

实战 OO：鲁棒分析

徐锋 / 文

www.cnki.net

曾经有人指出结构化分析技术存在一个致命的弱点，其在分析模型和系统的设计模型之间，存在着一个本质性的断层。这个断层导致系统构想与实现设计时产生了分离。而在面向对象系统中，是否也存在这个现象呢？从理论上说，OO技术是可以将系统中几乎独立的各个方面链接成一个语义上的整体。

但在许多 OO 的实践中，却也感受到了这种分析与设计的鸿沟，当完成了用例建模之后，要开始绘制顺序图或协作图，至而演化成为设计类图，总是感到有些力不从心。其实存在着一个方法，它可以有效地填补这个鸿沟，那就是 Ivar Jacobson 在 1991 年引入的 OO 世界的 Robustness 分析，中文译做鲁棒分析、健壮性分析。

虽然“Robustness 分析”这一名字中带有分析一词，但准



确地说它有着很多与设计类似的工作，只是它并不能够算是正式的设计，因此被归到分析阶段中。也许是由于简单易用，也许是由于它通常不被文档化保存，只是绘制成草图的形式，现在在 RUP 中、在很多 OO 技术书籍中，好像越来越少看到它的出现。但是希望这一切不要让你与这个强大的工具擦肩而过，学会它吧，它将成为你强大的秘密武器。

什么是鲁棒分析

在和大家一起运用鲁棒分析技术继续完成我们的任务之前，还是一起来了解一下什么是鲁棒分析？有了一些基本的概念和认识，才能够有效地应用它。

1 鲁棒性分析的作用

鲁棒性分析具体来说有什么用呢？它是如何帮助大家跨越分析与设计之间的鸿沟呢？在 OO 分析与设计中，它能够完成以下几个关键任务：

1) 正确性检查：鲁棒分析将通过比顺序图更简单、更有效的图形来描绘用例中的传递过程，从而确保用例是正确的，而且没有指定对于特定对象来说不合理、不可能的系统行为。而且，熟练掌握鲁棒分析，甚至可以成为编写用例描述时结合使用的一种有效方法。

2) 完整性检查：通过鲁棒分析，你可以很容易地找到用例描述中所有必须的扩展路径。

3) 持续发现对象：在做鲁棒分析时，你将开始思考具体的细节，从而很容易发现在域建模时遗失的对象，而这些对象在绘制顺序图时不容易被重新找到，而只会让你感觉根本无法绘制出相应的顺序图。

4) 对象确定：类似的，我们可以发现对象，也可以确认对象，也就是能够很容易地发现不正确的类。

5) 初步设计：正如后面会提到的，鲁棒分析将类分成了实体对象类、控制对象、边界对象三种，这种观点正好与 MVC 体系结构不谋而合，这样可以保证设计出更加稳健的系统。

2 鲁棒性分析的元素

1) 实体对象：通常来自域模型也就是现实世界，用来描述具体的实体，通常映射到数据库表格与文件中。

2) 控制对象：主要用来体现应用程序的执行逻辑，将其抽象出来，可以使得变化不影响用户界面和数据库中的表。

3) 边界对象：通常是用来完成参与者（用户、外部系统）与系统之间交互的对象，例如 Form、对话框、菜单、接口等。

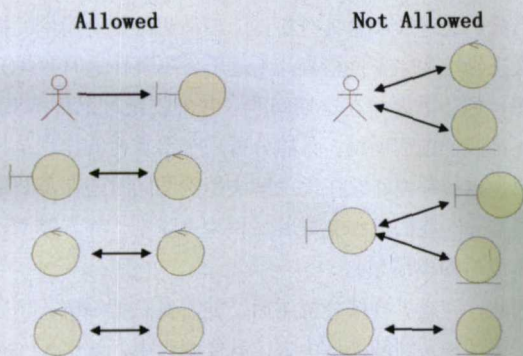
鲁棒分析为这三种元素赋予了一个比较直观的图形表示法（图1）。



另外，鲁棒分析也制定了一些相关的规则，透过这些规则，你将更加深入地理解鲁棒性分析的本质内容：

- 1) 参与者只能通过边界对象与系统交互；
- 2) 边界对象只能与控制对象或参与者交互；
- 3) 实体对象也只能与控制对象交互；
- 4) 控制对象可以与边界对象、控制对象、实体对象交互，但不能与参与者交互。

图表二更加清晰地表达了这一规则。



图二 鲁棒分析图规则

3.1 基本事件流

- 1) 图书管理员向系统发出“新增书籍信息”请求；
- 2) 系统要求图书管理员选择要新增的书籍是计算机类还是非计算机类；
- 3) 图书管理员做出选择后，显示相应界面，让图书管理员输入信息，并自动根据书号规则生成书号；
- 4) 图书管理员输入书籍的相关信息，包括：书名、作者、出版社、ISBN 号、开本、页数、定价、是否有 CDROM；
- 5) 系统确认输入的信息中书名未有重名；
- 6) 系统将所输入的信息存储建档。

3.2 扩展事件流

- 5a) 如果输入的书名有重名现象，则显示出重名的书籍，并要求图书管理员选择修改书名或取消输入；
- 5a1) 图书管理员选择取消输入，则结束用例，不做存储建档工作；
- 5a2) 图书管理员选择修改书名后，转到 5)

图三 鲁棒分析图规则

进行鲁棒分析

好了，现在大家对鲁棒性分析已经有了一个大致的了解，接下来，我们将回到了我们的工作——“个人图书管理系统”的开发中来。

1 从事件流开始

鲁棒分析是针对某个具体用例而言的，而其分析的基础是具体的用例描述，准确地说，是用例描述中关于事件流的部分。而且在分析时，还必须综合考虑基本路径和扩展路径两方面。在上一讲中，我们曾以“新增书籍信息”（用例 UC01）为例说明了如何编写用例描述，曾经得到了如图3所示的事件流描述。

2 寻找边界对象

在我们讨论的这个用例中，不存在外部系统，而且参与者也相对比较简单，只有一个，即“图书管理员”，因此要寻找边界对象只需以“图书管理员”这个参与者为线索，从用例描述的每一句去选择：

1) 图书管理员向系统发出“新增书籍信息”请求——图书管理员在什么地方向系统发出“新增书籍信息”的请求呢？通常我们会设计一个主窗口，并在上面摆放一些按钮来实现，因此我们在此句描述中可以发现两个边界类：主窗口、“新增书籍信息”按钮。

2) 系统要求图书管理员选择要新增的书籍是计算机类还是非计算机类——从这句中我们可以发现一个新的边界类：书籍类别列表框。

3) 图书管理员做出选择后，显示相应界面，让图书管理员输入信息，并自动根据书号规则生成书号——从这句中，我们可以发现最关键的一个边界类：“新书信息录入”窗口以及辅助的“提交”按钮。

而且，我们还可以发现新增书籍信息按钮是组成主窗口的一部分。书籍类别列表框、提交按钮则是组成新书信息录入窗

口的一部分。

3 找出相关的实体对象

在第一讲域建模中，我们就已经找到了与领域相关的类，这些就是系统中的实体对象。因此我们通过发现用例描述中用多少域模型中的概念类，来列举出实体对象。很显然，这里涉及到的实体对象，包括书籍、计算机书籍、非计算机书籍以及书籍列表四个。

4 “战前”准备

当我们找到了边界对象和实体对象之后，就可以从边界对象开始，根据用例描述逐渐地找到控制对象，完成鲁棒分析图。通常对于每个用例而言，比较适合的控制对象的数量大约在2-5个之间。另外，还要明确两点：

1) 在绘制时一定要注意鲁棒图中的箭头并不是和顺序图一样代表消息传递，而是代表一种逻辑流。这时你要做的只是理清对象之前的逻辑关系，而不是要生成一个严密定义设计方案。

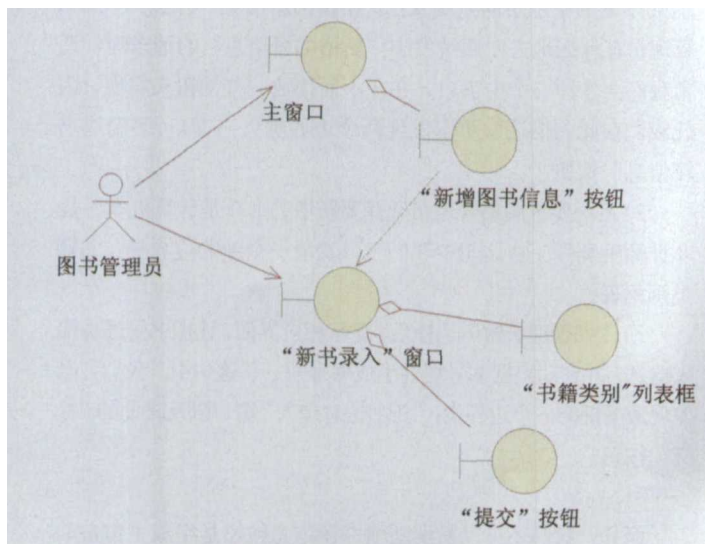
2) 在鲁棒分析图中出现的控制对象并不一定今后都将被作为一个设计类出现，其有可能会只是一个类中的成员方法。因此在绘制的时候不要过多地陷入“是否值得作为一个类”的判断中。

绘制鲁棒分析图的最好工具就是纸和笔，因为这样会给绘制者提供大量的自由空间。另外，鲁棒分析过程的重要性远大于分析的结果，因此使用CASE工具正式地绘制并没有太大意义。而在本文中为了给大家呈现一个更加清晰的结果，使用了Rational Rose。

5 千里之行的第一步：绘制边界对象

首先，我们可以将参与者和边界对象绘制出来，如图四。

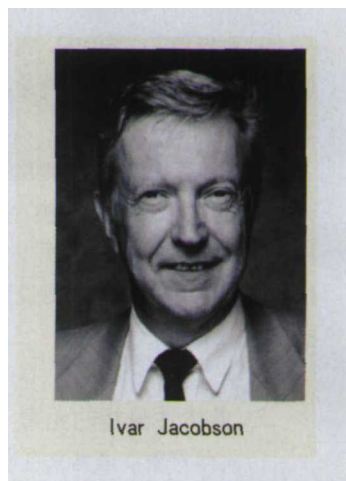
要说明的是，在鲁棒分析的规则中，明确地说明边界对象是不能够与边界对象进行交互的，而本例中却在“新增图书信



图四 鲁棒分析图(1)

息按钮”和“新书录入窗口”两个边界类之间有一个虚箭头连接起来，这是为什么呢？这又表示什么呢？

其实，这里的连接是说明“新书录入窗口”是“按下新增图书信息按钮时才显示出来的，其实其中可以引入一个“创建新书录入窗口”的控制对象，但写过程序的人都很清楚，这仅仅是一句程序就解决的问题，因此就省去了这个控制对象。



6 逐渐引入控制及实体对象

我们根据事件流中的步骤5，以及扩展路径的描述，就可以在原图上增加相应的控制对象，得到更进一步的鲁棒分析图（如下页图五）。

从图五中，我们表达的意思是，当按下提交按钮之后，就将进行书名的重复性检查（依据为书籍列表），如果重名将回到新书录入窗口重新输入，如果不重名就创建新的书籍，并加入书籍列表。

到此为止，我们还没有对步骤2、3的内容进行考虑，也就是根据所选的书籍类别，自动按规则生成书号，并作为创建书籍的相关信息，因此我们在图五的基础上再做一些修改，得到最终的鲁棒分析图（如下页图六）。

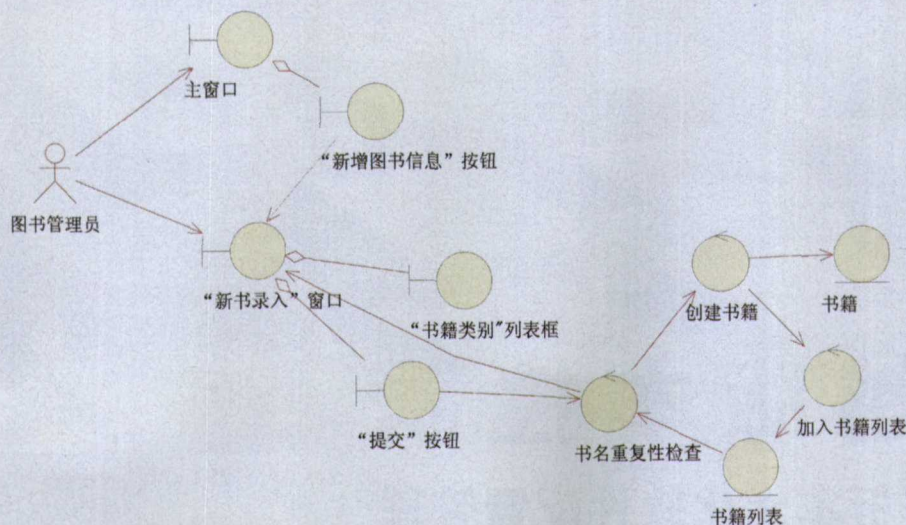
通过细节分析，并通过鲁棒分析图绘制出来，将会使得开发者对于系统细节了解得更加清楚。并且还找到了相关的界面信息（边界对象）和关键的业务逻辑（控制对象）。再次强调的是，在鲁棒分析阶段不要花费过多的时间，否则会陷入一种分析崩溃的陷阱。其实，也可以考虑在编写用例描述的同时，就结合鲁棒分析来启发思路，也是可行的一种办法。

鲁棒分析图画完之后

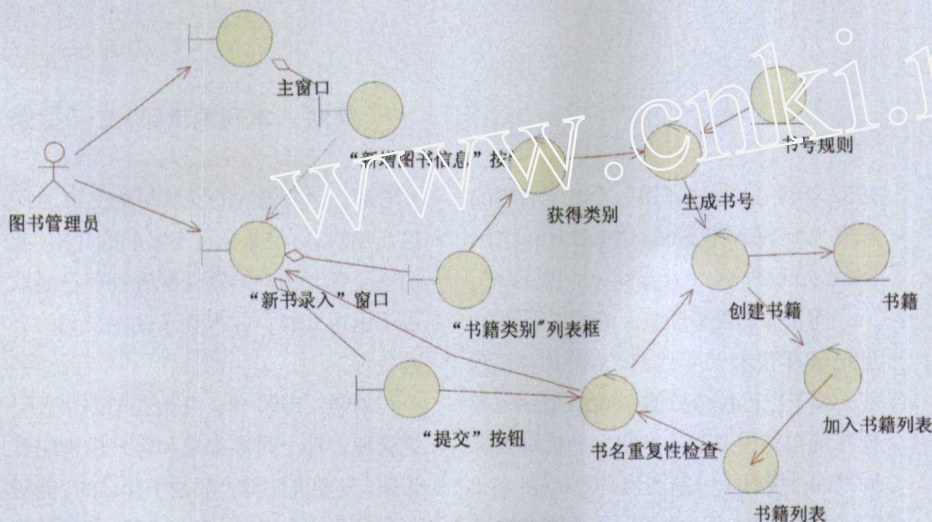
当我们通过绘制鲁棒分析图，理清思维，对用例的描述做了一次正确性和完善性检查。不过千万不要在这时就抛开鲁棒分析图，你还需要将在这期间的“意外收获”记载下来。

正如我们前面所说，鲁棒分析一个很重要的作用是持续发现对象，也就是说，它在设法解决用例的同时还在不断地细化静态模型——我们已经完成的域模型（注意，静态模型还包括设计完成后的类模型）。

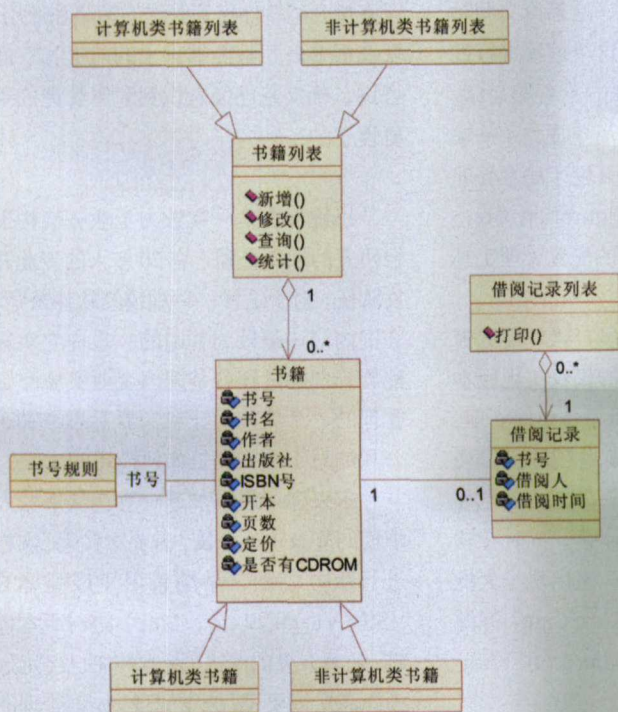
因此，完成了鲁棒分析图后，就需要总结一下，将发现的新类、属性加入到域模型中去。就像图六中，就引入一个新的实体对象——书号规则，这就是我们在域建模阶段遗漏的类。



图五 鲁棒分析图 (2)



图六 鲁棒分析图 (3)



图七 更新后的域模型



徐锋，高级程序员，系统分析师，CSAI顾问。自1992年开始就迷恋神奇的魅力深深吸引，一发不可收拾。从Apple II的汇编语言开始入门的我，一直就在计算机软件领域孜孜不倦、兢兢业业，不敢有一分懈怠。十余年来，涉猎面广泛，只叹自己仍博而不精。现致力于需求工程、系统分析与设计、过程改进等领域的研究与探索。

我们应该将其补充到域模型中去。如图七。

另外，我们还可以从用例描述中发现，书籍列表包括了两个：计算机类书籍列表、非计算机类书籍列表。并根据用例描述将书籍的属性做了一些更新。而书号规则则是与书籍关联的，关联的联系在于“书号”。

结语

在本讲中，我们在用例模型的基础上，利用鲁棒分析技术做了进一步分析，帮助我们对用例模型的正确性、完善性进行了检查，并且从中发现了一些新的对象/类，以及相关的成员属性，这些信息又对域模型的完善提供了支持。

不过，鲁棒性分析虽然很有用，但是当我们完成了分析之后，这些中间结果无需保留，它只是帮助大家理清更多细节，以便于绘制出更加形式化，更加贴近编码的交互图（包括顺序图、协作图）。而关于如何绘制交互图，我们将在下一讲中详细说明。