**业务规则:事物应当如何运作**

**“业务规则背后的核心理念是任何组织都具有逻辑，组织会使用逻辑来执行它的运作管理任务。”**

* **Malcolm Chisholm 1**

**“在规则管理中首要和最基本的原则**

**是应当将规则录入数据库”**

* **Ronald G. Ross 2**

每个企业都有流程、规则和逻辑，这些定义了企业的运作方式。例如，军队有标准的指挥命令规定，大多数公司遵守一般的会计准则（CAAP），投资银行机构需要遵守风险控制标准。这些业务规则具有不同的表达方式；例如，流程模型、标准文档、书以及记录这些规则的其他文件。通常情况下，完全没有正式的业务规则阐释，业务规则几乎是虚无的，可能只存在于职员的脑海里和经验中（最坏的情况）。数据和业务规则存在共存关系。数据受规则的约束，而规则仅仅是指导，除非它对数据产生一定程度的影响。鉴于业务规则的重要性，具有一组正式的数据模型模式是很重要的，它能够以一致严谨的方式表示业务规则。

**这类模式有哪些重要意义？**

数据模型获取了不同实体之间的关系。可以将这些关系看成一种业务规则。例如，你可以这样理解图8-1：“每个**订单**必定是由一个或多个**订单项**组成的，而每个**订单项**只能是一个**订单**的成分。”3 这是个很有用的表述，它指明了在企业中**订单**和**订单项**需要遵守的行为方式。例如，这个模型暗示了规则，就是一个订单必须至少包含一个订单条项。但是像 “某人是为女士，假如这个人是个女性，她的年龄是21岁或21岁以上” 这样的业务规则该如何表示呢？4明确地获取这样的业务规则，并将其作为数据模型中的关系是非常困难的，但是你可能还是希望使用一个结构来获取这样的业务规则并将其关联到**人员**或**当事人**。在这种情况下，本章节没有尝试为特定的规则声明建立模型，而是提供了一种模式，我们可以使用这种模式来开发维护业务规则的数据模型，比如说上面所提到的业务规则和因素，这些因素用来判断某人是女性、男性、小孩、还是青少年等等。因此，你可以开发一个将这些规则“录入数据库”的模型，该模型包含：特定的规则声明、规则与数据模型中其他实体之间的关系、影响规则的因素以及规则可能会产生的各种成效。

**图8-1** 订单、订单项

本章节所讨论的是个专业的话题；因此，它是讨论模式的最后一个章节。在某个非常重要的方面，本章节的这些模式与本书中的其他模式是不同的。这些业务规则模式建立‘元数据’模型，以区别于‘业务数据’。这样做的是什么意思呢？业务数据描述的是实体的诗句是如何收影响或被引导的。由于我们将会进一步地描述数据，所以可以将业务规则作为元数据。有些建模人员可能会问“为什么你在业务规则与元数据相关的时候解决业务规则，而不是在它与业务数据相关的时候处理业务规则？”我们只是将元数据视为数据，仅仅是一种数据而已。

根据我们的经验，在很多为业务数据需求建模的情形中，会经常出现规则建模问题。在这些情况下，通常业务代表都期望业务规则和业务数据是你正在试图解决的问题的两个成分。换句话说，就是业务不区分元数据和业务数据。出于以上原因，我们认为这些规则建模方法是很重要的，而不是只是说“哦，那个是元数据，不在我们建模工作的范围之内！”

例如，常见的数据建模需求是为产品的价格建模。这似乎像一个业务需求而不是元数据需求。然而，当你稍微深思下，你会注意到由于地理区域、当事人类型（政府与商业机构）、数量或所购容量等因素的影响，价格是不一样的（美国的价格与欧洲的价格）。因此，你正真要建的模型是影响产品价格的各个因素，或者换句话说，是为定价业务规则和影响产品价格的各种因素建模，这些因素可能已作为模型的实体。无论何时你都得为影响数据的各种因素建模（例如，产品的价格或产品的折扣百分比），这表明你在为规则建模。

在很多情形中都会需要规则建模需求，我们想要的模型需求如下：

* 哪种产品或服务的售价是多少。
* 如何给客户或供应行排优先级和/或顺序影响优先级排序的因素有哪些。例如，客户在某一段之内订购的货物或服务的交易金额达到一定额度时，便会被给予更高的级别。
* 在不同的情况下使用哪个生产工厂或使用哪个物流承运商。例如，如果订货订单是在香港的话，你应当选择上海的制造工厂来生产这些货物，而不是选择新西兰的工厂。
* 如何追踪处于不同情形的客户。例如，当某人首次作为客户时（比如说发送一封感谢信），或是当某人提交投诉时（比如说打电话给那个投诉人），我们应当建议采取什么样的措施？
* 不同国家的国际邮政地址的格式是怎样的。例如，在美国表示地址要先从详细的地址信息开始（收件人的名称），最后才是最泛化的地址信息（国家名称），但是在伊朗或俄罗斯却并非如此。
* 如何、何时补充库存。例如，你可能希望只有当你‘流水线’订单达到一定数量时，才补充库存。

你为规则建模的另一个线索是条件语句，比如说‘if, then’语句或‘while’语句。例如，如果运达地址的地理范围是在美国，如果订购数量超过100项，如果客户类型是政府的客户，那么这个产品的售价是$100。再例如，如果客户提交投诉，如果这个客户是个金品忠诚会员或者该客户今年给你下单的金额超过$10,000，那么你会让副总裁致电该客户。

这个模式的优点在于当你认识到你在处理涉及业务规则的数据需求时，你可以选择保持与本章节中我们将共享的业务规则模式一致，为这个或其他类型的业务规则建模。我们遇到过很多次这样的情况：我们为某物建模后，却认识到这个模型实际上是‘规则’数据模型结构。通过识别条件语句类型或数据需求— 因素和/或成效，你可以预先认识到规则。然后使用规则模式，通过将合适的各种规则集合到同一个数据模型结构中，你可以创建出更加一致的数据模型；通过使用深思熟虑的数据模型结构，还可以开发出更高质量的模型。

由于对于很多企业而言，业务规则是至关重要的，所以本章提供了一些模式，使用这些模式，我们可以采用一致的方式以及与数据模型中其他实体相融合的方法为这些业务规则建模。

**本章节有哪些内容？**

本章节介绍了一些数据模式，当企业希望在数据模型中维护业务规则时，可以使用这些模式支持企业需求。

本章节以业务规则的定义开头，然后使用数据模型结构为业务规则建模提供了两种功能强大的模式。

和本书的大多数章节一样，每种模式的建模风格都是从较为特定的风格逐渐过渡到较为泛化的建模风格。然而，本章节与书中的其他章节又有所不同，本章节没有介绍一级业务规则模式。创建一级业务规则模式是可能的，在该模式中，可以将规则因素和规则成效作为**实体规则**的属性（在这里实体是业务规则的主体）。因为在大多数情况下，而不是所有情况下，将因素和成效作为实体来维护，正如二级规则模式所示，所以我们选择不展示一级业务规则模式。

不同程度的泛化模式适用于不同的企业和情形。例如，想要维护特定业务规则的企业可以使用二级业务规则模式，而想要单一的，泛化结构的业务数据模型时可以使用三级业务规则模式。

在本章节中我们使用了以下两个业务规则场景阐释这种模式：

* 关于如何在不同的条件下对某一**事件类型**做出响应的规则（事件是指发生在某一给定地点和时间的活动，它可能会触发其他的活动，例如，回应客户投诉的电话）。
* 基于不同因素，关于**产品**或**产品特征**价格的规则。

本章节包含以下内容：

* 业务规则的定义
* 支持业务规则定义的各种模式
* 每个模式的关联
* 深入分析每种模式
* 何时使用、何时不使用这些模式
* 每个模式优缺点的简介

**什么是业务规则？**

业务规则可以有多种定义。例如，广义上可以将他们定义为“意图影响或制约业务行为的指令”2，狭义上可以将其定义为“业务数据规则是一种数据约束，这个约束优越于数据模型所暗含的约束”。最后一个定义很有意思因为它预示着业务（数据）规则是数据模型之上的约束。在本章节的二级业务规则模式中我们将进一步探索这个理念。

本章节是关于指导业务行为和维护约束的业务规则。例如，我们关注的是下面几个业务规则：

* **如果我们证实客户的投诉有效的话，必须给客户发送致歉信。**这个规则规定了在某一特定类型的时间发生时，我们该如何应对，在此案例中，是获得了一份有效的客户投诉。
* **我们必须对少数民族所有企业的所有产品给予2折优惠。**这个规则规定了我在不同的情形中必须如何处理产品的价格。
* **在美利坚合众国，正常情况下报税截止日期必须设置成“4月15日”。**这个业务规则特地与**纳税**的属性纳税截止日期和**地理范围**—“美利坚合众国”相关联。

本章节中的模式获取了有关业务规则的数据、影响业务规则的各种因素以及业务规则所产生的成效。

业务规则模式需要支持以下几个概念：

* 这些模式需要能够获取有关业务规则的核心数据。例如，他们应当能够获取业务规则表述，比如说“我们必须对少数民族所有企业的所有产品给予2折优惠。”这些模式还需要能够对业务规则进行分类。
* 业务规则受因素影响，因素也就是影响成效的情形。模式需要捕获这些因素。因素可能与特定的实体相关；例如，因素可以是某人所处的地理范围；例如，相同的电视机，在欧洲（**地理范围**）的价格可能与亚洲（另一个**地理范围**）的价格是不一样的，或者是产品具有业务规则比如说“对于金额超过$999,999，数量超过1000的订单，全都给予20%的折扣” ，反过来业务规则可能会受各种因素的影响，对于不同因素我们使用实体：**产品目录**、**订单价值**和**数量分段**来表示。
* 模式需要能够获取业务规则的所有成效（和成效类型）。例如，模式可以维护在特定条件下的价格成效，比如由于各种各样的因素影响，价格将会降低百分之二，或者是给价格添加百分之一的附加费。

**注：业务规则模式不捕获已发生之事的信息，只捕获不同情形中应该会发生的事情。换句话说就是，我们捕获规则声明、所有影响规则的因素以及所有的规则成效。当某一特定的规则被应用于一组特定因素的值时，规则表示应当会出现一个特定成效或一组成效。例如，在某种情况下折扣会影响特定产品价格或者是指明某一事件发生时间的规则，这样的规则需要特定的交流事件，比如说致电重要客户。**

通常人们认为流程建模（该模型获取了很多的业务规则）与数据建模完全不同。在流程模型中，企业通常通过规则字典、元数据库或规则引擎来获取业务规则，它不依赖于数据模型。然而，我们认为可以使用其他类型的模型来获取这些规则，在数据模型中维护诸多规则对你会更加有利。规则和数据通常是紧密相连的，很多时候在同一个模型中维护他们会更好。我们认为数据建模人员应当创建模版，以用于数据模型获取规则数据，并且在他们分析的过程中，还可以将这个模版用于他们所遇到的业务规则。在数据模型中维护业务规则，然后在后续的数据库设计中实现时，可以动态的更改他们。例如，你可以使用数据模型来维护有关产品各种定价规则（例如，当销售额超过100时，给予2%的折扣），那么当规则发生变化时（例如，当销售额超过100时，给予3%的折扣），通常情况下你可以只改变数据模型的实例（以及基于该数据模型的数据库），而不用改变数据模型的结构（或数据库）或应用。因此，你的模型更加贴近‘数据驱动’型模型，并且会更加灵活。

**二级规则模式**

业务规则可以采用特定的建模和关联方式，可以使用二级业务规则模式。使用这种模式可以获取某一特定类型业务规则数据，然后其他类型的业务规则可以复用这个模式。获取的数据包括：业务规则名称、业务规则声明、因素、效果以及业务规则的各种分类方法。使用这种模式，你可以使用数据模型中现有的特定因素来影响业务规则。你还可以使用数据模型中尚无的泛化因素。这种模式获取了业务规则所产生的各种效果。当数据专业人士希望为每类业务规则开发一个特定的数据模型时，这种模式为他们提供了灵活的策略。

**为什么我们需要这种模式？**

二级业务规则模式的目的是维护业务规则数据，比如说业务规则声明、业务规则因素和针对某一特定情形、问题领域和业务主题领域的业务规则成效。例如，试想下零售公司对补充库存的业务规则的定义和所有可能的库存置换规则成果感兴趣。对于这个例子，因素可能是产品、再订购水平、位置和时间。在这些因素集合的基础之上，可以分配“再订购数量”，这代表了‘成效’。例如，产品“A123”，如果秋天（一年中的这个时间是旺季）巴黎工厂（位置）的库存水平降落到100件以下（也就是再订购水平），那么就再订购200件这样的商品（在订购数量，这也是这种情形下的规则成效）。

在数据建模时会出现很多的情况，比如说上面的情况，这种模式可以让数据专业人士通过以下这种方式来解决每种情况：使用通用模版为规则、相关因素和成效建模，并将这些规则整合到他们的数据模型当中。

**注：有些企业家喜欢通过将规则数据记录到文档或电子表格中，将其作为数据模型的附件，通过这种方式来维护业务规则。5 可能会提供一种可行的格式：**

* **业务规则的名称。例如，“最多课程报名规则[Maximum Course Enrollment Rule]”。**
* **规则的业务描述。例如，“在招生的时候，录取学生的数量必须不能超过该门课最多所能提供的座位数”。**
* **规则的‘模型定义’。这是个SQL语句 — 就像规则的定义一样（或者是任何其他语言），使用模型中的实体和属性。以我们大学课程为例，登记学生的数量不能超过“最大课程座位数”，这个规则的模型定义可以是“if count(COURSE ENROLLMENT) > COURSE. Maximum Allowed Seats where COURSE ENROLLMENT. Course Id = COURSE. Course Id，那么可以发送ERROR MESSAGE. name — ‘已超过课程最大登记数’”。应当注意的是在数据模型中这个假设有一个具有最大容纳座位数属性的课程实体[COURSE]和课程登记[COURSE ENROLLMENT]实体。**
* **成效（比如说成功信息或错误信息）。例如，要么是提示成功结果信息“登记成功”，要么是提示错误信息比如说“已超过课程最大录取数”。**
* **影响规则的因素。例如，该课程最大能容纳的座位数和已注册学生数。**

**尽管在有些时候这样做是个不错的选择，在本章节中，我们不是在文档中记录业务规则，将其作为数据模型的附件，而是展示了如何使用数据模型及相关实体来维护诸多业务规则（最后在数据库中实现）。这让业务规则处于前沿和中心。不要只是说业务规则只是元数据，我们认为业务规则可以整合到数据模型和数据库层级的相关实体中。6**

**注：很多企业认为业务规则应当放入元数据存储库（或业务规则引擎）中，它与数据模型相关。我们不赞成这种做法，根据我们的经验，很多企业很难实现完全整合的元数据库（或业务规则引擎）。这种模式提供了有用的备选方案，可以直接在数据模型内维护很多的业务规则。即使你实现了元数据库（或业务规则引擎），但是这些模式只是在你实现加载运转之前作为权宜之计使用。**

**这种模式是如何运作的？**

图8-2所示的是维护业务规则数据的二级模式。这张图演示了业务规则模式的若干成分：

* **规则的主体（模型图的顶部所示）：**受一个规则影响的一个或多个实体。例如，价格的业务规则可能会影响**产品**和**产品特征**实体，这些将会成为价格规则的主体。
* **规则（实体）自身（在模型图的中间）：**规则实体是中间实体，它与主体、因素和成效相关联。规则还包括规则名称、规则声明以及分类方式（通过**实体规则类型**实体）。
* **因素（在模型图的中间部位的左边）：**这些是影响规则成效的条件。例如，地理范围或相关的当事人类型都会影响规则的成效。因素可以分类为：属性因素、特定因素或泛化因素。
* **属性因素**指的是将因素作为规则实体的属性。如图8-2所示的因素属性1[factor attribute 1]（“DATE”日期类型）和因素属性2[factor attribute 2]（“CHAR”数据类型），然而，因素属性可以属于任何数据类型，通常情况下这些属性因素是指有效起始日期[effective from date]和有效终止日期[effective thru date]（如443页图8-4所示），这些属性展现了规则条件，也就是规则只能在这个日期内有效。
* **特定因素**与模型中现存的特定实体相关。在图8-2中，这些实体是指实体3和实体4。
* **泛化因素**是用于那些没有作为属性也没有与现存实体相关的因素。在图8-2中，是在**实体规则因素**和**规则因素类型**中维护这些因素。
* 成效（在模型图的中间部位的右边）：这些是规则的成效。例如，比如说购买者的所处地理范围和所属当事人类型这些因素，可能会使用价格规则作为成效。与因素相似，成效也可分类成：属性成效、特定成效或泛化成效。
* **属性成效**是指将其作为规则实体的属性来维护。在图8-2中，这些属性是指成效属性1[outcome attribute 1]和成效属性2[outcome attribute 2]（当然，可以有任意数量的属性成效）。
* **特定成效与**已存在的特定实体相关，这些实体可能是规则的成效。在图8-2中特定规则是实体5和实体6。
* **泛化成效**是用于那些即不作为属性也不与现存实体相关的成效。在图8-2通过**实体规则成效**和**成效规则成效**获取这些泛化成效。

在图8-2中的模型图顶部，你可以看到**实体1**和**实体2**，这两个实体代表了你想存储实体规则的任何实体，换而言之就是业务规则主体。你可以看出“每个实体1（和/或实体2）可能会受一个或多个**实体规则**影响，并且每个实体规则必须实体1的唯一规则（和/或实体2）”。也就是，实体规则是影响实体1（和/或实体2）的业务规则，而实体1（和/或实体2）是实体规则的主体。这些实体（实体1或实体2）可以是产品、订单、装运、当事人或任意数量的维护规则的其他实体。因此，根据主体划分，可以有产品规则、订单规则、装运规则或当事人规则。例如，装运是装运规则的主体。装运规则将包含规则：比如说“电脑零部件必须不装运至朝鲜”或“没收到客户的有效签字的话，会认为装运未交付”。

和本书中的其他模式不一样，实体规则并不总是使用现成实体替换**实体**。可能会有一个**产品**，它是**产品补给规则（**规定了如何以及何时补充产品库存）的主体，还是**价格成分规则**（根据各种因素决定价格）的主体。可能有个**客户**实体，它是优先规则的主体，这个规则规定了如何基于各种因素给客户排列优先级或评级。

**（图）**

**图 8-2** 二级业务规则模式

通常情况下，业务规则的主体是一个单一的实体；然而，业务规则结构的主体也可以不止一个实体。例如，稍后本章节所示的图8-4，在该图中**产品**和**产品特征**（产品的差异性，比如说颜色、尺寸等等）都可以是**价格规则**的主体。**价格规则**既可以维护**产品**的价格也可以维护**产品特征**的价格。此外，有时候一个实体和业务规则实体之间可能会存在多重关系。例如，可能会存在与**价格成分规则**、**产品补给规则**和**产品管理规则**都相关的**产品**实体， 这表示在各种不同的情况下必须执行哪类规则。后面的这条规则可以维护产品基于不同因素的出售方式信息。例如，根据安全条例，金融证券公司不得向未具备一定程度净资产的当事人出售某一特定对冲基金（一项高收益高风险的投资），或者是将当事人在对冲基金中所占的投资组合金额限制到一定的比例。

**注：我们发现在企业中，很多核心实体都可能会有相关的业务规则。例如，物流公司，通常情况下它的数据模型需要有装运规则和当事人规则。对于投资银行而言，通常需要明确地体现实体交易规则、限制规则和价格规则（最终会成为表）。这样可能会让你产生这样的一种印象，即数据模型中的每个实体都必须有个实体规则来表示实体的业务规则。情况并非如此。一般情况下我们认为只有核心实体是业务规则的主体，数据模型中的其他实体可以作为业务规则实体（实体规则）的因素（也可能是成效）或者这些实体可能与业务规则毫无关系。**

实体规则获取了业务规则的属性和关系。因此，实体规则与主体、因素、成效、分类以及业务规则的其他数据相关。例如，使用**价格成分规则**的规则声明[rule statement]属性获取业务规则—“我们必须给予少数名族所有企业2%的折扣”。 这条规则可以有个规则名称[rule name]，比如说“价格规则1280 — 少数名族企业折扣”。

**注：可能还会添加其他的属性或实体来补充业务规则模型，这种模式可以包括这些属性或实体（后面的模式也可以）。例如，可能会有以下这些属性或实体：**

* **规则备注，一项说明或重要的注释。**
* **规则引用指的是一些权威的引用。例如，规则“国会不得制定关于下列事项的法律：确立国教或禁止信教自由……”是美利坚合众国宪法权利法案的第一修正案，其引自“美国国会图书馆”。关于这条规则我们还有其他的引证权威机构，比如说“康奈尔法学院”和“国会年鉴434（1789-6-8）”。**
* **规则来源指的是规则的出处（规则应当只有一个源头）。2“国会不得制定关于下列事项的法律：确立国教或禁止信教自由……”源自“美利坚合众国宪法第一修正案”。**
* **规则制定人，可以指明规则的制定者或定义人。还可能有一条规则规定“规则应当直接有具备相关知识的那些人来指定”。例如，“国会不得制定关于下列事项的法律：确立国教或禁止信教自由……”是由“詹姆士麦迪逊”制定的。**
* **规则管理者，通过给规则指定负责管理规则的当事人来管理规则。例如，“国会不得制定关于下列事项的法律：确立国教或禁止信教自由……”这条规则由“最高法院（规则的监护人、规则的仲裁者以及规则的诠释人）和总统（宣誓保存、保护和捍卫它）”管理。**
* **外部引用标识，它可以获取业务规则来源的标识，比如说集中的业务规则引擎或元数据库。例如，源自**[**http://www.usconstitution.net/const.html**](http://www.usconstitution.net/const.html) **的“国会不得制定关于下列事项的法律：确立国教或禁止信教自由……”**

可以将以上数据作为属性或独立的实体来维护。例如，要复用规则引用（例如，多次引用康奈尔法学院）或引用有其他的数据或关系（例如，引用的全称和简称），那么可以使用第三章节中的上下文角色模式来处理角色来源[rule source]、角色制定者[rule specified by]和角色管理者[rule managed by]数据。这个模式通过不同的角色将各个当事人关联到每个规则（如图8-7所示，在本章节的最后部分将会做出解释说明）。

这个模式表明“每个实体规则只能被一个实体规则类型分类。”例如，你可以对**价格成分规则**的实例进行分类，“我们必须给予少数名族所有企业2%的折扣”，这条规则属于“折扣”这一**价格成分规则类型**。另外一个**价格成分规则**实例的规则声明为“2009年产品A4672 在欧洲的售价是€15”，可以将此实例归类于名称为“价格”的**价格成分规则类型**。价格成分规则还有其他的分类，比如说“附加费”、“常规费用”和“使用费用”。实体规则类型的递归关系允许规则类型相互隶属。例如，可以将规则“我们必须给予少数名族所有企业2%的折扣”归类于“少数民族折扣”，而这个规则类型是在“折扣”这个规则类型之内。**实体规则类型**中的描述属性是很重要的，该属性能够解释**实体规则**所维护的规则特性。例如，有些实体规则被归类于“折扣”，“折扣”是**实体规则类型**名称。描述可以是“这个类型的规则指明了影响价格下降的因素以及降价产生的成效 — 要么是降低价格百分比要么是降低价格金额”。

**注：规则的另一种分类方法是根据BRS规则分类模式（BRS Rule Classification scheme）或根据规则表述模版（Rule Speak Template）对规则进行分类。2 如果你需要通过多种方式对业务规则进行分类，你可以使用三级分类模式，该模式使用多种方法对业务规则进行分类，如本章节中的图8-5所示。这说明了模式的优点。当你需要更灵活的业务规则分类方案时，你只要插入三级分类模式便能得到更加泛化的解决方案。如果需要更加特定的解决方案时，你可以用二级分类模式，甚至是一级分类模式替换三级分类模式。**

可以将因素定义为造成某一特定成效或情形的元素，也就是影响规则成效的条件。例如，价格成分规则的实例具有规则声明“如果客户居住在北美，那么价值在$500,000和$999,999之间的任何订单，都可以享受10%的折扣”。在这条业务规则中的一个因素是客户的居住地是否在“北美”这个地理范围内。另一个因素是订单的价值是否在特定的区间内，也就是在“$500,000”到“$999,999”之间。

可以将因素分成三种类别：第一，因素属性，作为业务规则的属性；第二，作为数据模型中已经存在的实体，或者是需要因素作为此规则的实体（我们将这些因素作为特定因素，因为他们是特定于模型中的实体的）；最后，是其他的泛化因素。

第一类是属性因素，这些因素作为业务规则的属性，在图8-2**实体规则**实体中以因素属性1[**factor attribute 1**]和因素属性2[**factor attribute 2**]为代表。这些因素通常是指日期范围比如说可以应用规则的时间段。例如，“皮毛大衣”的折扣规则只在夏季有效，也就在6月1号至9月1号之间有效，所以应用这种模式时，这些因素属性可以是有效起始日期[**effective from date**]和有效终止日期[**effective thru date**]。

第二类，特定因素，包括数据模型中已存在的因素实体。在图8-2中以左手边的实体3和实体4为代表。在我们的示例中，“如果客户居住在北美，那么价值在$500,000和$999,999之间的任何订单，都可以享受10%的折扣”，可以根据**地理范围**实体中“北美”实例的关系获取“北美”。数据模型中可能有已经定义好的地理范围。

重要的是指出业务规则不独立存在。他们通常与很多因素相关联，这些因素可能已经作为数据模型的值。当你使用这种模式获取业务规则时，重要的是：使用现存实体获取现存的各种影响业务规则成效的因素，让后将他们关联到**实体规则**。这样做，你可以避免在模型中冗余地捕获数据。例如，如果是地理因素，你不需要在**规则因素类型**实体中录入所有的地理区域，你可以将**实体规则**关联到现存的**地理范围**实体。此外，将这些因素关联到现存实体，通常你可以认识到某一特定实体的业务规则的真正范畴。也就是你可以看到表示业务规则条件的实体。另外，如果将业务规则整合到数据模型中，当提供因素数据的实体发生某种程度的变化时，有助于获取因业务规则变化而造成的影响。例如，如果规则规定对于一定量的北美销售额给予10%的折扣，你将地理范围更新成“N. America”，那么规则和余下的数据模型也会同步更新该数据值。

最后一类因素包括**规则因素类型**所维护的因素。这些是泛化因素，没有明确将它们关联到数据模型中的现存实体，也没有将其作为业务规则中的特定因素属性，而是将他们作为**规则因素类型**的实例。例如，在规则“本年度跨国旅游超过3次的乘客必须比跨国旅行0到2次的乘客优先享受升舱机会”，“0至2”和“3次或3次以上”是**实体规则因素**的因素值示例，它的规则因素类型是“本年度跨国旅游”。没有将这些因素建模成特定实体（可以创建一个**本年度跨国旅游**实体）的原因是这个数据可能会只在一个地方使用。为支持这样一个特定的需求创建一个新的实体有意义吗？还有更多这样没有建模成独立实体的特定因素。可以使用**实体规则因素**和**规则因素类型**支持这些因素，我们将这些因素称作为‘泛化因素’。

所以，在何时使用这些没有作为属性的‘泛化’因素呢？可以使用**实体规则因素**的因素值（维护相关的数据）和**规则因素类型**名称（维护“有效起始日期”和“有效终止日期”）为特定的属性建模，比如说有效起始日期[effective from date]和有效终止日期[effective thru date]。我们使用以下指导方针：在因素是单一的不可分割的数据块时候（它并不代表重复组的值），而这个特定的数据块也没有其他的数据或关系，不会一次又一次的复用同一个值的时候（因此将会使用一个查询实体），可以将这个因素作为属性来维护。

另外要考虑的是属性是否能够为大多数规则实例所共用。有效起始日期[Effective from date]和有效终止日期[effective thru date]是因素属性1和因素属性2的示例，这两个属性几乎可以应用于实体规则的每一个实例，因而一般情况下将他们作为特定的属性因素。‘泛化’因素可能是仅仅针对**实体规则**中的某些业务规则；因此，使用**实体规则因素**捕获了这些因素。例如，“从配送中心至最远客户位置的最大英里数”，这个因素仅适用于少数的**价格成分规则**实例，因此如果将这个因素作为属性的话，那么大多数时候这个属性是空的。还有，如果你将这些泛化因素全都作为属性的话，会对视灵活性，而且你还必须维护诸多属性。

使用**实体规则因素**和**规则因素类型**最大的好处是灵活性。随着时间的推移还回出现其他的因素类型，可以将他们作为实例添加到**规则因素类型**中。例如，随着时间的推移出现了影响舱位升级的其他因素，比如说乘客境内旅游次数或旅行总次数，我们可以将这些作为实例添加到**实体规则因素**和**规则因素类型**中。

至于你采用哪种方式为各种因素建模（分类成属性、特定因素或泛化因素），这由你决定（你应当反映业务需求）。也就是说，如果你将因素归类于，那么是因为他们适用于每个实体规则，而数据模型中已有的特定因素作为实体（比如说影响价格规则的**地理范围**）。而采用**实体规则因素**获取泛化因素是因为数据模型中没有相关的实体，比如说“本年度的跨国旅行”应当采用哪种建模方式呢？虽然我们提供了一些指导和选择，但是问题的答案通常要依据具体情况而定。

**注：通常我们认为泛化因素与派生数据相关，因为派生的数据没有存储在数据模型中。例如，某段时间，客户跨国旅行的次数就是派生数据，所以没有直接使用数据模型来维护。客户解决账单平均耗时也是派生数据。当涉及到这类派生数据时，这类数据可能是个很好的提示，表明适合使用泛化因素。**

需要注意的是对实体规则而言，与现存实体相关的各个特定因素是可选的。例如，“每个实体规则只能以一个**实体3**为基础，而每个**实体3**可以是一个或多个实体规则的条件”。**实体规则**是否可能受强制性因素影响或者是否有可能所有的因素都是强制的？答案是肯定的，但是通常我们认为业务规则混搭因素会产生不同的成效。例如，价格规则可以使用以下因素的各种组合：**地理范围**，**角色类型**、**产品类别**、**数量分段**和/或**订单价值**。其他规则里也可以使用这些因素的任意组合。因此，可以使用一个价格成分实例来记录：某段时间内，**地理范围**为“美利坚合众国”，**产品类别**为“配件”的产品给予5%的折扣。另外一个实例记录：某段时期，**地理范围**为“加拿大”，**订单价值**超过“CAN$2,000”的订单给予3%的折扣。因此，任何**规则实体**实例都可以与因素的任意组合相关。这就是为什么因素与业务规则之间存在可选关系，而**实体规则**中存在可选因素属性。

可以使用以下三种方式来维护业务规则的成效：

* 可以将成效作为属性来建模。在业务规则的成效是单一的不可分割的数据块（没有重复组的值），并且这个特定数据块没有其他的数据或关系的时候，在成效的值不会被一次又一次的复用的时候，可以将值作为**实体规则**的属性。图8-2所示的属性：成效属性1和成效属性2是可能满足这个标准的属性。另外一个示例，如本章节稍后所示的图8-4。价格成分规则可能会下面所示的一个成效或全部包含：价格金额[price amount]、折扣金额[discount amount]、附加费金额[surcharge amount]、折扣百分比[discount percentage]和附加费百分比[surcharge percentage]。这些都是模版属性成效属性1[outcome attribute 1]和成效属性2 [outcome attribute 2]的示例。由于这些成效没有其他的数据，所以这些值是不可分割的。此外，一般情况下，这些值不能为其他规则实例复用。也就是说，通常**价格成分规则**实例会产生一个仅针对该实例的价格金额、折扣金额、附加费金额、折扣百分比或附加费百分比。因此，要将他们作为实体的属性来建模。
* 第二，规则成效可以作为数据模型中实体的一个实例。例如图8-2中的实体5和实体6。例如，实体5可以是指工作计划类型，它可以是监管审计项目总支出的规则对象。如果项目超出预算达到一定的金额，那么规则会导致需要设立“审计项目”**工作计划类型**来调查项目超支的原因。
* 最后，可以使用更加灵活的方法处理业务规则成效，也就是将结过作为一个或多个**实体规则成效**。在图8-2的右下角，你可以看到代表**实体规则**泛化成效的实体。受不同因素的结合的影响（实体3、实体4、实体规则因素、因素属性1和因素属性2），业务规则可能会导致不同的成效。“在一天之间给金牌会员投诉客户致电并提供一些补偿”，此业务规则可能会有若干个不同的成效，例如，“一天之内给客户致电”和“提供客户购物礼券”。因此，可能会有若干个成效需要维护，而每个成效还有其他的数据，比如说“1”天之内中的“1”和提供的礼券金额。

所以，为什么这些成效很重要呢？成效维护的是规则的成果，换句话说就是在不同条件下会发生什么情况，因此，他们是规则不可分割的一部分。此外，可以将规则的这些成效输入到其他规则中，或者他们触发一个事件或流程。例如，如果交易**限制规则**的成效是“限制超过$100的交易”，这个成效会触发另外一个规则，该规则导致给银行的信贷风险部门发送邮件这一事件的发生。在数据模型方面，**限制规则成效**实体（**实体规则成效**的一个实例）是**交流事件规则**的一个因素。如果“超过了$100的限制”这一情况发生的话，那么**交流事件规则成效**便是“给信用部门发邮件”。因此限制规则成效可以是**交流事件规则**的特定因素和条件。

**注：尽管图8-2中模式展现了实体规则与因素、实体与成效之间的一对多关系，但是他们之间也存在多对多关系，所以你可能会需要更改模式以适应这样的需求。例如，可以根据若干个地理范围制定价格规则，比如说给“纽约州”、“新泽西州”和“康乃狄克州”提供一定的折扣率。因此，价格成分规则与地理范围这一特定因素之间可以存在多对多关系。本章节稍后介绍的三级业务规则模式既清楚展示了规则与特定因素的多对多关系，也展示了规则与成效的多对多关系。**

为了进一步说明这种模式，试想下一个称为坎特威茨电子的电脑硬件零售商。作为理解公司业务计划的一部分，公司聘请了具有丰富流程建模经验的数据建模人员来研究两个截然不同的业务规则成分。作为一部分任务，数据建模人员必须创作一个一致的方法以处理贯穿企业各方面的业务规则。基于这些需求和二级业务规则模式的应用，数据建模人士制作了图8-3和图8-4。

我们来看下图8-3。数据专业人士被要求检验各种事件的业务规则，比如说接听客户投诉电话。正如本章节先前所述，事件的定义是“在给定的时间和地点内发生的活动，并且这个活动有可能会触发其他的活动”。因为可能会发生的不同类型的事件都具有规则，所以**事件类型**是**事件类型规则**的主体。

数据专业人士询问了下服务部经理，职员应当如何处理不同类型的事件。经理起初回应：如何处理不同的事件类型取决于发生的交流事件类型。数据专业人士知道企业数据模型中已有**交流事件类型**（“电子邮件”、“打电话”、“面对面”等等），因而决定复用这些值，将其作为**事件类型规则**的特定因素。所以，“每个**事件类型规则**只能以一个**交流事件类型**为基础”。例如，客户投诉是一个事件，一个人如何回应这个事件通常情况下取决于**交流事件类型**（“打电话”、“传真”等等）和很多其他的因素。

如果客户通过电话或邮件提交投诉并且该投诉是有效的，那么客户服务人员就要给客户打致歉电话。这是**事件类型成效**中的一个成效示例， **事件类型成效**与**规则成效类型**中的“道歉”相关联。如果客户“面对面”提出有效投诉，规则要求客服人员本人要先道歉，然后送张信用优惠券，客户下次购物时可以使用（两个事件成效实例）。客服认为与电话投诉的客户或电子邮件投诉的客户相比，“面对面”提出投诉的客户需要更多的关心和关注。

（图）

图8-3 二级业务模式应用示例，交流事件

**注：在这个示例中，我们没有将任何的特定因素或成效作为属性。在下个例子中，也就是本章节稍后所示的图8-4，使用属性来捕获成效和因素。这说明可以调整业务规则模式以使其满足不同的需求。**

表8-1为图8-3中的可能实例提供了样例数据。在表的第一列描述的是规则实例。记住规则声明是个可选属性，企业可以选择不维护这个数据，因为事实上规则的其他数据描述了规则，通过维护规则的参数，比如说规则的因素和成效。

表的第一行维护的是事件类型规则的一个实例，这个实例的规则声明是“当客户拨打电话提出有效投诉时，我们必须致电道歉”。以上这个规则归类于名称为“客服”的事件类型规则类型。当某人希望查看所有的“客服”规则或“定价规则”等等规则时，这样的分类是很有用的。

这条规则的主体是事件类型，因为规则是指当某类事件发生时，应该采取什么样的措施。因此，该事件类型的名称是“客户投诉”。这个规则实例基于两个因素，在接到“电话”和接电话的人判断投诉电话的“有效性”的情况下。因此，其中一个因素是特定因素，因为该因素与现存的实体和名称为“打电话”的**沟通事件类型**实例相关。

由于现存实体没有维护指明判断投诉有效性的参考因素，所以使用的是泛化因素实体，并让名称为“有效”规则因素类型与这条规则相关。因此，这条规则基于两个因素，也就是“打电话”和判断投诉的“有效性”。规则的成效或成果是指当这些因素出现时，会有个名称为“道歉”的**规则成效类型**，通过“打电话”进行交流沟通的。

**注：规则有多种分类方法。例如，规则声明 —“当客户拨打电话提出有效投诉时，我们必须致电道歉”，可以归类于表8-1所示的“客服”规则类别，它还可以归类于“投诉”，这个规则类型是在“客服”规则内部，可以通过事件类型规则类型的递归关系来容纳“投诉”类型。另外一个选择是将它归类于“沟通”规则类型。因为有些规则需要多种分类，在三级模式中我们提供了更灵活的规则分类方法。**

**表8-1 使用二级规则模式的范例。**

**（表）**

**注：“规则声明”似乎是多余的，因为我们维护了各种因素和成效，并将他们关联到规则实体。然而，为了弄明白规则究竟指的是什么，使用‘直白语言’（任何语言都适用）陈述规则是通常很有用的。较为成熟的企业可以根据因素和成效推断出规则声明。在这种模式中，我们展示的规则声明是可选的，因为有些时候企业可能决定维护规则声明，而有些时候企业可能会选择不维护规则声明，例如，当有成千上万的规则实体实例时。**

表8-1中的其他行展示的是不同情形下的规则。例如，第二行维护了规则：当客户提交“客户投诉”（**事件类型**名称）时，他们通过“打电话”（**沟通事件类型**名称）和泛化因素“无效”（**规则类型因素**名称）来处理这个问题，那么这中情形的特定成效是通过沟通事件类型“打电话”传达“解释”（**规则成效类型**名称）。

需要注意的是第五行展示的规则可以产生多个成效。当投诉因素是“面对面”传达的并且是“有效”投诉的时候，企业制定了两个成效：应当“面对面”（**交流事件类型**名称）地“道歉”（规则成效类型名称），还有将面值“50（美元）”（**事件类型规则成效**中的成效值）的“信用优惠券”（**规则成效类型**名称），通过“礼包”的方式送给客户。（为了查看“美元”这个值，可以创建它与另外一个实体 — **货币类型**的关系）。

数据建模人员发现了很多没有与数据模型中现存实体相关联的因素，并且也没有使用**事件类型规则因素**和**规则因素类型**实体（泛化因素）为这些数据需求建模。例如，我们刚刚讨论的哪种因素 — 投诉是“有效的”还是“无效的”，就是没有与现存实体相关联的因素。现存实体没有包含的另外一个实体是客户企业业务往来的年数，也就是他们的“客户周年纪念日”，例如，1周年、2周年、3周年或10周年纪念日（**时间类型**名称是“客户周年纪念”）。表的最后一行展现的是客户有周年纪念日，并且是10周年的纪念（**规则因素类型**名称是“客户周年纪念日”，**事件规则因素**中的因素值是“10”），规则成效类型是客户会收到一份源自市场销售部客服人员的“感恩致谢”信（**沟通事件类型**是“信件”）。这表明泛化规则因素可以维护适用于它的参数，并且你可以使用因素值属性维护任何描述成效的其他参数。客户十周年纪念时的另外一个成效是收到“笔套装礼盒”，通过“包裹”（**沟通事件类型**名称）寄送给他们。

在这个示例中，可以有其他的泛化因素，而随着时间的推移也会出现其他的因素。例如，企业后来决定如何处理投诉要取决于投诉的性质。如果关于“账单”问题的投诉，与“产品不能用”的投诉， 可能会采用不同的回应，这些是影响规则成效的其他因素，可以将他们作为**规则因素类型**实例的名称。

规则因素类型使用递归关系可以让规则具有多层次的分类。**规则因素类型** —“账单”和“产品不能用”可以是另一个 **规则因素类型**实例 —“投诉性质”的分类。因此，可以将各种因素组合在一起形成更高层次的**规则因素**类型，比如说“投诉有效性”（进一步分类成“有效”和“无效”）、“严重程度”（可以分类成“高”、“低”）等等。这样可以维护大量的业务规则，这些业务规则依赖于随着时间推移而出现的其他类型的因素。

同一个数据专业人士也获取了价格的业务规则。在于销售部、市场部和会计部的领导会面后，数据专业人士使用业务规则模式制作了图8-4。从图8-4中你可以看到关于如何实现同一模式的示例。例如，“每个**产品**和/或**产品特征**可以根据有一个或多个**价格成分规则**来定价”。7大多数企业对他们的产品和产品特征（产品特征指的是产品的样式，比如说颜色、尺寸等等）具有不同的定价方式。在这个示例中，你可以看到坎特威茨零售使用了不同的因素常见它的价格成分规则。这里有此模型的若干个实体，这些实体是企业数据模型的一部分，被用来作为因素，例如：

* **地理范围**可以是指邮政区域、城市、州、国家、大陆、销售区域或其他诸多地理区域。在表8-2中，你可以看到地理范围包含了“北美”。
* **角色类型**包含了人员或组织所扮演的各类角色，这些角色影响了产品的价格。例如，坎特威茨零售会给“合作伙伴”这个角色更加优惠的价格。在例如，扮演“职员”角色的人可以享受批发价。
* **产品类别**可以让价格关联到不同的产品分类，比如说产品族、产品系列或产品类型。例如，某个产品系列在某一特定时期可以有折扣，比如说夏天的皮毛大衣，
* **数量分段**可以根据订购项总数来确定特定的价格层次。例如，订购100件，可以成为降低产品价格的一个因素。
* **订单价值**包含的是订单最低金额至最高金额的区间，当订单价值满足这个区间时，它便可以成为影响价格的一个因素。例如，订单价值在$500,000至$999,999之间的话，可以降价。订单价值与订单的金额是不同的，它是订单的一个属性（或根据订单项金额推断出来），因为**订单价值**表示的是价值范围，并且只是作为说明这个因素的方法。此外，如果数据模型没有捕获**订单价值**，那么可以将**订单价值**作为**价格成分规则因素**中的泛化因素。
* 在**价格成分规则因素**中维护其他的泛化因素。例如，从分发中心到客户交付地点有多远或客户支付历史订单所花费的平均天数。数据模型没有维护这类数据（与这个规则不同），因而，它没有绑定到模型中已存在的特定实体。

注：为什么这个实体被称为价格成分规则而不是产品规则？正如我们先前所述，**实体规则**名称（如图8-2所示）可能不会总是包含其规则主体名称，很多时候一个**实体**会有很多的业务规则实体，比如说产品，因而，每类业务规则需要更特定的业务名称。另外一个方案是泛化所有关于产品的规则，以至于它包含**价格成分**以及其他的产品规则，比如说**产品补货规则（**这个规则规定了如何以及何时补充产品库存）。尽管支持**产品规则**的业务规则模式更加的泛化全面 — 因为它需要包含因素、成效以及所有**产品**规则类型的分类，但是它将会很多相同的因素，而且该模式的结构业的结构看起来也与维护价格成分规则的模式非常相似。

在图8-4中，价格成分规则的成效是作为属性的，也就是，价格金额、折扣金额、附加费金额、折扣率和附加费百分比。

**（图）**

**图8-4** 二级业务规则模式应用示例，价格

* **价格金额**维护的是产品或产品特征的价格，是申请折扣或附加费后的价格。
* **折扣金额**维护的是一定金额的折扣，可以用来削减价格。
* **附加费金额**维护的是一定金额的附加费，可以用来涨价。**货币类型**可以维护价格、折扣和附加费金额是属于“美元”还是“欧元”或者是其他类型。
* **折扣率**和**附加费百分比**维护的是价格金额的调整度，它是价格金额的一部分。金额和/或百分比的值是根据相关实体的因素组合而得出的。

如图表8-2所示，产品“A23 Widget”的价格金额是“10”（美元）；折扣金额是“1”（美元）以及附加费是“0”（美元），并且这个是依据因素的组合：地理范围是“北美”，订购日期是“2009年1月1号”之后。

在图8-4最后一行，表8-2的最后一列，实体价格成分规则类型中的名称属性对规则的实例进行了分类。前两个实例被归类于“净价（NetPrice）”，因为该规则是关于根据不同的因素制定什么样的价格、折扣和附加费。最后三个实例被归类于“折扣”，因为这些规则仅仅描述了一条折扣规则，而该规则只会产生两个成效：折扣百分比或折扣金额。

**注：如果想更深入的了解定价规则，请参看数据模型资源手册卷1 （修订版）中的第3章节（Wiley. 2001）。**

需要注意的是所有的成效和某些属性会当成属性来维护。由于在任何特定实例中都不存在几个价格或百分比属于同一类型属性，并且这些属性都是不可分割的数据块，他们的值一般情况下是不会被复用的，所以我们没有使用独立的实体来捕获金额或百分比，比如说**价格成分规则成效**，如图8-2所示。然而，你可能会提出这样一种论点，由于金额还有货币、额度和金额货币类型标识这些特征，我们可以将他们分解成独立的规则成效实体。还需要注意的是有些因素是要作为属性的，也就是有效起始日期和有效终止日期（这两个属性是因素属性1和因素属性2（图8-2所示的模式） 的示例）。这些属性表示某一特定**价格成分规则**有效时间长度。表8-2的前两行所示的价格是以某些日期为基础和依据的。表8-2中另外一个价格成分规则示例是：‘10（美元）’的折扣金额是“盛夏促销（summer saver）”计划的一部分，该计划的有效日期是2009年6月至2009年9月。由于业务规则通常都有有效时间段，所以将这些时间段因素作为规则的属性。

**表8-2** 使用二级业务规则模式的范例，定价

**（表）**

**注：有效起始日期与有效终止日期与起始日期和终止日期是不同的。起始日期和终止日期是展示价格成分规则实例从开始到结束的有效时间长度属性，也就是说，这个实例第一次存在的时间到实例最终有效时间。有效起始日期和有效终止日期展示的是价格成分规则实例有效运作时间。例如，对于皮毛第一可以制定一条折扣规则，这个规则值夏季的时候（有效起始日期和有效终止日期）应用，但是在夏季到来之前的数月以及之后的数年中（起始日期和终止日期），折扣规则本身就可以存在。**

你可以看到价格成分规则因素中可以包含其他的因素，而这些因素不是坎特威茨零售企业数据模型的一部分。在表8-2的最后一行规则因素类型中名称为“从配送中心支最远交付地点的最大英里数”，代表的是规则的其他的因素实例。这个因素没有像其他因素一样直接关联到现存实体的值。这个因素表示只有当从分配中心到交付地点不是太远的时候，价格成分才能生效。有些规则因素类型要求与因素相对应的值，如果是这种情况的话，需要指定的值是英里数，也就是从分配中心到最远交付地址的英里数。例如，价格成分规则因素中的因素值是“500”，这表示距离分发中心500英里之内的所有交付地点都可以享受折扣价。这样便可以维护任意数量的其他因素，然后将他们应用到业务规则中。

有些价格成分规则实例可以包含实际的规则声明[rule statement]。表8-2的最后一行展示了这样的实例。该实例的规则名称[rule name]是“定价规则124”，这个特殊业务规则表明（在规则声明[rule statement]中）“只有满足以下条件才能享受25%的减价”:  
1、客户居住在北美

2、客户是合作伙伴

3、产品类别是硬件或配件

4、订购项超过1000

5、订单价值在$1,000,000或以上

6、客户的交付地点到最近的分发中心的距离必须在500英里范围之内。

7、价格的有效日期是从2009年的8月至2009年9月。

规则表示如果满足以上所有天剑，即可应用**价格成分规则**中的“25”的折扣率（成效）。

使用**产品成分规则**维护大量实例，是因为任意因素的组合都会影响价格的金额或百分比。例如，产品“A245”的价格（价格金额），在纽约州该产品的要价是$9.99，在加利佛利亚的要价是$10.99，英国的要价是€99.9,香港的要价是HK$99.99。通过应用特定或泛化的因素，可以调整这些价格。所以你可以说对于纽约州的“合作伙伴”（**角色类型**），当其**订单价值**超过“$1,000”（美元）时，该产品的价格可以降至“$8.99”。可能会有价格成分规则属于“附加费”价格成分规则类型，对于距离配送中心500英里之外的任何交付地点要收取金额为“$100.00”的附加费（规则声明示例中的第6点）。

**注：在这个示例中需要注意的是不是所有的价格成分规则实例都与产品或产品特征相关。例如，定价规则124就没有与某一特定产品或产品特征相关。有些折扣只应用于某一特定产品类别、角色类型和/或某一特定的地理范围，而且没有与特定产品或产品特征相关联。那么在这些实例中，产品类别、角色类型或地理范围变成主体了吗？从某种角度来说，的确如此。然而，审视这种模式的另外一种方式是将所有的主体视为潜在因素，并将所有的因素视为潜在主体。因此在这个示例中，我们可以将产品和产品特征视为影响成效的因素，正如影响成效的其他因素一样。**

应当在何时使用这种模式？

我们在以下情形中使用这种模式：

* **当企业关注如何将特定主题领域的业务规则集成到他们数据模型（或企业数据模型）中的其他实体当中时。**通常情况下，业务规则与数据模型其余部分相关联。在第一个实例中，你可以看到**事件类型规则**影响**事件类型**（比如说“客户投诉”或“客户周年纪念”）和**沟通事件类型**的发生，比如说“打电话”或“发邮件”可能会产生两个成效“致歉”或“解释”。这方面的知识对企业而言是很重要的。例如，如果企业添加一个新的**事件类型**或**沟通事件类型**，它可以维护相关的业务规则并将这些规则集成到数据模型的上下文中。
* **需要灵活的数据模型结构来获取规则。根据各种因素的组合，模式可以容纳大量的可能的成效以及可能会出现的其他的规则因素类型。**为了能够应用任意数量的其他泛化因素，此模式使用了**实体规则因素**和**规则因素类型**实体。例如，在图8-3中，你可以依据其他的因素，比如说“投诉性质”或投诉“严重性”，来添加规则
* **企业没有获取规则的元数据环境或业务规则引擎时。**这些模式提供了规则、因素和成效的存储空间。我们在很多公司中实现了这些模式，都是直接使用它们捕获业务规则，而后直接整合到他们的数据库中。
* **企业希望获取其他的没有直接与数据模型中的实体相关联的因素时，或需要适应将来可能会出现的因素类型时。**这就是为何我们还要使用 **实体规则因素**和**规则因素类型**来捕获其他因素的原因。在第二个示例中，如图8-4和表8-2所示，你可以看到 “从分发中心到最远客户地点的最大英里数”这个**规则因素类型**所捕获的因素便没有直接使用现存实体来维护。
* **当需要维护业务规则引擎或元数据存储卡环境的中间方案时。**如果企业（坎特威茨零售）创建集中的业务规则引擎或元数据环境时，那么这种模式可能是有用的权宜之计，它可以让人们在不用等待实现元数据环境的情况下创建规则驱动型环境。实现这种模式时，可以很容易地将它整合到元数据环境中（作为业务规则的来源或目标）。
* **有些时候数据专业人士需要向其他IT专业人士展示声明范畴时。**这种模式展示了业务规则适用领域的范围。不同的新类型的业务规则可能需要一组不同的因素、成效等等。展现这些不同之处是很重要的。

**这种模式有哪些缺陷？**

这种模式的缺陷如下：

* **很多数据专业人士以及几乎所有的非技术人员在理解这种模式是很可能都会面临困难。**这种模式分别适用于以下两种专业人士：对业务规则具有很好理解的专业人士；为了获取实现业务规则并将其作为数据模型和数据库的一部分而寻找另外一种方案的人士。
* **如果发现新因素，数据专业人士通常很容易只将该因素添加成规则因素类型，而不尝试发掘企业数据模型是否将这个因素作为单独的实体。**这种模式很容易让数据建模人士变得‘懒散’，而且这种模式没有整合各因素与现存实体的关系。为了查看因素是如何整合到现存数据模型当中的，对影响业务规则的每个因素做严格的评估是重要的。
* **需要更灵活的规则分类方法。**这种模式使用了二级分类模式，如果规则有很多种分类的话， 那么你可能需要使用三级分类模式。可以将这点看成分类模式的缺点而非业务规则模式的缺点。因为你可以在业务规则模式中替换更特定或更泛化的分类模式以适应你的特定需求。
* **很多种数据是可以为各种业务规则所共用的，**因此可以将这些数据统一到业务规则父类中。我们将在下一部分讨论这一点。

**简介**

二级业务规则模式可以让数据专业人士在数据模型中获取多种业务规则。可以将这种模式分解成三个主要成分：规则、因素和成效，如图8-2所示。

在模式的规则成分中，你可以维护**规则名称**、**规则声明**、业务规则分类、规则的主体、因素和成效。你还可以维护巩固规则定义的其他属性，比如说**规则来源**、**规则注释**、**规则引用**等等。

因素的定义是“影响谷子额成效的条件”。可以将因素作为规则实体的属性、现存实体的关系或当前现存实体未维护的但适用于各因素实例的泛化因素来维护。例如，在图8-4中，价格成分规则获取了有效起始日期[effective from date]和有效终止日期[effective thru date]这两个因素属性，这两个属性表示了成效有效的时间范围，比如说价格金额[price amount]或折扣率[discount percentage]的有效时间范畴。因此，可以使用他们指明夏季折扣或冬季特惠。

因素可以关联到现存实体，比如说某个与现存实体**地理范围**相关的因素，因此，你可以复用这个实体所维护的值。例如，你可以制定一个规则规定：源自“南美”（**地理范围**）的订单给予2%的折扣。

此外，你可以使用**规则因素类型**和**实体规则因素**获取未与现存实体相关联的因素。这些因素在特定业务规则的上下文中具有重要意义，但是没有明确地将这些值作为独立的实体或属性来维护。数据专业人士不应当变得‘懒散’，使用**因素类型**获取所有的因素。我们建议你认真理解每个影响规则的因素。

随着企业的拓展，这些因素会随着时间的推移而演变，数据模型中因素和其他数据也同样会发生变化，这样演变应当在业务规则中体现出来。因此，在数据模型中捕获业务规则表示业务规则应当随着数据模型的演变而变化（反之亦然）。此外，将业务规则整合到数据模型中，你可以推断变化会对数据模型会造成什么样的影响。例如，如果从数据模型中删除**地理范围**会发生什么呢？这样做会影响哪些规则呢？

在成效成分中，这个模式维护了业务规则不同的可能成效。你需要理解并捕获业务规则的成效，因为这个成效可能会影响在不同的情形中应当采取的措施，也就是说，要依据不同因素。与因素相似，这种模式可以使用三种方法获取成效：作为业务规则的属性，作为数据模型中现存实体，或者是作为泛化的**实体规则成效**的实例。

这是种更加复杂的模式，适合较高级的数据专业人士使用，在他们将业务规则整合到数据建模工作中的上下文之内，试图解决业务规则问题时，可以使用这种模式。使用这种模式并不意味着你替换企业中的元数据解决方案或业务规则引擎解决方案，但是我们发现很多企业并没有实现完整的元数据方案或业务规则引擎方案，而且/或者他们并不打算实现这些方案。这种模式有助于企业理解业务规则，还能帮助其将业务规则整合到数据模型中（和数据库中）。即使企业正在制作元数据方案（或集中业务规则引擎），你还是可以将这种模式作为非常有用的方法，应用它为业务规则建模，因为业务规则是数据建模工作不可或缺的一部分，比如说产品定价规则。我们发现数据建模人员通常都会为规则建模，甚至是熟悉的规则。因此，如果你认识到这是个‘if/then’情形，那么你可以使用这种模式，它可以高效一致地位这类数据需求建立模型。

**三级业务规则模式**

从先前小结中的图8-3和图8-4中可以看到两个二级业务规则模式的示例。在这些图中，每个‘规则’实体（**事件类型规则**和**价格成分规则**）都是使用了独立鲜明的实体来维护规则、因素、成效和分类。然而，这些数据模型结构具有相似之处，而在单个同一的模型中维护规则是非常有利的。三级业务规则的实质是使用同一个数据模型维护企业中的所有规则或企业中的某一主要领域的规则。

**为什么我们需要这种模式？**试想下使用二级业务规则模式，数据模型中可能存在大量的业务规则实体。三界业务规则解决企业这样的需求：在同一个数据模型中维护多种业务规则。为了适应多种业务规则以及随着时间推移而产生的变化，这种模式采用了非常灵活的格式。为了获取整个企业内的所有业务规则或某一领域的业务规则，比如说事务的所有规则或当事人的各种业务规则。当企业获取这些规则时，可能需要获取不同规则的通用方面，比如说业务规则声明、泛化的因素、分类和成效，除此之外还要获取某类规则的特定方面，比如说影响此类规则的不同的特定因素。为了做到这一点，这种模式提供了一致灵活的方法。

**注：我们发现在获取某一特定领域的所有业务规则时，这种模式很是有用。一个领域的很多的业务规则使用相同的因素。例如，在投资银行的交易环境中，使用这个通用结构你可以为交易获取所有的限制规则、市场风险规则和信用风险规则。在同一领域中，对于规则可以如何关联以及哪些因素通常会影响哪些规则这样的问题，这种模式提供了很好的指示。**

**这种模式是如何运作的？**

正如上一节所示，大多数规则都存在需要建模的通用成分：规则名称、规则声明、业务规则声明、特定因素、泛化因素以及规则成效。在这个模式中，我们采用更加泛化的方式为这些通用成分建模，这种模式提供了更多的灵活性，在相同的数据模型中能够维护多种规则。大多数的规则还存在只能应用于某类规则的特定实体或属性。例如，**价格成分规则**是基于**订单价值**的，这可能是特定于此类规则的关系。在这种模式中，我们提供了一种方法，该方法既可以为某一特定业务规则的特定方面建模也可以为应用于所有规则的通用方面建模。

在图8-5中，你可以看到名称为业务规则的子类。这个**业务规则**实体支持可应用于整个企业或企业的某一领域的各类规则这样的数据需求。这些规则可以有任意数目的主体，而这种模式阐明了这些主体，比如说**实体1**、**实体2**和**实体3**。**业务规则1**和**业务规则2**是**业务规则**的子类，他们与**实体1**、**实体2**和**实体3**相对应。例如，这些实体可以是指二级模式中所示的实体：**事件类型规则** （指的是针对事件类型的规则）和**价格成分规则**（指的是针对**产品**和**产品特征**的规则）。也可以有任意数目的**业务规则**子类，例如，**订单规则**、**工作计划规则**、**装运规则**、**当事人规则**、**产品补货规则**、**产品常规规则**和销**售制约规则**等等。这些子类都可以与数据模型中的各个核心实体相关联。

在图8-5中，还可以看出**实体2**既是**业务规则1**子类的主体也是**业务规则2**子类的主体，这说明了一个实体可以与多个规则相关联。例如，**产品**或**产品特征**可以与**价格成分规则**和**销售制约规则**相关联。换而言之，**产品**可以具有指定价格的规则，还可以具有这样的规则：指定你的产品可以销售到哪些国家和/或销售给哪些年龄段的群体。

在图8-5中，你可以看到**实体3**通过两种不同的方式与**业务规则2**相关联。第一，它是规则的主体。它也可以是规则的特定成效。例如，项目（一个**工作计划类型**）可以是审计所有项目总耗资规则的主体。如果某个项目超支，那么可以将规则的成效设置成“审计项目”（另外一个工作计划类型），可以调查项目超支的原因。事实上，我们可以试想实体是规则的主体，影响规则的一个因素，还可以是某一规则的特定成效。

（图）

**图 8-5** 三级业务规则模式

**注：这种模式可以包含规则与诸多实体主体相对应的多种业务规则。这种模式可以是非常有用的，这取决于贵企业的规模和复杂度，但是伴随着因素、成效和业务规则实体子类之间的诸多复杂关系，这种模式会变得非常庞大。为了保持这种模式可维护性，有时我们将这种数据模型结构用于模型的每个主要领域或区域。例如，我们可以具有如下业务规则实体：当事人规则、产品规则和事务规则。当事人规则可以覆盖所有当事人规则、当事人角色规则、当事人关系规则以及其他以当事人为导向的规则，产品规则可覆盖所有与产品主题数据领域相关的规则。事务规则可以覆盖所有关于订单装运、发票、支付以及其他类型的事务的规则。为了便于管理，减少模型中实体数量时，这个方法有助于平衡具有较特定风格的三级模式（与二级业务规则模式相比较）。然而，当企业认为使用较泛化风格的建模方式更好时，可以使用这种模式维护企业内的所有规则。**

**业务规则**实体包含了业务规则可以具有的所有通用属性和关系。例如，业务规则中有维护规则名称属性，用来获取规则的名称，比如说“定价规则124”，还有规则声明属性，以简明的语言捕获完整的业务规则，比如说“在客户提交的投诉有效的时候，必须发送致歉信”。在业务规则内部，你可以看到两个子类（**业务规则1**和**业务规则2**），这两个子类代表的是你想要捕获的特定类型的业务规则。这些子类可以有自己的特定关系和属性。例如，如果**订单**和**装运**是相应**实体1** 和**实体2**的主体，那么业务规则可以有**订单规则**和**装运规则**这两个实体。

注：根据经验，我们发现通常业务规则子类所特有的数据与那些代表特定因素、特定成效的实体相关，还与表示规则主体的实体相关（比如说图8-4中的**产品**和**产品特征**）。 然而，情况也并非总是如此。有些业务规则子类也可以有他们自己的属性，特别是自己的因素属性。一个示例便是之前的**利用率计费规则**中的数量属性（为了了解这个层级用途的价格，实体展示了用途出现的次数，例如文本信息的字数），**利用率计费规则**实体是**价格成分规则**实体的子类。这表示了在**业务规则**实体中，子类还可以包含子类。请参看数据模型资源手册卷1（修订版）的第三章节中的图3-7。

在图8-5的左边，你尅看到两种因素：“特定因素”和“泛化因素”。位于模型图左上角的特定因素与你数据模型中现存实体相关联。正如我们先前所说，捕获现存实体与业务规则相关联的方式是很重要的，这样你就不会冗余地捕获那些已经存在于数据模型当中的数据。例如， 如果某个因素与地理范围“纽约”相关联，那么你应当将该业务规则关联到已有的地理范围实例—“纽约”上，而不是在另外一个实体中（比如说，在**规则因素类型**中）冗余地重新创建这个数据。因此，特定因素与现存实体相关，关于这一点，图8-5展示了诸多示例，在该图中“每个**实体4**可以是一个或多个**业务规则1**的条件”。再例如，“每个**业务规则2**只能依据一个**实体5**和/或**实体6**”。

**业务规则2**到**业务规则实体8 因素**的关系展示了这种模式可以如何支持规则到某一特定因素的多对多关系。例如，假设价格成分规则依据同一特定因素的两个实例，比如说给予要么是“合作伙伴”要么是“供应商”的当事人折扣率（这个规则是基于两个角色类型实例）。如果**业务规则2**是**价格成分规则**，而**实体8**是角色类型，通过将**价格成分规则**以**业务规则价格成分因素**关联实体的两个实例— “当事人”角色类型和“供应商角色类型”为依据这种方式，这种模式可以支持这个条件。

在模型图的左下角，可以看到“泛化因素”。这些是没有与数据模型中现存实体相关联的因素，并且这些因素需要维护其他类型的因素以及规则的可能值。分别将他们作为**业务规则因素**、**规则因素类型**和**因素值类型**的实例。

需要注意的是业务规则因素是与业务规则子类相关联，而非与业务规则的特定子类相关联，因为这是维护各类业务规则因素的泛化结构。模型表明，“每个**业务规则**可以基于一个或多个**业务规则因素**”而“每个**业务规则因素**必须只能成为一个**业务规则**的条件”。这些**业务规则因素**根据**规则因素类型**来分类，并且他们可以使用一个或多个**规则因素值**，而每个**规则因素值**则可以根据**因素值类型**来归类。例如，规则因素类型的一个实例可以是当事人作为企业客户的时间长度（“当事人成为企业客户已有若干年限”）。这是个泛化因素的示例，不能（或不需要）在数据模型的其他地方维护这个示例，因此你可以在此维护它。这个业务规则因素可能会与**规则因素值**实体中因素值为“10”的属性相关，这表示如果当事人是十年的老客户，就会应用这条规则。一个规则因素（业务规则因素）可以有若干个可以应用的值（规则因素值）。当事人作为客户的年限是否在5年至10年之间，你或许需要根据以上因素来影响价格。对于这个因素，你可以维护名称为“当事人作为客户的年限”的**规则因素类型**实例，此名称属性有两个值，分别是“5”和“10”，“5”这个值是使用**规则因素规则**实体的因素值属性来维护的 ，这个值在因素值类型实体中的名称是“最小年限”；另一个值“10”也是使用规则因素值实体中的因素值属性来维护的，但是它所属的因素值类型是“最大年限”。

**注：根据我们的经验，很多泛化因素都有一个单一值。此外，一些具有多个值的因素可以通过指定多个因素实例来管理。例如，在先前的示例中，我们可以有两个规则因素类型，分别是“作为客户的最低年限数”和“作为客户的最大年限数”。因此，另外一种数据模型结构是在业务规则因素实体中添加因素值属性（如图8-2），而不是维护规则因素值实体。然而，图8-5中我们为泛化因素建模了，这种方法提供了更多的灵活性，可以使用诸多不同场景。**

注：模型中使用泛化规则因素类型和业务规则因素的一个问题是并不是所有规则因素类型都可以应用于业务规则。这种模式兼顾了泛化**规则因素类型**和**业务规则因素**所提供的灵活性和在将**规则因素类型**与**业务规则**相关联时所需的特定性。例如，泛化因素类型“当事人作为客户的年限”没有与**装运规则**或**账单规则**相关。我们所在的企业执行两个IT业务规则，分别是“所有特定的规则因素必须在数据模型中使用独立的实体”和“泛化规则必须影响所有的业务规则”。这导致了大量额外的实体，而这些实体只是用来获取规则因素，并且**规则因素类型**和**业务规则因素**实体会非常稀少！这不一定是最佳解决方案。

与因素相似，在模型图的右边是两类成效：“特定成效”和“泛化成效”。正如因素可以应用到模型中的现有实体一样，成效可以与模型中的现存实体当中的数据相关联。例如，常见成效是在某种情况下应当设置某一类型的工作计划。所以业务规则可以表明：当收到非常重要的投诉时，应当设置一个“管理投诉”（工作计划类型）的工作计划，用来追踪投诉的最终处理成效。在这种情况下，规则成效（比如说**事件类型规则**）可以是特定的成效，我们可以通过将规则关联到现存数据模型实体—**工作计划类型**这种方式来维护该规则成效。

特定的因素与规则之间存在多对多关系，与这一理念类似，特定的成效与规则之间也存在多对多关系。例如，当收到非常重要的投诉时，可能会存在需要为成效创建多个工作计划类型的业务规则，比如说创建工作计划类型“管理投诉”和“评估流程（导致投诉）”。关联实体**业务规则实体7成效**说明了容纳规则的数据模型结构可以具有多个同类型的特定成效。

在图8-5的右下角，使用以下实体维护业务规则泛化成效：**业务规则成效**、**规则成效类型**、**规则成效值**和**成效值类型**。这反映了泛化因素的数据模型结构，这个结构可以维护各种业务规则成效，每个成效可以有多个相关的值。因此，**业务规则**直接关联到**业务规则成效**的话，可以灵活地添加任意数目的泛化成效，也就是说，“每个业务规则可以产生一个或多个业务规则成效。”重要的是，我们只能说规则成效指的是应当会发生的情况而非实际发生的情况。因此，只是陈述或描述一个规则，比如说“对于有效投诉应当致歉”，未必保证这个事件真实发生了。换个说法，就是维护规则的目的只是陈述应当会发生什么而非已经或将来一定会发生什么。

还是与因素相似，每个规则实例可以有很多的成效，而每个成效可以有很多的值。例如，正如表8-3中的第二个规则实例所示，业务规则表明在客户庆祝他们十周年纪念（一个因素）的时候，会产生两个成效：一封感恩/致谢信件和一个礼物。至于礼物成效，可能会有两个值，比如说收到“2”套钢笔套装和价值“50”美元的优惠券。因此，业务规则成效可以根据名称为“感恩致谢信”的规则成效类型来分类，而另一个业务规则成效则是根据“礼物”这个规则成效类型来分类的。至于与“礼物”相关的业务规则成效有两个规则成效值，一个是“2”，另一个是“50”，这两个值都是使用规则成效值实体中的成效值这个属性来维护的，但是“2”这个值的成效值类型是“笔类套件”而“5”的成效值类型则是“美元优惠券”。使用这个结构，当结果的参数随着时间发生变化时，企业可以保留同类型的结果，但是可以修改变更的成效值（改变优惠券金额或礼物类型）。

还要注意的是同一个业务规则实例既可以有特定结果也可以有泛化结果，就像它还可以具有特定因素和泛化因素一样。例如，“客户投诉”事件类型可能会产生特定结果“管理投诉”（**工作计划类型**），还可能会导致泛化结果 —“送客户打折优惠券”。

注意由于这是一种更加泛化的数据模型解决方案，所以我们决定不将因素或结果捕获成特定的属性，比如说图8-4所示的有效起始日期、价格金额或折扣率，相反，我们使用了泛化因素实体（**业务规则因素**、**规则因素类型**、**规则因素值**以及**因素值类型**）和/或泛化结果实体（**业务规则结果**、**规则结果类型**、**规则成效值**以及**成效值类型**）。这样的模型会更加一致、灵活，因为我们使用了相同的格式来维护因素和/或结果，并且为了容纳那些随着时间推移可能会出现的任何其他因素或实体，我们可以使用这个结构。例如，表8-2第一行中的有效起始日期“2009年1月1日”，现在表8-3的第三个业务规则将其捕获成名称为“有效起始日期”的**规则因素类型**。另外一个示例是具有很多因素的规则，它的结果是“25%”的折扣率，如图8-4和表8-2所示，在业务规则结果中维护这个结果，它的规则结果类型名称是“折扣”，在实体**规则成效值**中的成效值是“25”，其所属成效值类型的名称是“百分比”。

**注：使用这种模式还可能在业务规则子类中（比如说价格成分规则）保留特定的属性结果，比如说价格金额（以及任何其他因素或结果属性），而使用泛化成效维护其他的成效。然而，这意味着使用不一致的方法处理企业中的因素和成效。有些规则成效可能是在业务规则成效中维护的，而有些则是作为业务规则父类或子类的属性。就像有些因素是在业务规则因素实体中维护的一样。这样做可能会让开发人员感到很困惑。但是从语义上来说它可能更加准确。一如往常，它起到了平衡灵活性和语义严谨性的作用。同样的逻辑也可以应用于像有效起始时间这样非常特定的因素。**

三级业务规则模式与二级业务规则模式的不同之处在于它应用了三级分类模式，如图8-5所示实体：**业务规则类别分类**、**业务规则类别**和**业务规则类别类型**。这种模式允许采用多种方式对规则进行分类。在二级模式中，只用一种分类方式，也就是，实体规则类型。表8-2展示了价格成分规则的实例可以是“净价”（用于维护扣除折扣和附加费后的价格）和“折扣”（用于维护折扣，可能是一笔固定的金额也可能是百分比）。然而，假如你也希望实例的不同分类该怎么办呢，假如说该实例可以分类成“一次性费用”、“常规费用”或“使用费用”？例如，需要使用多种方式对价格成分规则实例，比如说根据 “价格成分性质”这一**业务规则类别**来分类的话，可以分为“价格”、“折扣”和“附加费”，这些类别都是**业务规则类别**的实例，根据名称为“价格成分频率”的**业务规则类别类型**来分类的话，可以分类成“一次性费用”、“常规费用”和“使用费用”，将这些分类作为**业务规则类别**的实例。这样你可以指定一个规则，规定**产品**的月费既可以归类于“价格”这一**业务规则类别**，也可以归类于“常规费用”这一**业务规则类别**。因此，我们可以使用两个**业务规则类别分类**实例来维护这些分类。

思考下先前小节中的示例，在该示例中当业务规则与**事件类型**和**价格成分规则**相关时，要求数据专业人数捕获这些业务规则。在数据专业人士制作了图8-3和图8-4之后，坎特威茨零售想要以同样的方式捕获整个企业中的所有业务规则。为了展示这样做的可行性和有效性，数据专业人士依据三级业务规则模式（图8-5所示）将图8-3和图8-4整合到图8-6中。

在图8-6中，你可以看到当前事件类型规则和价格成分规则是业务规则的子类。所以表8-3 的前两行是**事件类型规则**子类的规则实例，后面两行是**价格成分规则**子类的实例。业务规则实体包含了通用属性规则名称和规则声明，在表8-3的第二行和第三行分别展示了这两个属性。你可以看到事件类型规则子类有两个规则名称和规则声明。例如，第一行中，规则名称是“投诉规则100”，规则声明是“当客户拨打电话提交有效投诉时，应当给顾客拨打致歉电话，而且必须设置一个管理这项事件的工作任务”。那么价格规则成分子类将会有两个规则，其中一个规则（表的倒数第二行）没有规则名称或规则声明，用于处理产品的价格。价格成分规则子类的第二个规则的规则名称是“定价规则125”，规则声明是“当满足以下条件时，给予25%的优惠”：  
1、客户居住于北美

2、客户是合作伙伴

3、产品类别是“硬件”

4、订购项超过1000

5、客户支付时间平均在66-90天内

6、日期在“2009年10月1号”之后

**表8-3** 使用三级业务规则模式示例，定价

**（表）**

**（图）**

**图8-6** 使用三级业务规则模式示例，沟通事件类型和定价

**注：“定价规则125”是我们在表8-2中所介绍的“定价规则124”的新的迭代，它改变了规则的三个方面：删除了最小订单值的限制，它将泛化因素从交付地点因素变成了与客户平均支付时间相关的因素，它改变了有效日期。这样的变化引发了一个有趣的问题：这是个新规则？还是现存规则的修订版？我们建议企业使用参数定义何时可以将更新数据的规则看成是同一个规则实例，什么时候将修订版规则可以是另外一个实例；什么情况下它构成的是全新的规则。此外，如果规则是修订版的，或许业务规则会需要递归关系。这个模式的另外一个优化是在这个模式中可以应用第四章节中的递归模式，这样便可以展现出规则之间相互关联的方式（例如，当一个规则是另一个规则的修订版时）。**

表8-3展示了**事件类型规则**和**价格成分规则**的特定因素与泛化因素、特定成效与泛化成效。例如，第一行处理的是事件类型规则，该规则的名称是“投诉规则100”，这一行展示一个特定因素，也就是该规则可以应用到名称为“打电话”（因为这个因素与数据模型中的特定实体相关）的**沟通事件类型**中。“有效的”这个因素（也就是说这个投诉是有效投诉）被当成泛化因素，因为这个因素没有与现存实体中的数据项关联，而只是用于处理规则。表8-3的最后一行展示了规则名称为“定价规则125”的**价格成分规则**，这个规则有若干个特定因素，这些因素是基于数据模型中已存在的实体的数据，比如说**地理范围**、**角色类型**等等。这个实例的规则声明表示我们需要容纳新的因素—关于“客户支付时间平均在60天到90天之内”，因此，业务规则因素具有两个值，一个是“60”，另一个是“90”，这两个值是使用实体**规则因素值**中的因素值属性来维护的，分别表示“最小可行支付平均天数”和“最大可行支付平均天数”。

同样地，事件类型规则和价格成分规则有特定的成效和泛化的成效。在表8-3中，第一个示例是特定的成效 — “管理投诉”和“管理周年纪念事件”，因为他们是特地与数据模型中的现存实体相关联的成效，也就是实体工作计划类型。这些规则表明一旦满足这些因素，会产生成效，设置工作计划：“管理投诉”和“管理周年纪念事件”（**工作计划类型**），当发生客户投诉或客户周年纪念事件时，设置工作计划有助于确保对所需采取的措施做一个全面的分析。

表8-3还展示了很多的泛化成效，比如说针对事件类型规则会有以下成效：“道歉”、“感恩致谢信”和“礼物”，针对价格成分规则会有“价格”和“折扣”。还要注意的是有些业务规则成效可以与特定的实体相关联。在图8-6和表8-3中展示了**事件类型规则**实例产生了**业务规则成效**，这些成效是通过**沟通事件类型**来交流的。例如，第一行的业务规则成效是通过**沟通事件类型**“打电话”来沟通的（在表格的括号中）。

为什么使用同一数据模型维护不同类型的规则？为什么使用同一个数据模型处理**业务规则**、**业务规则因素**和**业务规则成效**。捕获每个业务规则子类的所有特定因素和成效，可以深入了解不同的规则集合之间是任何相互关联的。例如，如果你发现**价格成分规则**和**成本成分规则**共用以下实体：**地理范围**、**当事人类型**、**产品类别类型**和**订单价值**，这可能表示类似的规则应当或可以在两个地方使用。在一个地方捕获所有的业务规则因素和规则因素类型，你可以发现不同规则因素之间的协同作用。例如，你发现“有效的”和“无效的”这两个规则因素类型被用于很多不同的业务规则子类，那么这表示根据他们在企业各部门的业务规则中的重要性，事实上他们应当作为独立的实体。

使用一致的方法维护因素和类型也具有很大的好处，使用非常灵活的数据模型结构可以容易地容纳任何新出现的因素或成效。在一个地方捕获所有的业务规则相当于获取了不同的规则字典，而不管他们适用于业务规则父类中哪些规则。作为元数据存储计划（或业务规则引擎计划）的一部分，这样可以更有效地对规则进行分类和下定义。

但是，公平地说，对于一些受众来说，在一个单一的模型中捕获所有的业务规则可能是抽象的难以理解的概念。很多**业务规则**子类可以复用同类型的因素，比如说**价格成分规则**和**成本成分规则**。试想下其他四个规则也分享这些因素。**地理范围**、**角色类型**和**产品类别**通常是被很多种业务规则所共享的因素。么讴歌因素到每个业务规则子类的关系使得这个模型变得更加复杂。也存在很多通用类型的成效，如示例所示。一个恰当的例子是很多的业务规则都可以共享工作计划类型的成效。但是，你可以单独展示每个**业务规则**子类的数据模型视图，这样有助于理解。另一件要考虑的事情是：与特定因素属性、成效属性建模相反，当你使用该类的因素和成效实体维护这些值时，模型更加难以理解。例如，通过此模式中的各种泛化因素实体来记录一个简单的有效起始日期（和图8-4所示的二级模式示例的做法一样）会需要更多的导航。

因此，尽管这种模式更加难以理解，但是使用一个单一数据模型来管理所有的业务规则可以帮助你更好地理解规则，发现规则之间的协同作用、减少捕获规则方式的臆测，最终有助于节省为每个业务领域的制作规则解决方案所花费的时间和精力。

应当在什么时候使用这种模式？

我们在以下情形中使用这种模式：

* **为了维护数据模型的一致性和简便性，对使用同一个数据模型维护业务规则感兴趣的企业。**使用标准的数据建模格式获取各种业务规则，这样提供了一致性，使得业务规则和模型的管理变得更加容易，而且降低了这种可能性：对于不同类型的业务规则使用不同的约定。因为在这个模式中使用了单一的成效来处理业务规则，所以需要维护的实体要少得多。如果需要改变影响所有规则的业务规则模型，那么可以在一个地方更新模型，而不像二级模式那样，需要更新每个业务规则建模结构。
* **正在创建元数据存储库或规则引擎的企业。**使用这种模式将所有的业务规则放在一起，这种方式可以有效的对不同的业务规则、业务规则的因素和成效进行分类。此外，通过这种方式，你可以展示创建元数据存储库时所需的各种规则的广度和深度。你还可以展示通用的因素和/或成效，这些因素和成效可以应用于多个业务规则实体。
* **为了帮助实现业务规则，需要以一种通用的方式来处理所有的业务规则。**试想下程序员创建一个标准接口，用于管理可以复用的业务规则因素和业务规则成效。只有当该代码的底层是通用的、稳定的结构时，才可以做到这一点。对所有的规则使用一个单一数据模型，有助于创建通用的架构、规则、流程和服务。
* **当需要灵活地维护业务规则时。**当需要其他的业务规则类型、因素类型、成效类型和类别类型时，这种模式添加这些会更加容易，并且不需要改变数据模型。由于这种模式使用的是泛化的因素实体和泛化的成效实体，而非属性因素和属性成效，所以它更加灵活，在需要时可以在不改变模型的情况下，添加新类型的因素和成效。

**这种模式用哪些缺点？**  
这种模式的缺点如下：

* **和二级模式相比，这种模式会更难被很多数据建模人员和非技术人员所理解。**可以通过展现样例数据、提供培训以及将此章节作为教具等方式来解决理解困难问题。
* **在一个单一的模型中维护所有的业务规则，可能会让模型变得很笨重。**当所有的**业务规则**子类都显示在一个包含了所有**业务规则**实体的模型中时，很快你发觉模型变得非常庞大复杂。然而，可以通过以下方式来管理这个模式：为每个业务规则子类创建不同的数据模型视图以及展示整个业务规则模型的关键实体。

**简介**

在这部分我们介绍了如何将不同类型的业务规则‘包装’到一个父类业务规则实体中。这样做的原因是大多数业务规则所维护的基本数据类型相同。例如，所有的业务规则子类都有：

* 业务规则主体
* 通用的业务规则施行比如说规则名称和规则声明
* 特定的和泛化的因素
* 特定的和泛化的成效
* 业务规则分类

在此模式中，我们通过创建业务规则父类展示了三级业务规则模式，该模式使用同一个数据模型结构为通用的业务规则数据建模，它也可以容纳每个业务规则子类的特定方面。这个模式在很多方面都有助于开发元数据存储库或业务规则引擎。第一，它可以帮助展现不同规则的范围，还可以帮助你深入理解各类业务规则的通用因素和成效。第二，它可以对所有需要检查的规则、因素和成效进行分类。最后，它可以是业务规则实现的良好设计基础。这是一种复杂泛化的模式，适用于正在解决各类业务规则整合问题的数据专业人士，例如，整合某一主题领域或整个企业的业务规则。它提供了巨大优势，也就是为各类业务规则建模提供了非常一致灵活的方式。

**维护当事人角色的业务规则**

为了有效使用管理业务规则，模型需要支持当事人在业务规则中所扮演的各种角色。在最后一部分，我们解答一种非常重要的支承结构，数据专业人士可能会发现此结构在支持业务规则所涉及到的各种角色方面是很有用的，比如说谁指定了业务规则、谁管理业务规则、谁是业务规则的来源或谁被授权使用业务规则。

**为什么我们需要这种模式？**

在罗恩罗斯的杰作*业务规则方法的原则*中，他介绍了一些支持业务规则的原则（爱迪生-韦斯利，2003）。我们将会在下一种模式中介绍一些书中所提到的原则，这种模式支持以下原则：

* 应该对规则加以管理
* 授权当事人应当可以访问规则
* 规则应当由那些具备相关知识的人员来指定
* 规则应当只有一个来源。换句话说，企业不应当在不同的部门对同一个规则下不同的定义。规则权威的来源是至关重要的。

**这种模式是如何运作的？**

这四个原则分别解决了业务规则的管理、授权当事人、由授权当事人定制规范和业务规则的唯一来源问题。每个规则都指出了当事人是如何与业务规则相关联的。在第3章节中，我们介绍了三级上下文角色模式，该模式使用了三叉关联实体（上下文角色），灵活地关联了实体、当事人和角色类型。这样可以让当事人随时都可以与任何角色类型、任何实体相关联。在图8-7中，我们应用了与业务规则相关联的三级上下文角色模式

**注：正如我们在本章节以及其他章节中所阐释的一样，模式通常可以应用到其他模式当中。在这一小节，我们选择阐释上下文角色模式是如何应用到业务规则模式中的，在本章节的最后一部分，我们介绍了如何将分类模式应用到业务规则模式中。例如，应用维护规则的三级递归模式又呼吁维护规则之间相互作用的方式，而应用三级状态模式可以维护规则所处的状态，比如说“已提议”、“执行中”或“中断”。如果你正打算开发新的数据模型，我们建议你先查看下有哪些模式可以应用，然后再考虑应用他们。**

**（图）**

**图8-7** 维护相关当事人的业务规则

在图8-7中，每个业务规则可以涉及到一个或多个**业务规则角色**，而每个**业务规则角色**只能由一个**当事人**来扮演并且也只能被一个**角色类型**描述。这意味着这个模型容纳四个角色（以及任意数目的其他角色），这个四个角色分别应用于四个业务规则原则。这些角色分别是“规则管理员”、“授权规则用户”、“规则指定人”和“规则源头”，这些角色被作为角色类型的实例来维护，每个角色的解释如下：

* **“规则管理员”**：规则管理员负责维护业务规则数据，包括规则的声明、分类、因素和成效。换句话说，这个人必须确保指定规则的人保持那些规范是最新的。这个人可以是元数据系统的管理员或负责的业务代表。
* **“授权规则用户”**：这个人可以是有权以某种方式使用这个规则的任何当事人。图8-7中的模型是可以拓展的，添加用来记录了各种情形下的授权和权限的实体，通过这种方式，模型可以展现出哪些类型的用户具有哪些类型的权限。（你可以通过应用本章节中的业务规则模式为授权建模！）。
* **“规则指定人”**：这些人都是“具备相关知识的人员”，他们可以直接指定业务规则。他们可以是那些知道如何有效表述规则、规定规则的分类、因素和成效的业务人员。IT专业人士也可能被认为是规则指定人，从某种意义上来说，为了实现规则他们可以指定数据库和/或应用程序设计。
* **“规则源头”**：规则的起源。提供规则的权威人员或组织（当事人）。

使用这种灵活的上下文角色模式的话，最后一个角色会存在一个问题。业务规则原则表示“规则只能有一个起源”。而这种模式却允许规则有多个起源。为了满足业务规则原则的要求，我们可以应用二级上下文角色模式，将**规则起源**以一对多的方式直接关联到**业务规则**，明确地展示**业务规则**只能起源于一个**规则源头**。然而，我们认为随着时间的推移，一个业务规则也可能具有多个起源。例如，外贸规则起初起源于“欧共体”，后来“欧共体”为“EEC（欧洲经济共同体）”所取代，之后“EEC”又被“EU（欧盟）”给取代了。这三个都是外贸规则的起源，具体情况取决于你查看规则定义的时间。

**注：还有其他方法可以维护业务规则的各种角色，我们建议你看看第3章节中有关的备选方案的讨论。这种模式支持贵企业定义的任何其他的业务规则角色。尽管你不用局限于我们根据罗恩罗斯的业务规则原则而定义的角色，但是罗恩罗斯的业务规则角色是创建相关的业务规则角色的一个很好的指导。**

**应当在什么时候使用这种模式？**  
我们在以下情况中使用这种模式：

* **需要获取随时间推移而充当各种角色类型并与业务规则相关联的当事人。角色类型**可能包括“规则管理人员”、“授权规则用户”、“规则指定人”和“规则源头”；然而，模式支持任意数目的角色、相关当事人以及不同时期的任何**业务规则**。

**这种模式有哪些缺点？**此模式的缺点如下：

* **有些企业可能并不需要这样么多的细节。他们可能支队业务规则的规范以及一两个关键角色感兴趣。**因此，备选方案是应用其他的上下文角色模式，而非使用本章所介绍的三级上下文角色模式。

**简介**

在本章节我们介绍了扮演不同角色类型的当事人是如何与不同时刻的业务规则相关联的。我们介绍了以下几个角色类型实例：“规则管理人员”、“授权规则用户”、“规则指定人”和“规则源头”，这些角色支持这样的原则：只有授权当事人才可以访问业务规则，规则必须有恰当的人员（或组织）来指定，管理，并且只能有一个起源。这个方案是以本书第3章节所介绍的概念为基础的。

**模式总结**

表8-4 包含本章所有的模式简介

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模式 | 描述 | 何时使用 | 何时不使用 | 优点 | 缺点 |
| 二级业务规则模式 | 为业务规则、规则因素和规则成效建立更为特定的模型，每类业务规则都有独特的数据模型结构。 | 需要将业务规则整合到数据模型中，因为他们是数据需求不可分割的一部分并且与其他的数据模型实体相关。  作为各种业务规则的声明范畴。  需要使用灵活一致的数据模型结构来获取业务规则时。  当企业希望成为数据驱动型企业，以动态的方式来维护业务规则时，这样当规则的参数发生变化时，会比较容易更新他们。  当企业对特定主题领域的业务规则是如何关联到数据模型的其他实体上（或企业数据模型）的感兴趣时。  当企业没有使用元数据环境或规则引擎捕获规则时，或者尚未实现元数据环境时，这些模式可以为那些规则、因素和成效提供一个存储场所。  当企业希望能有一个灵活的方式来维护将来可能会出现的其他业务规则因素和成效时，并且这些因素和成效没有与现存的数据模型实体相关联。 | 企业已有集中规则引擎或整合的元数据环境并且已决定在那里维护所有业务规则时，或者在流程模型和那些没有与数据模型相关联的规则字典中捕获业务规则时。  当需要更加一致的方法时，也就是需要使用一个单一数据模型来维护所有的业务规则时。 | 可以灵活的维护规则数据、规则因素和成效。  使用因素和成效的泛化实体，这种模式可以容纳那些随着时间推移而出现的新的因素或成效类型。  是一致的建模方法，可以根据数据需求为各种条件语句（‘if/then’或‘while’）建模。  是一种更加以数据位导向的方法，在数据模型中获取了业务规则的因素和成效。  将业务规则整合到数据模型中，或加深你对数据、业务规则以及规则关联方式的理解。 | 这种模式很复杂，也不是介绍企业业务规则的最佳方法。但是，我们发现有些业务代表与这些模型有着密切关系。  有些数据建模人士发觉如果没有相关的业务规则经验的话，使用这种模型会很困难。  出于一致性和可管理性的考虑，可以将诸多规则实体统一到一个规则实体中。  一个规则值提供了一个分类；但是，使用三级分类模式替换的话，就可以很容易地改变这种情况。  数据专业人士可能会错误地使用泛化因素和成效实体，而不是将因素或成效关联到现存的数据模型实体，这样可能会导致捕获冗余不一致的数据。 |

（续）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模式 | 描述 | 何时使用 | 何时不使用 | 优点 | 缺点 |
| 三级业务规则模式 | 是一种以灵活的方式将所有的业务规则捕获在同一个数据模型中的方法。 | 当企业想要使用单一、通用的数据模型一致地捕获业务规则时。  需要灵活地为企业诸多领域中的业务规则建模时。  当企业正致力于更加全面的元数据存储库或规则引擎时。使用这种方式（将所有的业务规则捕获在一起）可以对不同的业务规则、规则的因素和规则的成效进行有效的分类。  作为元数据存储库的声明范围。  当需要一个单一模型实现通用架构和服务时。 | 作为声明范围或者是收集和/或验证非技术业务代表的业务规则数据需求的方法。  当需要更加特定风格的业务规则模型时 | 提供了非常灵活的模型，可以容纳任意数量的因素、成效和分类，并且在需要新的因素 、成效和/或分类时，不需要更改数据模型。  通过将各种业务规则包装到一个父类中，你可以创建一个单一、一致的方法来维护所有的业务规则，而且这样会让数据模型的维护变得更加简单。  以通用的方式处理所有的业务规则，这样有助于业务规则的实现，这种方式允许通过非常类似的途径来访问业务规则。  将业务规则整合到数据模型中，可以加深你对数据、业务规则以及关联方式的理解。 | 这是种非常泛化的数据模型，对有些受众来说，可能很难理解。  与因素和成效作为属性相比，使用泛化因素和成效需要更多的数据模型导航。  如果发现了新的业务规则分类、因素或成效，数据专业人士通常很容易假定模型已经包含了它，而不做适当的分析。  如果所有的规则都被当成单一模型图中的包罗万象的**业务规则实体**的子类的话，那么这个模型会变得很庞大而笨重。  我们建议将这单一的模型图分解成多个业务规则子类数据模型图。 |

（续）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模式 | 描述 | 何时使用 | 何时不使用 | 优点 | 缺点 |
| 维护当事人角色的业务规则 | 捕获相关业务规则角色的灵活方法 | 当你需要捕获诸多业务规则角色时，比如说管理业务规则的人、指定业务规则的人、授权使用人和业务规则的来源。 | 当你不需要捕获人员或组织在业务规则中所扮演的众多角色时。 | 可以灵活容易的支持任何当事人随时所扮演的任何角色，比如说包括：业务规则的授权角色、指定角色、管理角色以及来源角色。 | 当只需要捕获一两个业务规则角色时，或者是角色的数目和类型不会随着时间的推移而发生变化时，如果是这种情况，使用这种模式实在是大材小用了，这种情况更适合使用一级或二级上下文角色模式。 |

**参考文献**

1源自马尔科姆齐泽姆的 *How to Build a Business Rules Engine: Extending Application Functionality through Metadata Engineering* (Morgan Kaufman, 2004).

2 源自 罗纳德罗斯所著的*Principles of the Business Rule Approach* (Addison-Wesley, 2003).

3 更多关于订单数据模型的内容，请参看Silverston所著的数据模型资源手册卷1，修订版，企业通用建模手册（Wiley, 2001）

4 源自罗纳德罗斯所著的*Principles of the Business Rule Approach*中的“Sentence Patterns for Rule Statements”。

5 源自杰出的数据建模人员，Ed Landale。

6 业务规则物理实现的一个很好的参考资料是Malcolm Chisholm所著的*How to Build a Business Rules Engine: Extending Application Functionality through Metadata Engineering* (Morgan Kaufman, 2004).

7 定价模型的先前版本在L.Silverston所著的数据模型资源手册，修订版，卷1，企业通用数据模型中看到(Wiley, 2001)。