

第1阶段(业务分析)：整理现实世界的工作的推进方法

第2阶段(需求定义)：确定交给计算机的作业范围

第3阶段(设计)：确定软件的编写方法

首先是**业务分析**。在业务分析阶段，需要整理好如何对现实世界中的工作进行职责分配，以及如何推进工作。除此之外，还应提取出业务相关的课题，并以此为信息依据来确定交给计算机的工作。业务分析是整理为什么(Why)使用计算机。

接下来是**需求定义**，即定义可以交给计算机的现实世界中的工作。由于计算机擅长记忆工作和固定工作，所以我们需要从现实世界的工作中选出这些工作。需求定义相当于确定让计算机干什么(What)。

最后是**设计**。要在规定时间内高质量地完成大规模的软件是非常辛苦的，因此，在开始编程之前，我们需要充分讨论并定义软件结构，以使多名成员能够有效地展开工作。设计就相当于确定管理计算机的软件如何(How)实现。

像这样，编程之前的工作大致可分为上述3个阶段。

这3个阶段的工作完全不同，但是都很重要。如果业务分析和需求定义不充分，那么最终肯定无法编写出对用户有用的系统。如果设计有所欠缺，那么就难以完成系统，即使侥幸完成了，之后的维护和功能扩展也难以推进。

9.4 建模是顺利推进这3个阶段的工作的技术

实际上，在面向对象出现之前，业务分析、需求定义和设计这3个工作就已经在做了。那么，面向对象对这3个工作有什么帮助呢？答案就是“建模”。

建模(modeling)就是“创建模型”的意思。英文“model”(模型)有塑料模型、汽车模型等“模型”，以及“时装模特”“为理解复杂现象而简化的理论和假设”等含义，而这里的建模是指使用UML，用二维图形来表

示软件功能和内部结构。使用 UML 的建模通常被称为**面向对象建模**。不过，正如第 8 章中介绍的那样，UML 中还包含一些与面向对象并无直接关系的图形，因此，本书中只是称为“建模”。

在建模中，即使是同一个应用程序，业务分析、需求定义和设计的成果也不一样。这是因为 3 个阶段的目的各不相同，创建模型的观点也不一样。填补现实世界和软件之间沟壑的这 3 个阶段也可以看作从直接表示现实世界的观点切换为结合计算机的情况进行考虑的观点。

< 建模的目的 >

业务分析：直接把握现实世界的情形。

需求定义：考虑计算机的性质，确定让计算机承担的工作范围。

设计：考虑硬件性能、操作系统和中间件的特性以及编程语言的表现能力等，确定软件结构。

9.5 应用程序不同，建模的内容也不一样

虽然简单地称为建模，但是根据应用程序性质的不同，其推进方法和要点也有很大差别。一般来说，常用的应用程序大致可以分为如下几类。

● 业务应用程序

企业等的业务活动中使用的系统。诸如出货、订货、库存管理、制造业的生产管理、银行的账目系统、会计和人事等各种系统。购物网站也可以归为此类。

● 嵌入式软件

管理电器及各种设备的软件。因为软件是在嵌入装置的 CPU 上运行的，所以这样命名。

● 单机应用程序

是指在个人计算机或便携式终端等上面运行的软件。比如电子邮件、

浏览器、文字处理软件、电子表格软件、进度管理软件和游戏软件等都属于单机应用程序。之前单机应用程序大多只在客户端环境中运行，但近来通过网络与服务器通信的应用程序也不断增多。

除此之外，还有在底层支持这些应用程序的基础软件。

● 基础软件

诸如 Windows 和 Linux 等操作系统，以及管理数据库处理和通信控制的中间件等。

根据这些软件的类别的不同，业务分析、需求定义和设计这 3 个阶段的工作的推进方法及创建的模型也有很大不同。

接下来，我们将介绍这 3 个阶段的工作的推进方法和所创建的模型示例，这里以最具有代表性的业务应用程序和近年来备受关注的嵌入式软件为例进行说明。

9.6 业务应用程序记录现实中的事情

首先来介绍一下业务应用程序。

所谓**业务应用程序**，就是支持企业等的业务活动的软件。典型的业务应用程序有出货、订货、库存管理、会计和人事等系统，有时也称为事务处理系统。说现在的企业活动是靠业务应用程序支撑的也毫不为过。

这些应用程序之前都构建在大型机或办公计算机上，但是现在，使用 JavaEE 和 .NET 技术，在网络环境下运行这些应用程序的情况也变普遍了。

可以说业务应用程序最重要的工作就是记录现实中的事情。企业的业务工作中会与许多客户进行诸多交易。在计算机出现之前，这些信息都是记录在账本等纸面上，而现在通常都是使用计算机进行管理。在纸面上管理这么庞大的信息是非常辛苦的，而如果使用计算机，即使是几百万、几千万的交易信息，也都可以准确记录，并随时取出。在该领域，擅长记忆工作的计算机能够充分发挥其本领。

在很多情况下，业务应用程序并不只是记录信息，还会基于记录的信息进行计算。例如，计算工资、计算银行利息、核对账单和进款等。在大部分情况下，这些计算处理中使用的只是小学高年级学生都能够理解的加减乘除等简单计算。在引入计算机之前，这些工作都是办公人员使用算盘手工进行的，而现在则大多交给了计算机。

9.7 对图书馆的借阅业务进行建模

下面我们来看一下建模示例。这里以我们身边的图书馆为例进行介绍。由于和使用者并无金钱交易，所以相比出货、订货和库存管理等，图书馆的借阅业务比较简单，但也具备业务应用程序的基本特征。

我们先从业务分析开始。业务分析需要表现人们在引入计算机之前进行的工作的情况。由于现在一般都使用计算机，所以对于已经使用计算机进行的工作，很多情况下还希望进一步提高工作效率。在这种情况下，在整理既有业务时也会用到计算机。不过，这里我们以全部都是手工作业为前提进行介绍。

我们使用第8章中介绍过的活动图来表示工作流程。图书馆的工作包括图书的借阅和预约、图书的采购和报废、盘点等，这里以最常见的借阅业务为例进行介绍。用活动图表示借阅业务，如图9-2所示。该图虽然很简单，但是能够让我们一目了然地了解图书借阅工作（对于用户来说可能是娱乐而不是工作）是怎样进行的。

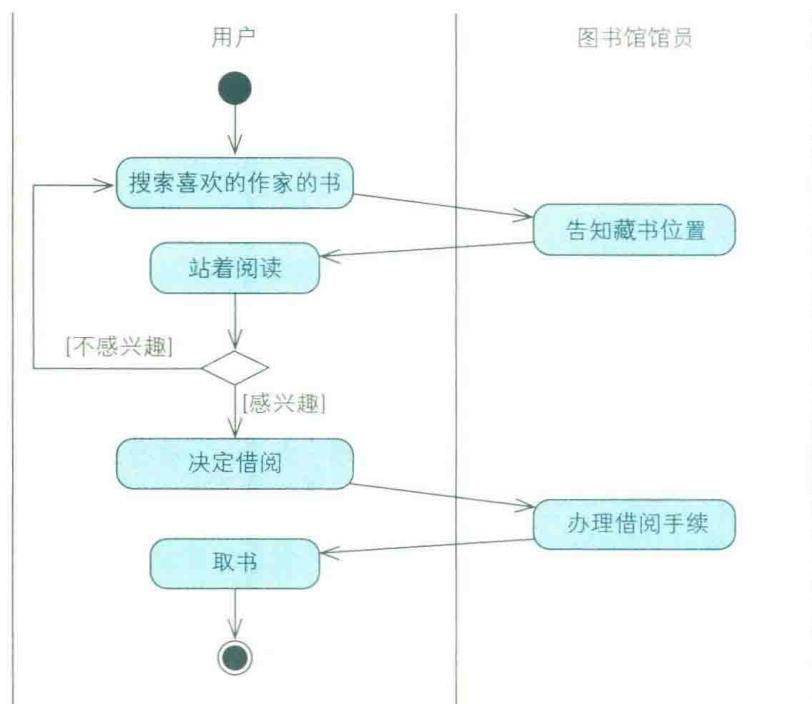


图 9-2 图书馆的借阅业务流程

我们基本上都知道图书馆的图书借阅流程，因此，即使不画该图，或许也能够确定交给计算机的工作。不过，在企业业务或律师、税务人员等我们不熟悉的工作的情况下，活动图可以帮助我们理解实际的工作情形。即使是图书馆这种谁都熟悉的例子，通过画活动图来直观地表示整体的模型，也有助于推进对业务课题或改善要点的讨论。

一般来说，业务分析是一边聆听相应工作的负责人（在该示例中为图书馆馆员）的介绍一边进行的。不过，这里并不是毫无目的地聆听，通过边听边画如图 9-2 所示的图形，就可以避免在讨论时遗漏要点，从而总结成容易理解的成果。对于图 9-2，即使不是计算机专家，也能够轻松理解其含义。如果在展开讨论时充分利用这种图，用户就会不断提出现状中的不便之处，以及希望在新系统中实现的建议等。这些在确定新系统功能时都是重要的信息来源。建模能够促进人们之间的交流。

9.8 使用用例图来表示图书馆业务

接下来是需求定义。需求定义是查看图 9-2 的活动图，找出计算机擅长的记忆工作和固定工作，从而定义交给计算机的工作。

使用用例图，可以简洁地表示交给计算机的工作。以图书馆系统为例，其用例图如图 9-3 所示。该图也很简单，非常容易理解，在与计算机专家之外的人进行交流时，就可以使用该图。通过一边画用例图一边讨论，就能够涌现出许多意见，比如谁使用系统、提供什么样的功能以更好地服务于用户等。

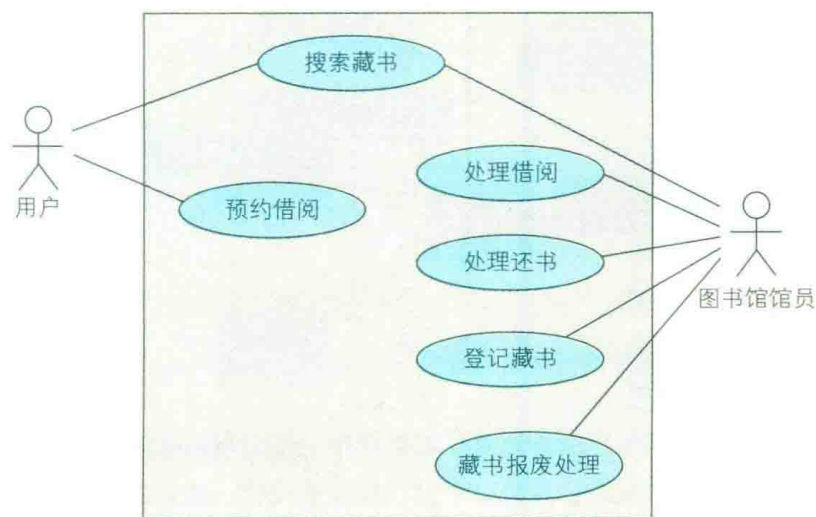


图 9-3 图书馆系统的用例图

在确定计算机的工作范围之后，我们再来改写前面在进行业务分析时画的活动图（图 9-2）。引入计算机之后的借阅业务流程的活动图如图 9-4 所示。最右边增加的列 [在 UML 中称为泳道 (partition)] 表示计算机承担的工作。从图 9-4 中可以看出，即使在引入计算机之后，许多工作还是需要由人来完成。

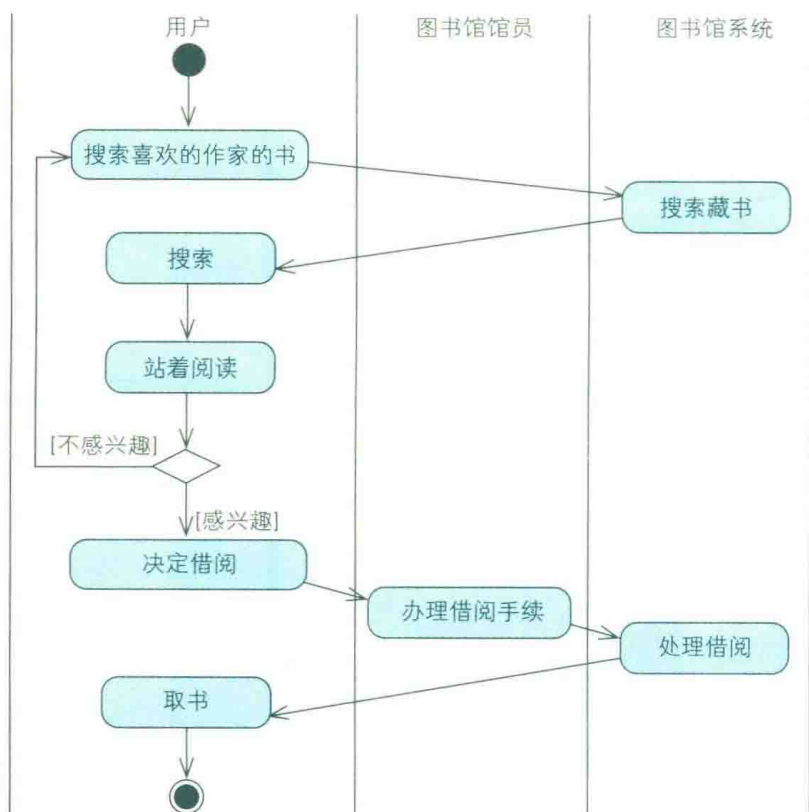


图 9-4 引入计算机后的借阅业务流程

通过将藏书和用户信息记录在计算机中，我们可以把大部分搜索藏书和处理借阅的工作交给计算机。不过，像站着阅读、决定借书和取书等工作是无法交给计算机处理的。从该示例中也可以看出，业务应用程序通常只将现实世界中的一部分工作交给计算机。

最近，随着电子书的迅速普及，站着阅读、取书等业务也逐渐有计算机参与进来。不过，即使计算机能为人类提供推荐和朗读等服务，最终确定要读什么书及怎样理解这本书的内容也还是人类的工作。因此，计算机并不能完全替换现实世界。

9.9 用概念模型表示图书馆系统的信息

在业务应用程序的需求定义阶段，我们还应该绘制一个重要图形，那就是类图。类图可以表示集合与其元素之间的关系，被用于表示所管理的信息的结构。将系统管理的信息在结构上表示出来的图形称为**概念模型**。由于大部分业务应用程序使用数据库来管理信息，所以该图也就表示数据库的结构。

图书馆系统的概念模型示例如图 9-5 所示。

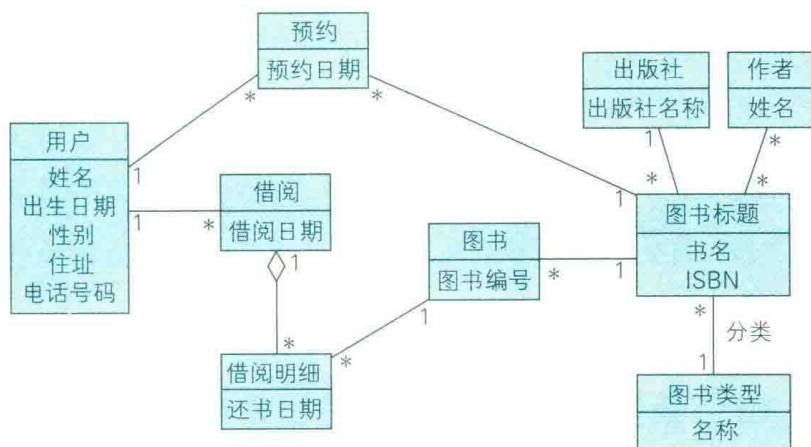


图 9-5 图书馆系统的概念模型

下面，我们用文字来描述一下该图表示的内容。

- 管理姓名、出生日期、性别、住址和电话号码等用户的信息
- 由于会收藏多本相同标题的书，所以分别将书的类型称为“图书标题”，将一本实实在在的书称为“图书”
- 给每本图书都加上图书编号
- 将图书标题分为小说、专业书籍、实用书籍、参考书籍、童书和写真集等类型进行管理

- 除了书名、ISBN 之外，图书标题中还管理出版社和作者信息
- 用户可以预约多个图书标题
- 用户一次可以同时借阅多本图书
- 同时借阅的图书可以分开归还

怎么样？与图 9-5 相比，大家可能感觉文字说明更容易理解。不过，该图逻辑性地表示出了图书馆系统中应该管理的信息。我们可以非常轻松地根据该图来设计关系型数据库的模式（schema）。另外，这里描述的都是很自然的内容，但为了更好地总结该图，我们需要一个一个地确认业务规则。比如，经常被看漏的“图书标题”（类型）和“图书”（一本实实在在的书）的区别等在画图过程中也能够发现。

与表示程序结构时一样，相比使用一维的文字来描述，在使用二维图形表示时，由于将图形和内容放在一起，所以更容易记忆。根据笔者的经验，即使在很久之后再次看到图形，也能够立刻想起和用户讨论时的场景以及当时印象深刻的发言内容等。第 8 章中也介绍过，图形有助于人们记忆。

9.10 业务应用程序中只有数据是无缝的

到目前为止，我们介绍了业务应用程序在需求定义阶段的成果，即用例图和概念模型（类图）。这里我们再来重新看一下这两种图形，就会发现业务应用程序的特征。

我们先来看一下用例图。这个用例图表示了借阅、藏书的登记和报废等图书馆中的业务。不过，虽然我们使用计算机实现了这些功能，但是现实世界中还是存在一些必须由人来完成的工作。即使在计算机上进行图书借阅、藏书的登记和报废处理，也还是需要将实际的书借给用户，以及将书摆在书架上、实际进行报废等。计算机只是记录现实世界的事情，以供之后参考。像这样，许多用例都提供信息的输入和引用功能，这是业务应用程序的一般特征。其依据是，大部分用例的名称中带有“登记”“刷

新”“删除”“维护”“搜索”“查询”等词语。

接下来，我们看一下概念模型（类图）。概念模型用来表示系统中应该记录的信息。比如，图 9-5 中表示了用户、图书、作者等现实世界中存在的人和物，借阅和预约等事情，以及它们之间的关系。在支撑企业工作的业务应用程序中，存在该企业的客户、顾客、商品、合同、交易内容和公司职员等重要信息。

用例图只是表示信息的输入和引用，而概念模型则能够表示现实世界中的人和物、发生的事情。这正是业务应用程序的特征。计算机的主要工作是将现实世界中的事物和事情作为信息进行记录，并搜索这些信息，而判断和交涉、商品和货款的交付等实际工作依然由人在计算机外部的现实世界中进行^①。

大家知道 **seamless** 一词么？该词是“无缝的”的意思，在说明“面向对象直接将现实世界的情形反映到软件中”时，经常会使用该词。根据这一说明，当应用面向对象时，由于能够知道现实世界和软件结构的对应关系，所以在现实世界发生改变的情况下，能够立马确定软件的修改位置。实际上，正如本书中反复强调的那样，现实世界与软件表示的内容和结构都有很大不同，因此，问题并不是那么简单。

不过，如果只是讲数据结构，那么现实世界和软件几乎就是无缝的（图 9-6）。如果结合数据结构来设计访问数据库的程序，那么仅就这一部分而言，现实世界可以直接反映为软件。

① 不过，关于金钱交易，也有不少系统只根据计算机上的记录进行处理，而不直接接收实际的纸币或硬币。由于金钱本质上并不是有形的东西，而是“看不见的价值”，所以容易用计算机来替换现实世界中的工作。

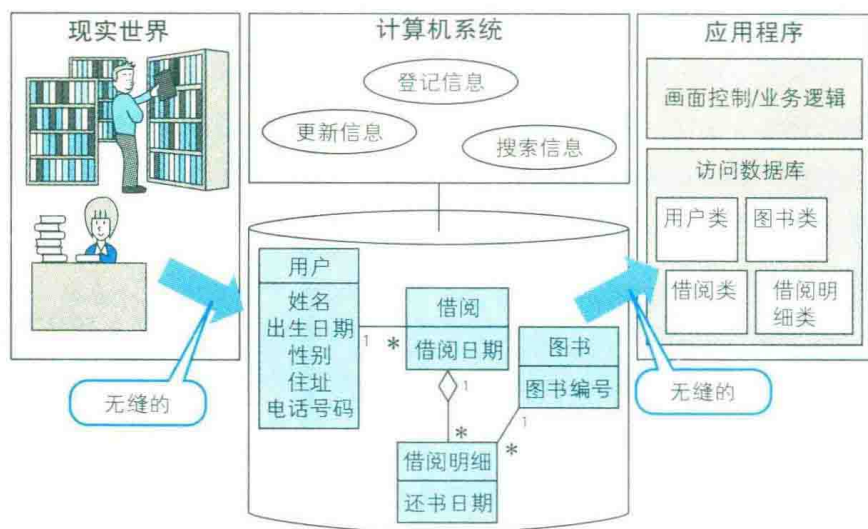


图 9-6 在业务应用程序中，数据结构反映现实世界

整体而言，这种将面向对象作为“直接将现实世界替换为软件的技术”的说明并不恰当，但就业务应用程序的数据而言却是基本成立的。大家很难注意到这种模糊的说明其实是一种误解，原因正在于此。

在业务应用程序中，数据结构反映现实世界。

9.11 嵌入式软件替换现实世界的工作

接下来，我们介绍一下嵌入式软件。

之所以叫嵌入式软件，是因为它是“嵌入到”机器中的。有人可能不明白嵌入式软件究竟是什么，实际上，它在我们身边很常见。现在许多电器产品和机器都是通过计算机控制来进行工作的。空调、冰箱、洗衣机和电饭煲等电器产品都由其中嵌入的微处理器进行控制。手机、DVD 播放器、汽车导航、液晶电视和摄像机等高科技产品中都运行着高性能 CPU。令人惊叹的是，一台汽车中嵌入了几十个微处理器，来控制引擎、动力转

向和上锁等，它们通过车中的网络互相连接。其他的嵌入式软件还有很多种，比如心脏起搏器、电梯的运行控制以及导弹控制等。

这些嵌入式软件拥有业务应用程序所没有的重要特征。那就是嵌入式软件驱动机器，原样替换现实世界的工作。例如，电饭煲实际做饭、洗衣机洗脏衣服，等等。业务应用程序的主要工作是记录信息，许多实际的工作还是由人来做，这与嵌入式软件有很大不同。

前面介绍过，计算机只是承担了现实世界中的一部分工作，而嵌入式软件则稍有不同。就像在河边洗衣服的老奶奶化身为软件，并被嵌入到洗衣机中，每天工作一样。第2章中否定的“面向对象编程的结构直接表示现实世界”，在嵌入式软件中也是成立的。

读到这里，有人可能会感觉有点混乱。稍微有点夸张地说，这种混乱非常重要。计算机到底是干什么的？软件承担了人们的哪些工作？为了理解这些根本内容，大家最好自己认真思考一下。下面我们将一边介绍嵌入式软件的建模，一边介绍笔者的想法，但大家也一定要自己思考一下。

9.12 嵌入式软件中设备的研究开发很重要

我们先来思考一下嵌入式软件的业务分析。所谓业务分析，就是整理计算机替换之前的现实世界中的工作的情形。嵌入式软件与机器一起来替换现实世界的工作，因此，整理这种机器被发明之前的工作的情形就相当于业务分析。

在洗衣机被发明之前，人们在河边或浴池洗衣服；在电饭煲被发明之前，人们用灶台做饭。与业务应用程序的情况一样，这些人们工作的情形也可以使用活动图来表示（图9-7、图9-8）。

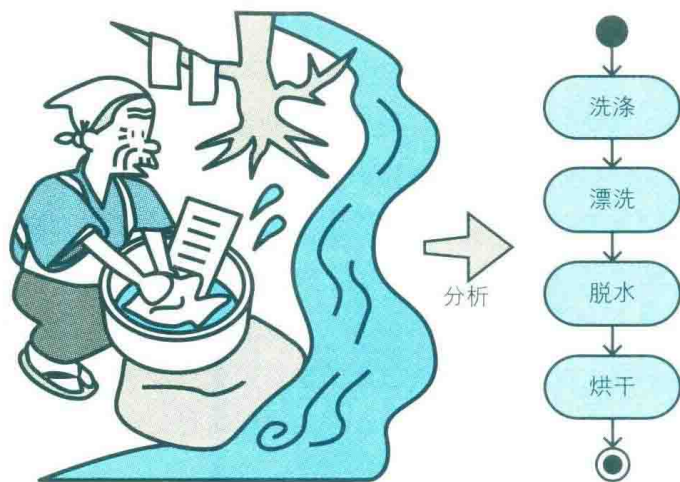


图 9-7 洗衣机的业务分析

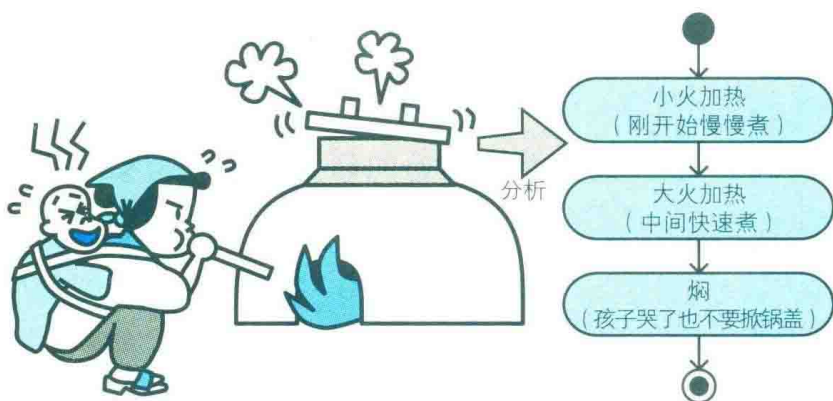


图 9-8 电饭煲的业务分析

像这样，在嵌入式软件中，我们也可以使用活动图来表示现实世界的工作情形。而在实际开发嵌入式软件的情况下，基本上不进行业务分析。这有两个原因。

第一个原因是，在很多情况下，相比业务分析，机器的发明和改良更加重要。最初开发洗衣机或者电饭煲时，一定调查过人们工作的情形。另外，也应该确定了工作方法的基本原理，比如刚开始慢慢煮，中间快速煮，

孩子哭了也不要掀锅盖。不过，即使我们明白这些原理，实际的工作还是由机器来做。因此，设备的发明是关键。以电饭煲为例，就是将锅设计成什么形状、选用什么材料，以及如何设计加热功能等。为此，通常会先对设备进行设计，然后再考虑交给嵌入式软件的工作。

在嵌入式软件中不进行业务分析的另一个原因是，这些工作人不一定能做。现在，随着技术的进步，很多能够进行人做不到的事情的机器被发明出来，比如我们身边的空调、手机和汽车导航等。虽然这些机器的工作也可以强行对应于人的工作，比如使用冰和扇子给屋子降温、使用烽火或信鸽进行交流、副驾驶的人使用指南针或地图时刻判断当前位置来导航，但实际上，就算对这些人的工作情形进行分析，也没有什么用。

因此，嵌入式软件中通常不会执行业务应用程序的业务分析工作，而是会对新设备的发明、既有设备的改良等加以研究。另外，在进行设备的研究开发的同时开发软件。

9.13 使用状态机图来表示全自动工作的情形

许多嵌入式软件不需要人的参与就可以持续运行。电器产品中常使用“全自动”一词来表示该特征。当使用洗衣机或电饭煲时，人只要在最开始按下按钮即可，接下来的工作都是全自动地持续进行的。除了这种根据请求来执行一整套工作的机器之外，还有空调、冰箱等机器，只要不断电，就可以一直运转几天甚至几个月。

另外，业务应用程序的主要工作是记录现实世界中的事物和事情，而这一工作在嵌入式软件中有时并不重要。其中一个原因是，这要求目标设备具有很高的可信赖性。我们需要磁盘等存储介质来记录大量的信息，而另一方面，汽车引擎、冰箱等在温度条件等非常严峻的环境下很难长期运行。嵌入式软件与业务应用程序的性质有很大不同，因此，需求定义中的建模内容和成果也会大不相同。

为了实现全自动控制，持续驱动机器的嵌入式软件会使用传感器等判

断当时的状况，自律地运行。因此，在需求定义中，确定在什么状况下执行什么动作也非常重要。为了表示这种规格，UML 的状态机图非常有用。这里以空调为例进行说明（图 9-9）。

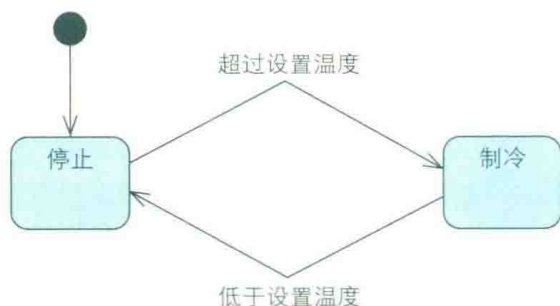


图 9-9 表示空调动作的状态机图

空调的运转原理是，制冷剂吸收屋内的热气，通过被外挂机压缩而放出热量，然后再次回到屋内吸收热气。虽然这些动作是由机器执行的，但是温度设置、风量调节等都由嵌入式软件管理。图 9-9 非常简单，实际的空调还可以通过定时功能来控制电源开关，并且具备快速制冷等功能。这些附加功能也可以使用状态机图来表示。

9.14 嵌入式软件一直执行单调的工作

嵌入式软件持续驱动机器，直接替换人们的工作。最近也出现了一些进行精细控制、看起来会执行高级判断的机器。而实际上，这只不过是机器中嵌入的计算机一直在运行程序中编写的逻辑而已。如果让人不分昼夜地一直做这些工作，那肯定是难以忍受的，但计算机只要通上电，就不会厌烦，也不会抱怨，一直不停地执行这些单调的工作。得益于计算机承担了固定工作，人们才能生活方便，安心休息。

现在的嵌入式软件变得越来越复杂，手机、汽车导航等程序都多达 100 万行以上。随着机器性能的提高，在以固定工作为中心的嵌入式软件领域，类似于手机号码簿之类的记忆工作也变得越来越重要。