

# 第8章.面向对象分析与 UML建模

---

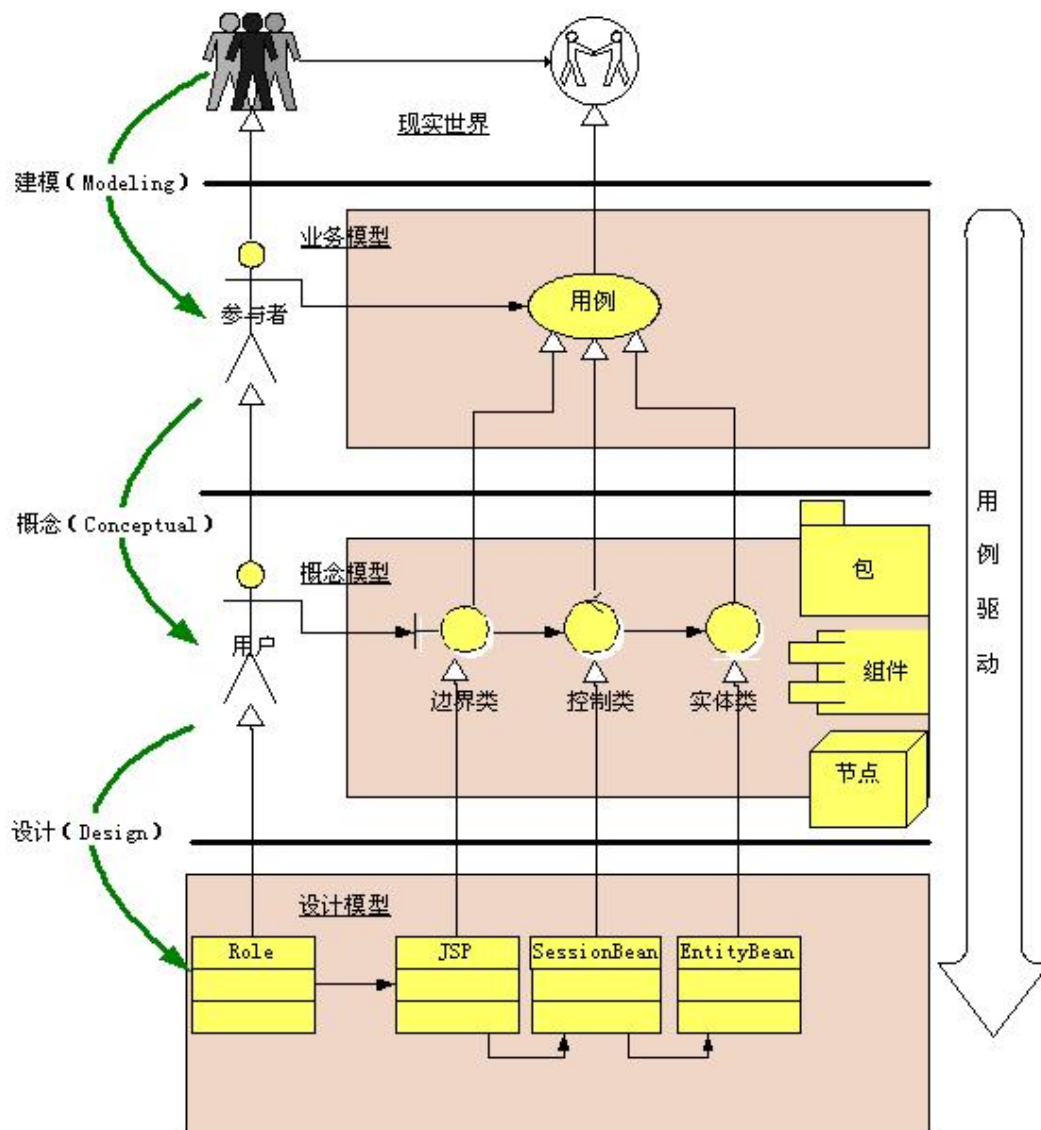
韩锐

北京理工大学 计算机学院

Email: [379068433@qq.com](mailto:379068433@qq.com)  
[hanrui@bit.edu.cn](mailto:hanrui@bit.edu.cn)

# 面向对象分析与UML建模(6)

## ■ 开发时间管理示例



# 确定业务需求

总经理：为什么我们的开发项目进度计划总是那么不准确，延期经常出现，更可恨的是甚至无法给出一个相对比较明确的延迟时间。这样给市场的推广会带来很大的影响，不确定因素使得应对十分困难。

研发经理：唉，这个问题我花了很多时间来解决，但一直收效不好。最初我用 WBS 方法，根据用例包、用例的方式来组织需求，然后将某个用例或子用例作为工作任务分配的开发人员，并指定了相应的完成时间，但到了时间开发人员总是完不成，都反应时间安排不合理。后来，在技术顾问的指导下，改为自底向上的估计方法，任务明确后让开发人员反馈工作量及所需的工作天数。虽然有所好转，但是还是有一些工作任务，开发人员反馈的天数到了，仍然无法完成，甚至无法告诉我要延迟多少天。汇总起来，就形成了这样的结果了。

总经理：这样呀，那有什么好办法呢？

技术顾问：其实问题的关键还是在于“估算”的经验上，对于软件开发而言，实际上没有万能的、准确的估算公式…

（研发经理抢过话题）

研发经理：对对对！我一直在尝试使用 PP、CDCMD 模型来，仍然得不出合理的估计值，真难办。

# 确定业务需求

技术顾问：呵呵，急了！其实估算的基础是经验数据，对于不同的开发人员而言其产能是不一致的，甚至对于相同的开发人员而言，不同的任务所需的时间也是不同的。因此关键在于积累这种经验数据。例如，我在编写技术书籍时，就采用了 PSP（个人软件开发过程）的思路，对所有的工作过程进行了时间的记录，在半年之后，就积累了许多相关的产能数据，现在给编辑的时间承诺总是能够比较的准确。

总经理：哦，难怪你做的承诺都一般很少延误，这种经验能否适用于软件开发的管理呢？

技术顾问：呵呵，这是当然。PSP 是个人软件开发过程，它本来就是为软件开发设计。它是 CMM 的创始人提出的，PSP、TSP 和 CMM 分别针对软件开发员、软件开发小组和软件开发组织。通过 PSP 的贯彻，就一定能够提高软件开发人员的时间安排、时间估算的能力。

研发经理&总经理（几乎同时）：那我们就尝试一下！

技术顾问：哈哈，不过贯彻 PSP 有两个困难。一是开发人员很难适应，每天都要记录自己的工作时间很繁琐，而且产生数据不容易使用；二是时间日志做出来后，管理者会忍不住用来考核开发人员，给他们带来心理压力。

# 确定业务需求

研发经理：那我们可以开发一套软件来帮助他们记录，通过写到数据库中，这样数据的使用问题也就解决了。

技术顾问：对，这就是我的建议。那后者呢？

总经理：我们不考核就是了！

技术顾问：没那么简单！我认为要从以下几点来进行：一是鼓励，鼓励记录时间日志，奖励估算准确的开发人员，从而避免做假时间的情况；二是宣扬，宣扬有效工作时间的概念，我的经验是每个开发人员一天有效的工作时间在 4 个小时之上就是比较好的，树立这种概念能够打消开发人员的顾虑；三是培训，从理论高度建立开发人员执行 PSP 的意识。

总经理：好！我来修订绩效考核，解决鼓励的问题；小陈（研发经理），我配合你树立“每天有效工作 4 小时”的概念；至于培训嘛，那只好拜托你了。

技术顾问：好！没问题。

**为开发人员提供一个PSP工具，简化时间记录工作；  
同时提供数据使用的工具，帮助开发人提高估算能力。**

# 需求捕获

技术顾问：根据我的经验，认为整个系统应该包括以下几个主要的方面。第一，项目及任务安排，由研发经理或项目经理创建项目和任务，开发人员在接到任务后进行估算填写时间计划，研发经理或项目经理对其进行确认。第二，时间记录，开发人员对自己的开发时间进行记录，与任务关联起来。第三，产能分析，研发经理及公司领导可以根据任务和相应的时间记录，来统计公司员工产能数据。

开发人员甲：我认为，开发人员自己应该能够通过这套系统来统计自己的产能数据。

研发经理：那么产能数据怎么表示呢？任务可是不同的呀。

技术顾问：我认为比较合适的是KLOC/天（每天编写的千代码行数）。

开发人员乙：但不同的程序KLOC可能接近，但难度不同所花的时间是不同的。

技术顾问：对，我们可以在每个任务中加上难度系数，产能中的KLOC=实际的KLOC\*难度系数。

研发经理：那么测试任务怎么算？

# 需求捕获

技术顾问：我认为这套系统主要关注的是开发时间、而对于前期的分析和概要设计，以及后续的集成和系统测试等工作可以先忽略，放在系统范围之外，这里只考虑详细设计、编码和相应的测试工作。

研发经理：我明白了，就是对于一个任务而言所花的时间。对，这样比较合理。

开发人员甲：我希望系统能够在让我们填写估算值时，可以查询历史数据，否则仍然没有意义。

开发人员丙：查询历史数据时，还应该有关键词吧！这样我们才能够根据自己将要完成的任务情况找到有参考依据的统计数据。

开发人员乙：还有就是时间记录一定要方便，另外像我们这样经常要在现场开发，如何完成时间记录？

研发经理：可以考虑有一个离线版本的时间记录程序，等回公司连接服务器后再进行数据同步。

.....

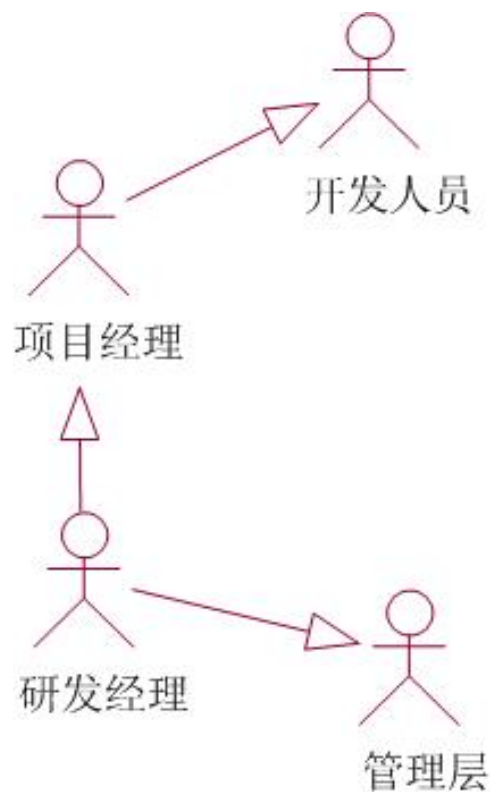
# 获取需求特性表

编号	特性
FEAT01	研发经理能够创建项目、指定或修改项目经理、删除尚未分配工作任务的项目
FEAT02	项目经理可以对项目设置工作包，工作包允许多级嵌套，它只用来组织工作任务
FEAT03	项目经理可以为开发人员指派工作任务，工作任务属于特定的工作包
FEAT04	项目经理在分配工作任务时，能够查阅开发人员的日程安排表，可以按开发人员查询、也可按日程查询
FEAT05	开发人员接到任务时，通过系统填写计划时间（计划开始时间和计划结束时间），项目经理确认后，更新日程安排表
FEAT06	开发人员可以查询相近工作任务的历史数据（估算数据、实际数据）
FEAT07	开发人员任务执行将超计划时，应报告项目经理，项目经理通过系统更新其日程表
FEAT08	当任务完成之后，项目经理负责Close任务，并填入实际的完成情况（KLOC、实际结束时间）
FEAT09	开发人员可以随时记录自己的时间，提供“开始计时”、“暂停计时”、“停止计时”，在停止时，填入任务编号（在线则选择）、工作关键字（以逗号分隔的多个），自动生成开始时间、暂停时间、停止时间、总时长、有效时长（总时长-中断时长）
FEAT10	开发人员可以根据任务编号、关键字、起止时间进行分类组合查询与统计
FEAT11	时间记录程序会自动连接服务器，完成时间日志上传的工作，未能连接服务器，则在本机暂存时间日志
FEAT12	项目经理可以按项目、任务、关键字统计实际工作时长、产能
FEAT13	研发经理及管理层可以按个人、任务、项目、关键字查看工作时长、统计产能





# 建立用例模型—识别参与者



# 建立用例模型—合并特性获得用例

参与者	特性	用例
开发人员	FEAT05.开发人员接到任务时，应通过系统填写计划时间（计划开始时间和计划结束时间），项目经理确认后，更新日程安排表	UC01.填写任务计划
	FEAT06.开发人员可以查询相近工作任务的历史数据（估算数据、实际数据） FEAT10.开发人员可以根据任务编号、关键字、起止时间进行分类组合查询与统计	UC02.查询历史任务数据(UC01的扩展)
	FEAT09.开发人员可以随时记录自己的时间，提供“开始计时”、“暂停计时”、“停止计时”，在停止时，填入任务编号（在线则选择）、工作关键字（以逗号分隔的多个），自动生成开始时间、暂停时间、停止时间、总时长、有效时长（总时长-中断时长） FEAT11.时间记录程序会自动连接服务器，完成时间日志上传的工作，未能连接服务器，则在本机暂存时间日志	UC03.记录时间日志

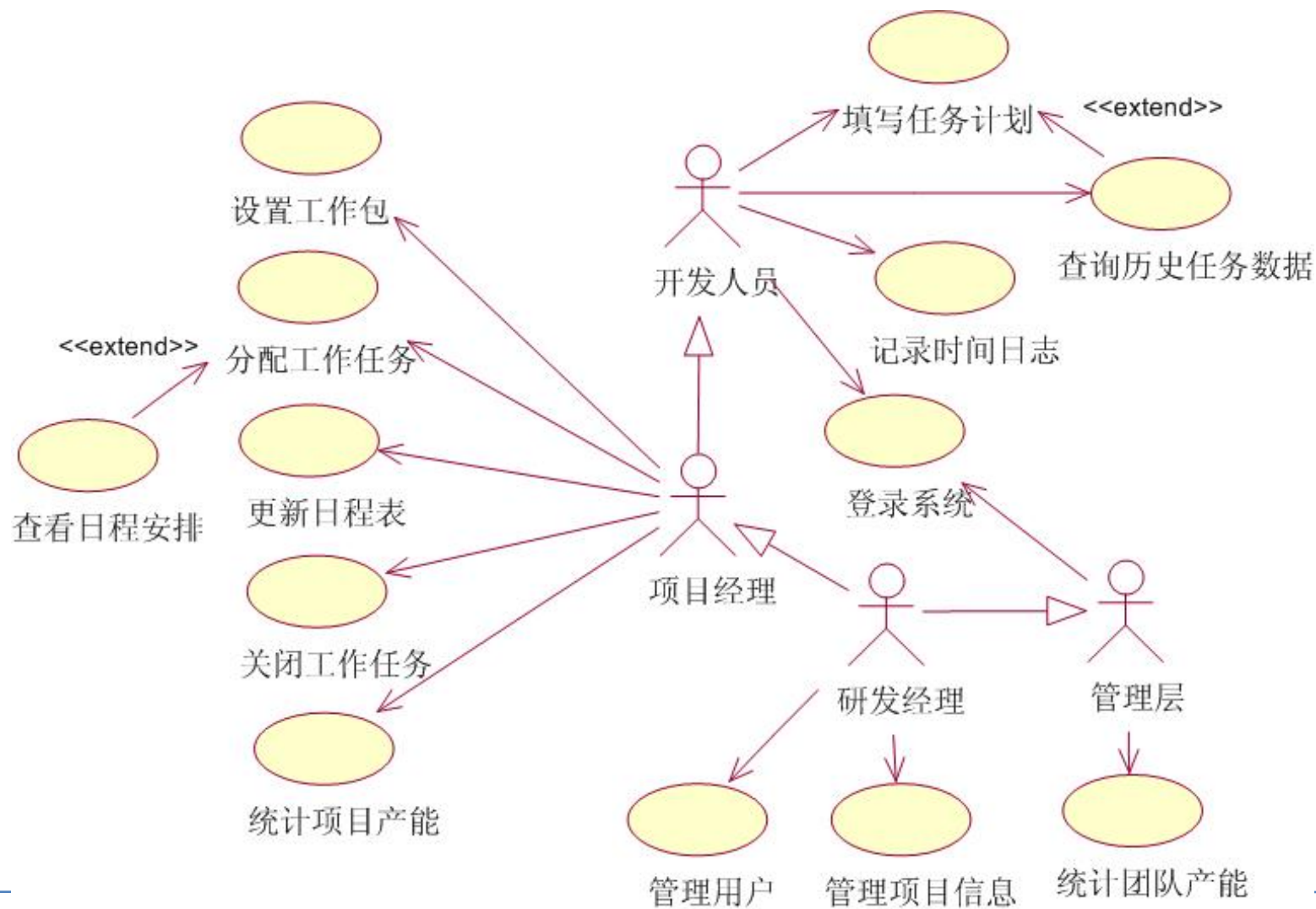


# 建立用例模型—合并特性获得用例

项目经理	FEAT02.项目经理可以对项目设置工作包，工作包允许多级嵌套，它只用来组织工作任务	UC04.设置工作包
	FEAT03.项目经理可以为开发人员指派工作任务，工作任务属于特定的工作包 FEAT04.项目经理在分配工作任务时，能够查阅开发人员的日程安排表，可以按开发人员查询、也可按日程查询	UC05.分配工作任务 UC5A.查看日程安排 (扩展用例)
	FEAT07.开发人员任务执行将超计划时，应报告项目经理，项目经理通过系统更新其日程表	UC06.更新日程表
	FEAT08.当任务完成之后，项目经理负责Close任务，并填入实际的完成情况（KLOC、实际结束时间）	UC07.关闭工作任务
	FEAT12.项目经理可以按项目、任务、关键字统计实际工作时长、产能	UC08.统计项目产能
研发经理	FEAT01.研发经理能够创建项目、指定或修改项目经理、删除尚未分配工作任务的项目	UC09.管理项目信息
管理层	FEAT13.研发经理及管理层可以按个人、任务、项目、关键字查看工作时长、统计产能	UC10.统计团队产能



# 建立用例模型—绘制用例图



# 建立用例模型—简要描述用例

用例编号	UC01
用例名称	填写任务计划
用例概述	开发人员对项目经理安排给自己的工作任务进行计划，填入计划开始时间和计划完成时间。
主参与者	开发人员
补充说明	在填入计划开始时间和计划完成时间时，开发人员可以查询与该任务的关键字相关的历史任务的数据。



# 建立用例模型—详细描述用例

用例编号	UC03	
用例名称	记录时间日志	
用例概述	开发人员可以随时记录自己的时间，提供“开始计时”、“暂停计时”、“停止计时”等功能，在停止时，填入任务编号（在线则选择）、工作关键字（以逗号分隔的多个），自动生成开始时间、暂停时间、停止时间、总时长、有效时长（总时长-中断时长）。	
主参与者	开发人员	
前置条件	用户进入“记录时间日志”程序	
后置条件	将本次时间日志存入数据库	
基本事件流	步骤	活动
	1	系统显示“开始”、“暂停”和“停止”按钮，但仅“开始”可用
	2	用户点击“开始”，系统记录开始时间，并将“开始”置为不可用，使“暂停”和“停止”按钮可用
	3	用户点击“停止”按钮，系统记录停止时间，并统计暂时时间、暂停次数、总时长、有效时长，并要求用户选择任务编号、输入工作关键字和相关信息。填写完成后，点击确定，用例完成。
扩展事件流	3a	在此期间，若用户点击“暂停”按钮，系统则记录暂停开始时间，并使暂停次数增加1次，并使“暂停”按钮变为“恢复”，使“停用”按钮不可用
	3a1	当用户点击“恢复”按钮，用当前时间减去暂停开始时间得到本次暂停时间，并累加到“暂停时间”时间中，并使“恢复”按钮变为“暂停”，使“停用”按钮恢复可用
规则与约束	时间记录程序应以离线式工作，该程序会自动连接服务器，完成时间日志上传的工作，如果未能连接服务器，则在本机暂存时间日志	



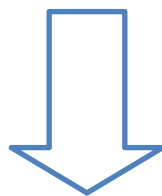
# 建立用例模型—划分用例优先级

优先级	用例	说明
1	UC11.登录系统	系统使用的基础，并且可复用原有资源
	UC09.管理项目信息 UC04.设置工作包 UC05.分配工作任务 UC01.填写任务计划	任务管理的完整流程，是记录时间日志的基础
	UC03.记录时间日志	系统核心功能
2	UC07.关闭工作任务	只是对任务信息进行更新，重要性次之
	UC06.更新日程表 UC5A.查看日程安排	对日程安排进行优化，使任务安排合理化
3	UC02.查询历史任务数据 UC08.统计项目产能 UC10.统计团队产能	对系统记录的时间记录进行有效的利用，必须有前面的信息才能够开发
	UC12.管理用户	前期可以通过直接往数据库中写值的方式进行使用，最后提供界面操作即可



# 建立类模型—发现类

研发经理	项目	项目经理	工作任务	工作包
开发人员	日程安排表	计划时间	历史数据	估算数据
实际数据	任务编号	工作关键字	开始时间	暂停时间
停止时间	总时长	有效时长	服务器	产能
管理层	时间日志			

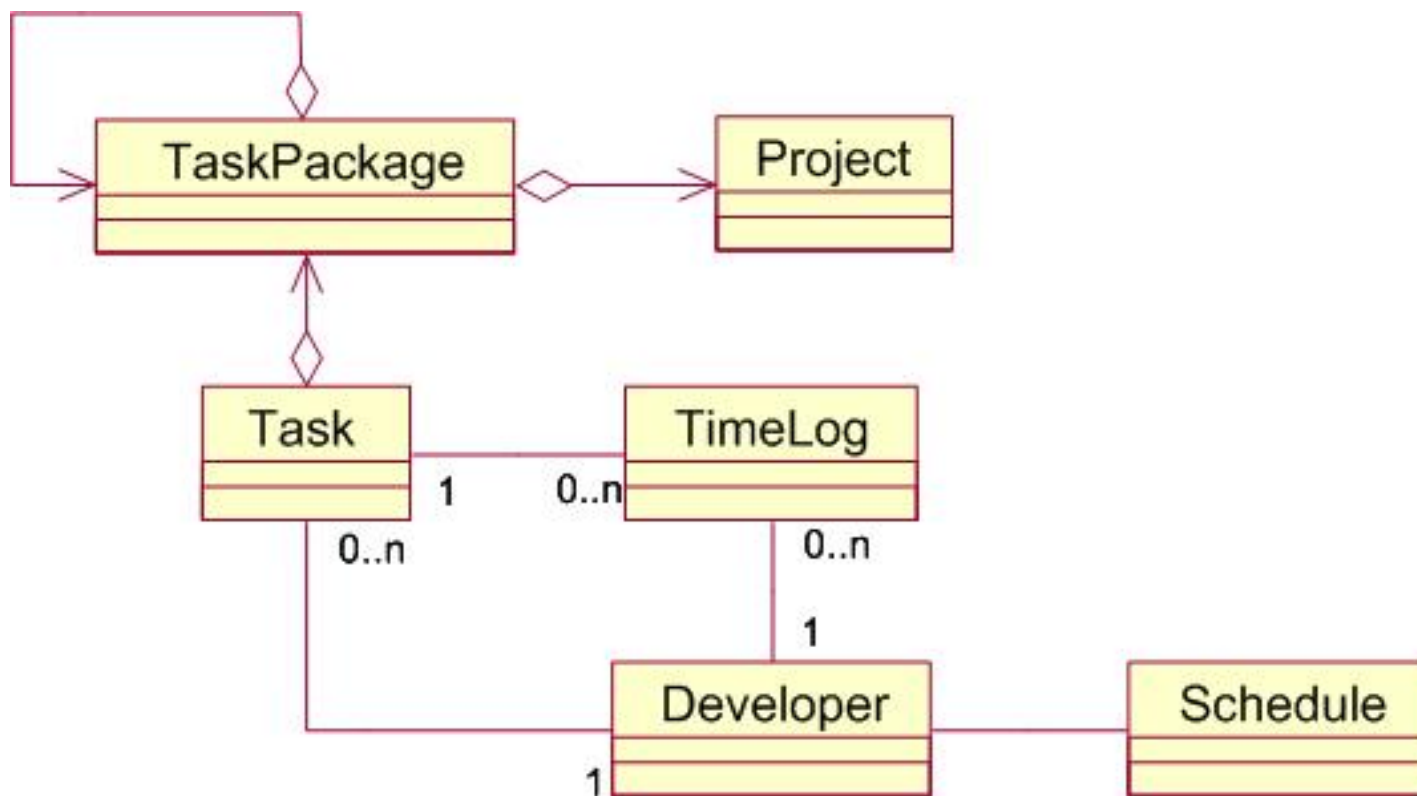


项目	工作任务	工作包	开发人员
日程安排表	时间日志		

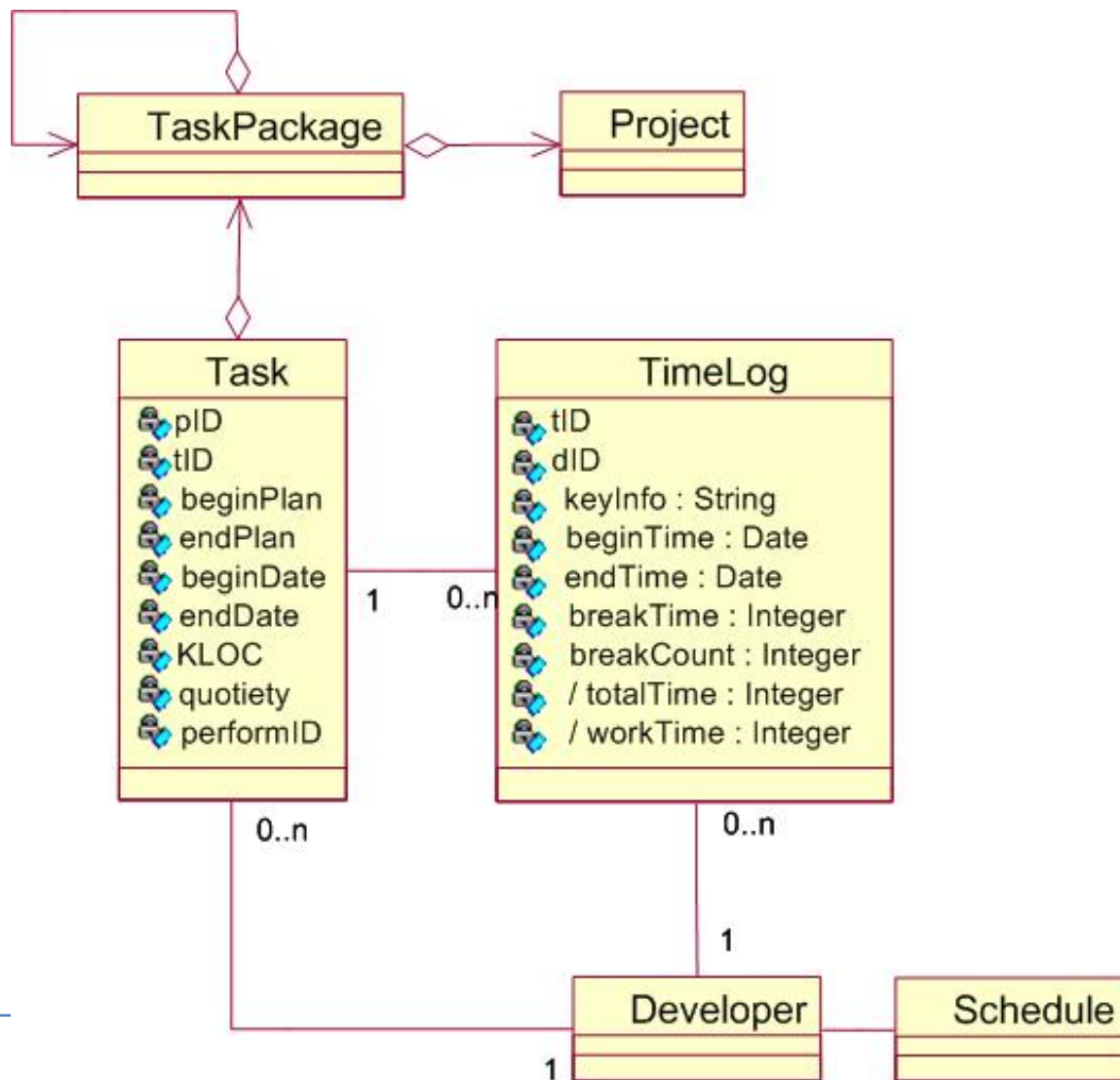




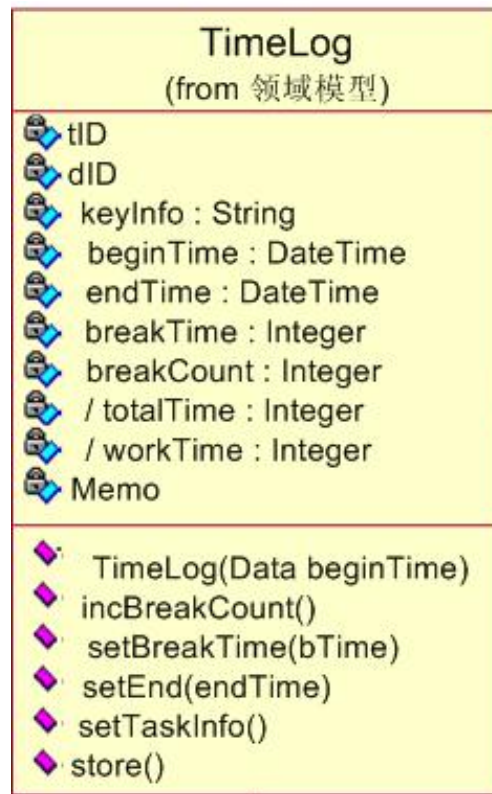
# 建立类模型—类关联分析



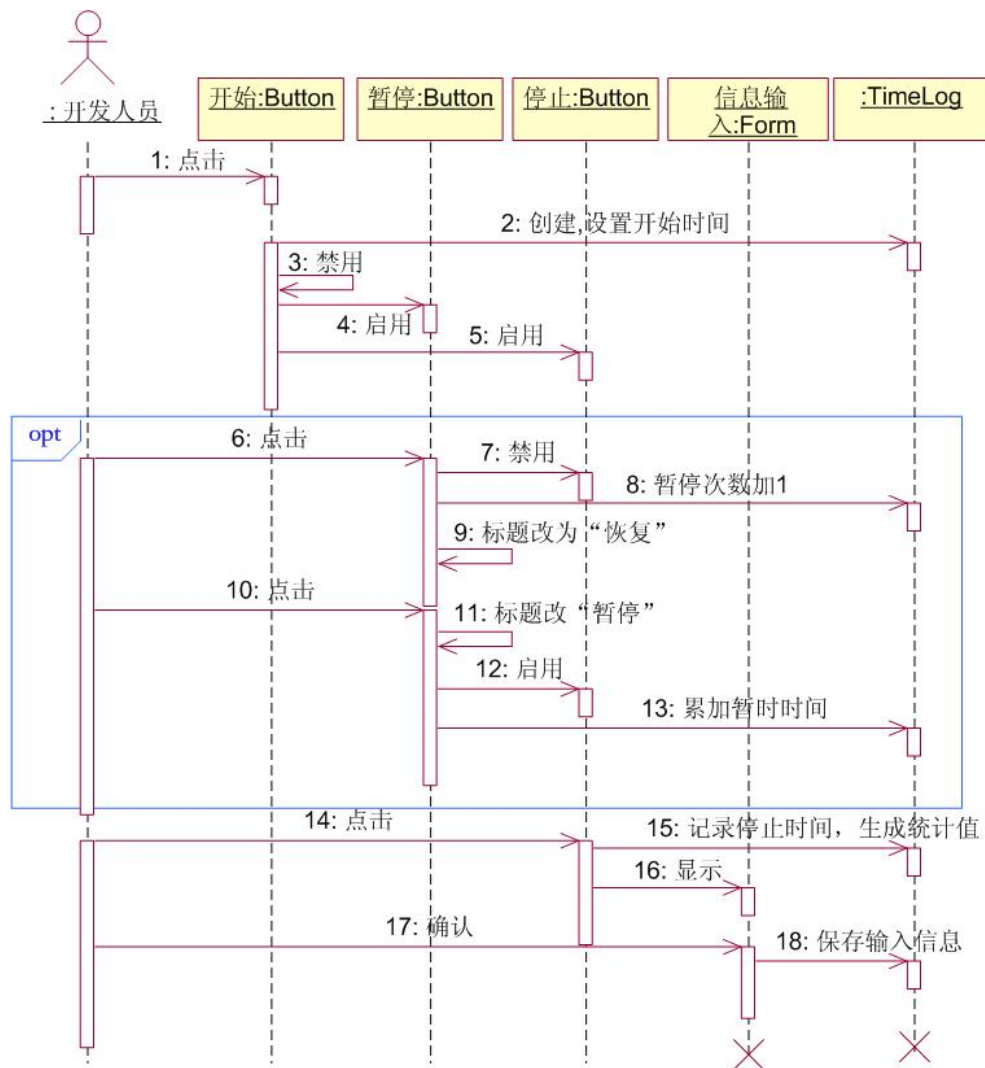
# 建立类模型—细化类的属性



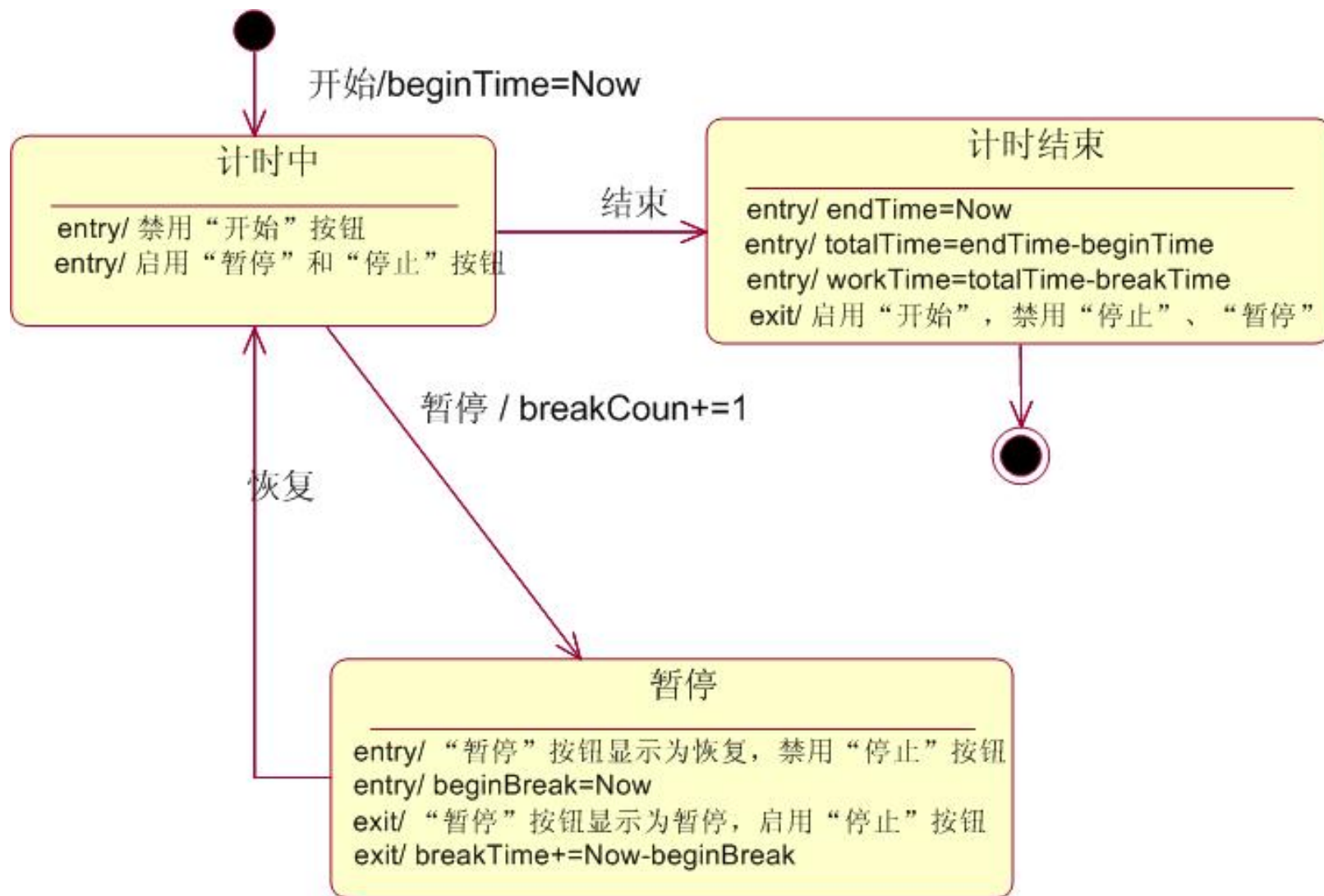
# 建立类模型—细化类的操作



# 建立交互图



# 建立状态图



# 建立状态图&用户界面设计

