器

编译器将编程语言翻译为机器语言。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。

编译器将编程语言中的源代码转换为对象代码，该代码可以与其他对象代码链接并在计算机上执行。

## 11、移动机器人

此问题侧重于嵌入式实时系统。这些系统必须处理外部传感器和执行器，并且它们必须及时响应与系统在其环境中的活动相称的响应。

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。
* 控制车轮和其他可移动部件的运动
* 规划其未来路径。

使任务复杂化的因素很多：

* 障碍物可能会阻塞机器人的路径。
* 传感器输入可能不完美。
* 机器人可能耗尽电源。
* 机械限制可能会限制机器人移动的精度。
* 机器人可能操纵危险物质。
* 不可预知的事件可能几乎没有时间作出回应。