共同服务于患者;在饭店,服务员、厨师共同服务于顾客·····基本上所有的企业活动都是如此。另外,这不仅适用于某个独立的企业,还适用于零售业、批发业、制造业、物流业等企业的连锁活动(供应链管理,Supply Chain Management, SCM)。

使用通信图来表示现实世界中的职责分配的情形,如图 8-12 所示。一般来说,在使用通信图表示现实世界的情况下,实例就是人或组织。仅从形状上来看,该图与前面的图 8-6 是一样的,但实际表示的内容是完全不同的。

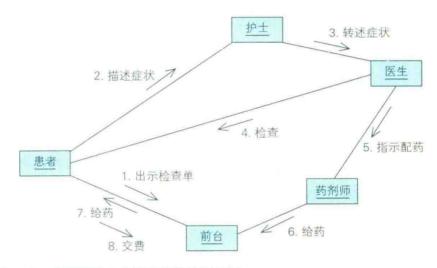


图 8-12 表示医院工作情形的通信图示例

其中一个不同之处就是箭头表示的信息。在用通信图表示程序结构的情况下,箭头表示消息调用,而在表示现实世界的情况下,则是通过对话等进行沟通的意思。人们之间的沟通与方法调用的情况不同,可以采用对话、留言、肢体语言等各种形式。即使是相同的信息,具体的表达方式也取决于当事人。

更大的区别是,现实世界中的工作不会像图 8-12 那样每次都按照同样的方式进行。在处理现实世界中的工作时,人们会随机应变,所以工作的顺序和内容经常会发生变化,比如提前进行某些工作,或者特意省略某些细节等。因此,图 8-12 表示的模型只是一种典型模式。

不过,绘制图 8-12 那样的典型模式也是有意义的。通过绘制该图,现

实世界中模糊的工作情形就会变得一目了然,这样也就更便于对当前课题 和改善对策等进行讨论。

另外,使用通信图表示的内容也可以使用时序图来表示。比如,用时序图重新表示图 8-12,结果就如图 8-13 所示。

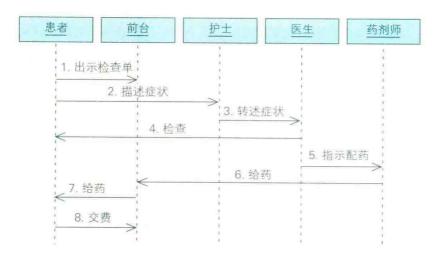


图 8-13 表示医院工作情形的时序图示例

到这里为止,我们就介绍完了表示作为归纳整理法的集合论和职责分 配的图形。与前面一样,我们也来简单地总结一下这三种图形的特征。

< 类图 >

表示根据集合论进行分类整理的现实世界的事物之间的关系。

< 时序图 >

将进行了职责分配的人或组织协作完成整个工作的情形表示为 时间序列。

<通信图>

将进行了职责分配的人或组织协作完成整个工作的情形以结构 为中心进行表示。

8.10 UML 的使用方法之三:表示非面向对象

最后,我们来介绍一下 UML 表示面向对象无法表示的信息时的使用方法。对于认为 UML 就是面向对象的人来说,这可能会让人感觉有些意外。

表示集合论和职责分配的归纳整理法的概念非常强大,能够表示各种信息。不过,即便如此,面向对象也不是万能的,于是,一些之前使用的面向对象没有覆盖到的图形表示就被加入到了 UML 中。

这里就来介绍一下其中具有代表性的用例图、活动图和状态机图。

8.11 使用用例图表示交给计算机的工作

用例图用于明确表示计算机的工作范围。具体来说,就是确定对象系统和外部(用户或其他系统)的界限,简洁地表示交给计算机的工作内容。

用例(use case)是用例图的中心,意思是"实际使用的例子",这里指计算机提供给用户的功能。

UML入门书中经常会将用例图放在开头,于是很多人都认为用例是面向对象或 UML 特有的技术。但其实该思想在 UML 出现之前就已经存在,绝不是什么新奇的概念。

用例图的示例如图 8-14 所示。大长方形表示系统边界,内部是用例,描述了系统提供的功能,外部是用户或其他关联系统。

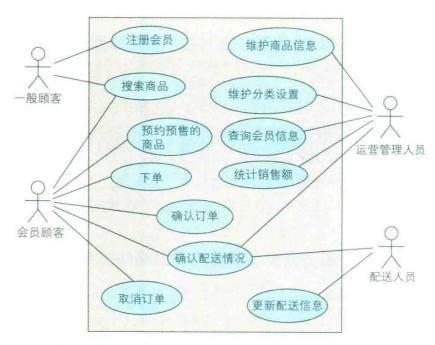


图 8-14 用例图示例

该图非常简单,可以说其最大的好处就是容易理解,另外,对粗略地 掌握系统全貌也非常有用。

8.12 使用活动图表示工作流程

接下来介绍一下活动图。

活动图是面向对象出现之前就已经在使用的流程图的发展形式。

传统的流程图主要用来描述程序的逻辑。而在使用 OOP 编写应用程序的情况下,由于大多以较小的单位来创建类和方法,所以通常一个方法中编写的算法都很简单。因此,使用活动图表示算法的情况并不是很多。

取而代之的是,活动图经常被用来表示现实世界的业务流程(图 8-15)。 虽然现实世界的工作流程也可以使用前面介绍的时序图和通信图来表示, 但在分析实际的工作情况时,相比人物的职责分配,理解整体流程可能更 为重要,因此,使用便于人们直观理解流程的活动图就非常方便。

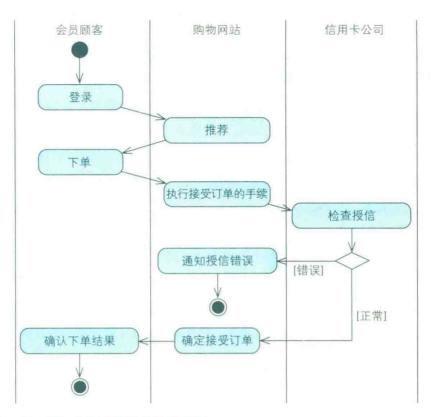


图 8-15 表示业务流程的活动图示例

8.13 使用状态机图表示状态的变化

最后为大家介绍状态机图。

状态机图是通信等控制系统软件中从过去就一直在使用的图形表示。 如果说状态迁移图,可能有人就会觉得比较熟悉。

状态机图表示事物状态根据外部事件变化的情形,可以用来表示用OOP编写的实例的状态迁移、系统整体的状态迁移等各种对象。图 8-16 是表示购物网站订单的状态迁移的状态机图示例。

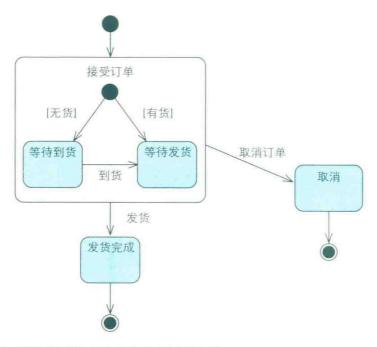


图 8-16 表示订单状态迁移的状态机图示例

我们对 UML 的图形的介绍就到这里。最后,我们来总结一下表示面向对象无法表示的信息的三种图形的特征。

<用例图>

表示交给计算机的工作范围。

<活动图>

表示现实世界的工作流程。

< 状态机图 >

表示外部事件导致的状态变化。

8.14 弥补自然语言和计算机语言缺点的"语言"

在本章最后,我们来思考一下为什么 UML 会被称为语言。

提到语言,大家应该立刻就会想到我们平时使用的自然语言。软件世界中有 Java、C 和 COBOL 等编程语言,以及 XML、HTML等标记语言。编程语言和标记语言的目的都是被读入到计算机中,使计算机进行工作,所以这里统称为计算机语言。

下面,我们通过将 UML 与自然语言和计算机语言进行比较,来思考一下 UML 的目的。

自然语言用于人们交流的会话和文档,使用声音和字符来表示。虽然 其语法是固定的,但允许省略表示,也存在方言和流行语,自由度比较高。

计算机语言是人们用来指示计算机执行某些作业。实际上,计算机只可以解释机器语言的命令,但人们理解起来非常费劲,因此,计算机语言是高级内容,使用字符来表示。不过,由于计算机是死板的机器,所以语言规范的定义必须严谨,在实际使用时,也必须严格遵守其语法。

与自然语言一样,UML也是用于人们之间的交流。但UML与其他两种语言不同,最大特征就是使用图形进行表示。其他两种语言的基础都是字符,当表示的对象极其复杂时,信息量也会变得很庞大。而UML是图形表示的,只是取出庞大信息中的重要部分,表示简洁,便于直观理解。

这些内容汇总在表 8-2 中。

表 8-2 三种语言的比较

语言	自然语言	计算机语言 (编程语言、标记语言)	建模语言 (UML)
目的	人们之间的交流	向计算机指示作业	人们之间的交流
形式	声音、字符	字符	图形
特征	有基本语法,但详细语法 比较宽松。相同语言中也 允许方言	极其严格	重视直观理解

国 虽然 UML 的中心是图形表示,但同时也规定了以文本形式进行记述的 OCL (Object Constraint Language,对象约束语言)。

UML 可以说是为了弥补自然语言和计算机语言的缺点(图 8-17)。因此,熟练使用的技巧只是一种辅助手段。相比于一些极端的使用方法,如仅使用 UML 来表示所有的规格说明,或者使用时序图和活动图来表示程序的所有逻辑等,UML 只是帮助理解文档和程序的一种手段,大家轻松开始使用之旅吧。

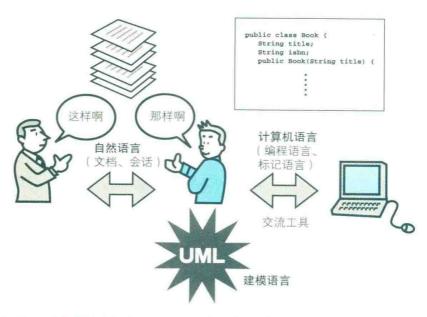


图 8-17 建模语言弥补自然语言和计算机语言的缺点

深入学习的参考书籍

[1] オージス総研オブジェクトの広場編集部. その場でつかえるしっかり 学べる UML2.0[M]. 东京: 秀和システム, 2006.

好好好

该书用大量例图全面讲解了UML2中的所有图形,我们可以将其作为参考手册使用。此外,书中每个讲解项目都被分为初级、中级和高级三个阶段,直至介绍到开发流程等内容,结构非常精炼。

[2] Martin Fowler. UML 精粹:标准对象建模语言简明指南(第3版)[M]. 潘加宇,译.北京:电子工业出版社,2012.

T 1/2

该书介绍了 UML 的绘制方法和使用方法,此外还总结了基于建模实践的技术窍门。作为开发流程的实践指南,书中还介绍了模式和重构等面向对象技术的主题。

[3] 井上树. ダイアグラム別 UML 徹底活用 [M]. 东京: 翔泳社, 2005.

分分

该书介绍了UML中具有代表性的图形的使用方法。通过在书中设置概要、用途和注意事项等内容,以通俗易懂的方式总结了使用各个图形的 诀窍和注意事项。



本章的关键词

第章

建模、业务分析、需求定义、业务应用程 序、嵌入式软件

建模:

填补现实世界和软件之间的沟壑

热身问答

在阅读正文之前,请挑战一下下面的问题来热热身吧。



使用 "seamless" (无缝的) 一词表示的面向对象的优点是下面哪一项?

- A. 由于现实世界的情形被直接反映到软件结构中, 所以软件维护很 轻松
- B. 使用 UML 描述成果,可以消除相关人员之间的沟通成本,有效 推进开发
- C. 通过将许多功能作为类库提供,即使语言规范升级,也无须再次 编译程序
- D. 由于"以对象为中心"的思想可以适用于各种领域,所以可以统一开发方法



A. 由于现实世界的情形被直接反映到软件结构中, 所以软件维护 很轻松

英文"seamless"是"无缝的"的意思。由于当应用面向对象时,现实世界的情形被直接反映到软件结构中,所以业务分析、需求定义和设计等各个工程中并无本质区别,能够无缝、顺利地推进开发。在介绍这些内容时,经常会使用"seamless"一词。另外,还有一种情况下也经常使用该词,那就是当现实世界的业务发生变化时,由于能够马上确定软件的修改位置,所以维护变得比过去轻松得多。

然而,实际上,软件并不会完全替换现实世界的工作,现实世界和OOP结构也是"似是而非的"。因此,"seamless"一词表示的效果实际上根本不会存在。不过,如果限定于业务应用程序的数据结构,那么该说明还是比较贴切的。关于这一点,本章会为大家详细讲述。

本章 重点

本章的主题是建模。计算机是为了让人们变轻松而承担了现实世界中的一部分工作的机器。不过,正如本书前面介绍的那样,现实世界和软件之间存在很大的沟壑。为了填补这道沟壑,我们需要进行3个阶段的工作,即整理现实世界情形的"业务分析"确

我们需要进行 3 个阶段的工作,即整理现实世界情形的"业务分析"、确定交给计算机的工作范围的"需求定义"和定义软件结构的"设计",而使用了 UML 的建模就是顺利推进这些工作的技术。

本章将介绍业务应用程序和嵌入式软件的建模示例。让我们通过这些示例,来一起思考一下计算机担当什么性质的工作,以及现实世界的情形反映到了软件的哪一部分上。

🧧 9.1 现实世界和软件之间存在沟壑

本书前面一直强调,现实世界和软件表示的世界之间存在沟壑。第2章中介绍过,面向对象并不是直接将现实世界表示为软件的技术。第7章中还介绍过,计算机只是承担了人们的一部分工作,并不会完全替换现实世界。因此,当开发软件时,我们需要填补这道沟壑。

另一方面, 计算机是为了让人们变轻松而承担了现实世界中的一部分 工作的机器。虽然存在沟壑, 但是管理计算机的软件也应该会以某种形式 反映现实世界的情形。

本章将介绍填补现实世界和软件之间沟壑的 3 个阶段的工作,以及顺利推进这些工作的建模技术。这里将重点介绍前 2 个阶段的业务分析和需求定义,即整理现实世界的情形,并定义交给计算机的工作。关于设计,我们将在第 10 章中介绍。

另外,本章还将介绍业务应用程序和嵌入式软件这两种性质存在很大 不同的应用程序的建模示例,并通过这两种应用程序来重新审视一下现实 世界和软件之间的关系。

9.2 计算机擅长固定工作和记忆工作

在介绍建模之前,我们先来思考一下计算机擅长什么样的工作。

人工智能曾经掀起过几次热潮。虽然推理机、专家系统和自然语言处理等许多领域都取得了一定研究成果,但是很遗憾,能够匹敌人脑的计算机仍未出现。据说这种计算机至少也得几十年后才能出现。

不过以现在的技术水平,有一些工作是人类无论如何也赶不上计算机的,那就是固定工作和记忆工作(图 9-1)。

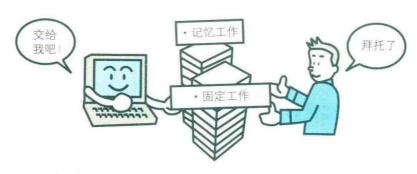


图 9-1 计算机擅长固定工作和记忆工作

计算机擅长固定工作和记忆工作。

第一种是固定工作。计算机如实且飞快地执行程序中的命令。只要软件没有 bug,硬件没有故障,就绝不会出错。另外,只要条件相同,那么无论执行多少次,结果都一样。计算机不会对工作感到厌倦,也不会抱怨。而如果让人们来做大量的单调作业,那么一定会存在很多疏漏吧。

计算员工工资、计算银行存款利息、计算证券交易所的股票成交额等, 都属于固定工作。在计算机出现之前,这些工作都是人们使用算盘手工计 算的。而现在如果没有计算机,那简直无法想象。在迅速、准确地执行这 些工作方面,计算机远比人优秀。 计算机擅长的另一种工作是记忆工作。现在的计算机硬盘都是 GB 或 TB 单位。得益于这么大容量的存储设备,我们可以准确地存储庞大的信息,并随时取出。另外,只要设备没有故障,计算机就绝不会忘记所记忆的内容。

在计算机出现之前,这些信息都是记录在纸上。考虑一下百科词典的 例子就能发现,从存储空间、查找速度、备份的便捷性等方面来说,计算 机的存储介质都要远比纸张优秀。

如上所述, 计算机在固定工作和记忆工作方面拥有绝对的实力。而人 们并不擅长这两项工作, 因此, 利用计算机, 人们能够变得轻松很多。

9.3 通过业务分析、需求定义和设计来填补沟壑

在确认了计算机擅长的工作之后,现在我们来介绍一下填补现实世界 和软件之间沟壑的相关内容。

计算机承担了现实世界中的一部分工作,而管理计算机的是软件。不过,如果我们得到的是"希望使用计算机让医院或者银行的业务变轻松"这种笼统的要求,那是无法立刻开始编写程序的。

在现实世界中,人们在确定自己的职责之后,也会根据情况随时调整,或者即使是同一件工作,不同人的做法也稍微存在区别。另外,人们还会犯错,也会生病或者受伤。

虽然计算机可以承担现实世界中的一部分工作,但是它擅长的是固定 工作和记忆工作,是一个死板的机器。因此,我们必须从现实世界的工作 中选出可以交给计算机的工作。

而编写管理计算机的软件也是一件非常辛苦的工作。仅保证它能正确运行就已经很难了,还必须考虑使用方便、运行效率高、可维护性强以及易于扩展等。在有几十万行代码的大规模系统的情况下,还需要分配工作,确保多名开发人员能够顺利地进行团队开发。

基于这些,在推进软件开发时,需要进行下述3个阶段的工作。