

使用 JRules 进行企业业务 规则管理



使用 JRules 进行 企业业务规则管理

白皮书

© ILOG, 2005 年 2 月 — 未经许可, 不得复制。

ILOG、CPLEX 及其各自的徽标是注册商标。

所有其他公司名和产品名均是其各自所有者的商标或注册商标。
本文档中提供的材料是概述性的, 随时可能更改, 不具契约效力,
只用于一般性参考, 而不构成陈述。

1 简介

近些年来，数据量巨大的组织机构已经越来越清楚地认识到传统软件开发系统所存在的局限性。他们常常发现，主要优势和规范标准信息被锁定在多个软件系统内，并以高深的技术语言来表示，导致负责实施决策的经理和专家们通常无法理解。因此，他们无法做到迅速响应，而在当今不断变化的复杂市场和法规环境中，这一能力是重要的竞争优势。

公司需要能够迅速可靠地改变策略和规则，从而保持持续增长，提高效率，同时符合企业规范。但是，管理应用中业务规则的传统方法既复杂又缓慢，无法实现这一能力。与传统方法相比，企业需要一种解决方案来更有效地支持策略变化，而且还要直接利用策略管理者的专业经验。

企业业务规则管理系统 (BRMS) 正是这样的解决方案。

对策略管理者而言，BRMS 从软件系统中提取策略并放在一个中心位置，人们可以通过熟悉的业务语言在这里管理和编写其规则。企业用户可以根据自己的计划来更新软件系统，而不必考虑 IT 人员。

IT 专业人员也将受益匪浅，借助于 BRMS，IT 人员可以通过企业级技术架构在整个企业内公开业务规则，并灵活地将业务规则与既有的软件开发范例和工具相集成。

BRMS 改变了企业策略管理的性质。例如，新的相关人员可以自行编写和管理业务规则，从而使现有的相关人员具有更高的灵活性，并投入更多的时间改进系统。这不仅产生了企业用户可直接参与的全新业务规则生命周期，同时还完善和扩展了传统的软件开发周期。

ILOG 推出的 ILOG JRules 5.0 是业界第一个也是唯一一个能够满足业务规则生命周期中所有相关人员特定需求的 BRMS，可全面推动企业的业务规则管理。

2 业务规则生命周期与相关人员

2.1 项目目标和生命周期

BRMS 项目的直接目标通常包括：

- 使知识渊博的业务专家能够使用熟悉惯用的语言和语句结构直接编写业务规则或策略。
- 支持业务规则跨时间、产品、辖区、客户和其它领域的高度可变性。

将业务规则生命周期与软件开发生命周期相分离，即可最有效地实现上述目标。这样，规则编写者（即策略管理者）便可独立于软件开发周期进行工作，从而使软件开发周期和规则生命周期可以并行进展。图 1 中对此进行了阐释。在该图上半部分的软件开发周期中，应用版本受到重大需求变化和外部（如应用服务器或 ERP）产品发布计划的驱动。系统设计师、业务分析人员和开发人员按照传统的软件开发周期（包括需求规范、分析、设计、开发、测试和部署）推出新的软件版本。

在大多数情况下，业务规则沿更精细的时间线变化，并受到业务策略变化（代表项目当前发行版的既有功能的变动或扩展）的驱动。图 1 中的下半部分“规则管理周期”中对此进行了说明。这方面的变化要求策略管理者的编写和测试周期更短、更具针对性，并且要求及时部署到生产环境。根据应用的需要，完成这一业务规则周期可能需要长达几个月，也可能只需短短几个小时。

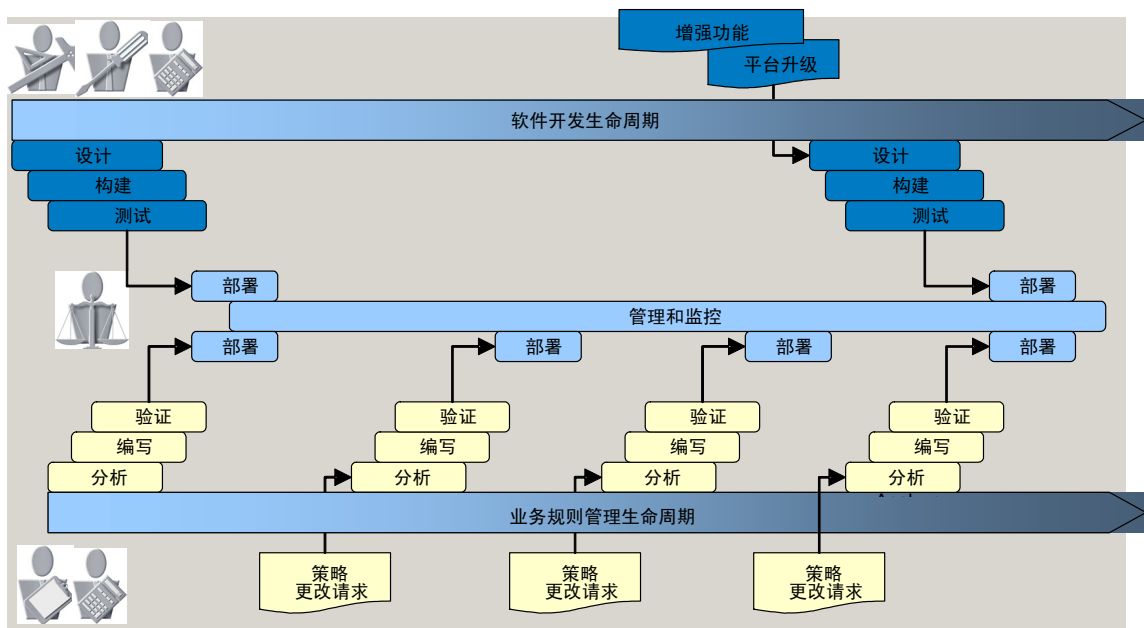


图 1：规则和软件开发生命周期

2.2 相关人员

许多不同的项目相关人员共同推动业务规则应用的软件开发和业务规则生命周期。图 1 列出了业务规则管理项目中的主要项目相关人员，包括：

- **系统设计师：** 系统设计师负责应用的整体设计，确保设计在功能、效率和性能上满足长期业务需要。
- **业务分析人员：** 业务分析人员负责分析和理解业务需求，并将这些信息转化为数据和流程说明。
- **开发人员：** 开发人员创建和测试应用。
- **策略管理者：** 策略管理者是负责将业务策略转化为详细业务规则的专家。
- **系统管理员：** 系统管理员负责在生产过程中管理应用，以达到所要求的正常运行时间和性能目标。

3 ILOG JRules 5.0 业务规则管理工具

ILOG JRules 5.0 为项目中的每个相关人员提供了与其角色相对应的工具，以满足应用的软件开发和业务规则的需要。

3.1 系统设计师

应用设计师对业务规则应用的开发起着至关重要的作用。他们的主要目标是为应用及其关联的数据流创建行之有效且高效率的整体结构。ILOG JRules 5.0 为系统设计师提供了一整套可应用于各种不同应用架构的灵活架构元素。以下各部分从系统设计师的角度介绍这些组件。图 2 显示了各元素之间的关系。

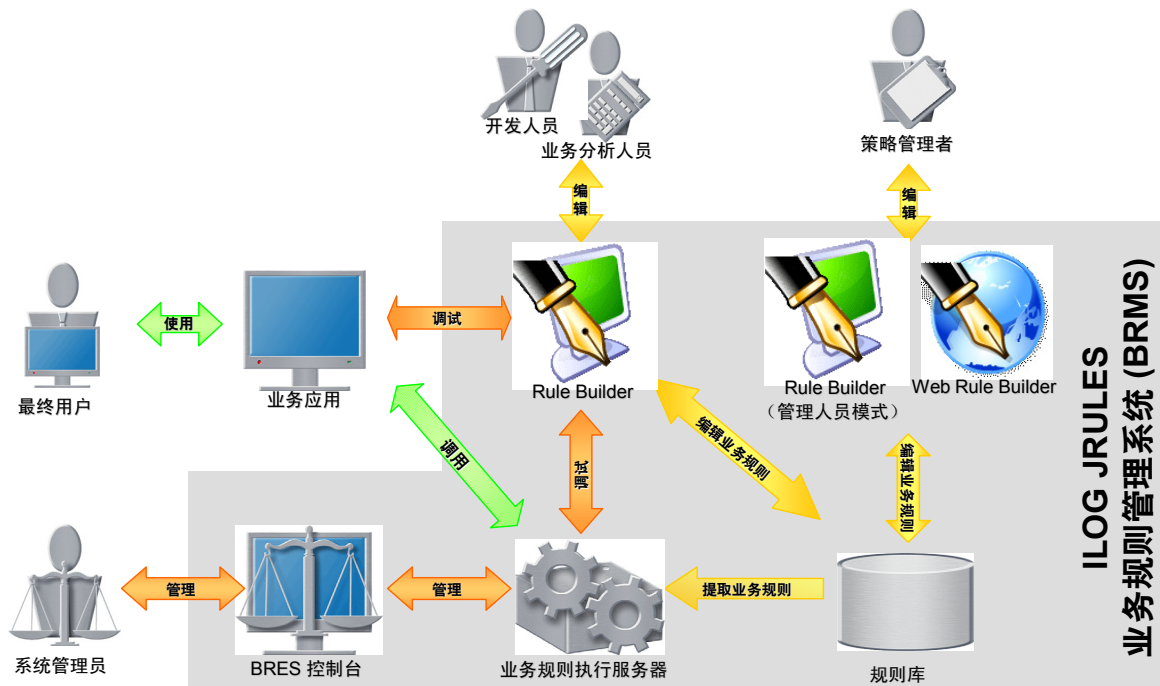


图 2: ILOG JRules 架构元素

3.1.1 规则库

在业务规则的整个生命周期中，应用必须支持对其业务规则进行维护和管理。为此，ILOG JRules 提供了规则库。规则库是为正在开发、测试或生产的业务规则提供的功能强大的中央存储区域。它具有以下特性：

- **树形组织结构：**规则库使用程序包树（类似于 Java 程序包树）的形式组织业务规则。
- **元数据扩展：**规则库可跟踪作者、创建日期和修改时间等标准元数据。同时还为规则状态和其它常用的元数据字段提供标准扩展。规则库还可以进行扩展，从而包括应用或企业特有的元数据。扩展的元数据可以是任何能够用 Java 对象表示的类型。
- **权限管理：**业务规则应用常常需求非常细致的权限管理，根据用户角色来确定用户可更改哪些规则属性（前提是存在规则属性）。ILOG JRules 为规则库提供了通用的权限管理器界面，使系统设计师能够依靠 LDAP 或数据库目录，或是所选择的任何角色和授权服务器，实现基于角色的权限管理。
- **版本管理：**在实际的业务规则生命周期中，业务规则可能会经历一系列复杂的变化，这些变化要求业务规则在不同阶段（如开发、质量保证和生产）具有不同的版本。ILOG JRules 规则库支持规则的多个受控版本。
- **历史记录：**ILOG JRules 规则库可维护有关所有规则变化的完整历史记录。用户可按规则逐条查看和浏览历史记录。
- **并发控制：**ILOG JRules 规则库采用内容锁定机制，可支持多个用户同时进行规则编写和管理。
- **语言定义和扩展：**ILOG JRules 提供了用户友好的业务操作语言 (BAL)，无需定制即可用于大多数项目中。如果项目不适合采用业务操作语言，系统设计师可以使用业务规则语言定义工具 (BRLDF) 来创建自定义的规则语言。

3.1.2 ILOG JRules Builder 和 Web Builder

基于 ILOG JRules 的应用为策略管理者提供了用于编写和维护规则的用户界面。如图 2 中所示，ILOG JRules 5.0 提供了两个用于实现此功能的组件：

- **Builder:** ILOG JRules Builder 是一种具有丰富图形用户界面的 Java 应用，策略管理者可用它对规则库进行全面维护，包括：
 - 规则编写
 - 元数据管理
 - 规则库查询

此外，开发人员还可以使用 Builder 执行更高级的操作，包括：

- 测试和调试规则
- 提取和部署规则集
- 分析规则集

用户可对 Builder 中的功能进行扩展（添加新菜单和操作，指定自定义的编辑工具）或限制（删除菜单和操作），以创建适合各级别开发人员或策略管理者的规则管理界面。

- **Web Builder:** Web Builder 是一个 J2EE (Servlet) 应用，它以瘦客户端形式为策略管理者提供基本的规则管理功能。Web Builder 支持基本的策略管理者操作：规则编写、元数据管理和查询。

3.1.3 规则引擎

业务规则应用通过调用规则引擎来调用规则。ILOG JRules 5.0 规则引擎是一个灵活、高性能的执行平台，适合单独用于 J2SE 或 J2EE 应用。该引擎的主要特性包括：

- **RETE 和顺序模式执行:** ILOG JRules 引擎支持传统的 RETE 规则（正向链接或推理）和高性能的顺序模式执行。在顺序模式下，可使用实时 (Just-In-Time, JIT) 技术将业务规则编译成 Java 字节代码，然后按预先确定的顺序直接执行这些代码。
- **调试接口:** ILOG JRules 引擎支持对正在运行的规则集进行本地和远程的规则调试。
- **分析接口:** 规则引擎提供了一个工具接口，使您能够对正在运行的规则集进行远程分析。
- **J2SE 和 J2EE 部署:** 系统设计师可利用规则引擎中丰富、灵活的应用编程接口 (API)，在 J2SE 应用中可将引擎部署为可直接调用的服务；或者在 J2EE 容器中将引擎部署为远程服务或本地服务。3.1.4 一节将更详细地介绍 J2EE 部署。

3.1.4 ILOG JRules 业务规则执行服务器

ILOG JRules 5.0 中的业务规则执行服务器（BRE 服务器）使 J2EE 业务规则应用的设计师受益匪浅。BRE 服务器是一个预先打包的 J2EE 应用，它通过标准的 J2EE 连接器架构 (J2C) 和 EJB 接口提供了 ILOG JRules 引擎。

对于要转向面向服务的架构 (SOA) 的企业而言，BRE 服务器是一个理想的构建块。通过使用 BRE 服务器以及 J2EE 平台的 SOA 特性，设计师可以将任何基于规则的决策过程部署为 Web 服务，或通过 EJB 会话或消息 Bean 接口提供这类决策过程。正如 ILOG JRules 能够直接根据 XML Schema 定义的 XML 文档进行推理一样，ILOG JRules 可以从任何特定的 Java 表示形式的业务模型中提取业务逻辑，这种能力非常适合 SOA。

同样，旨在建立企业级业务流程管理 (BPM) 平台的企业可借助 SOA 技术，利用 ILOG JRules 作为决策引擎，将基于业务规则的服务集成到 BPM 决策节点中。

BRE 服务器在架构上（如图 3 所示）由三个独立的模块组成：执行单元 (XU)、模型管理和监控组件以及持久性组件。

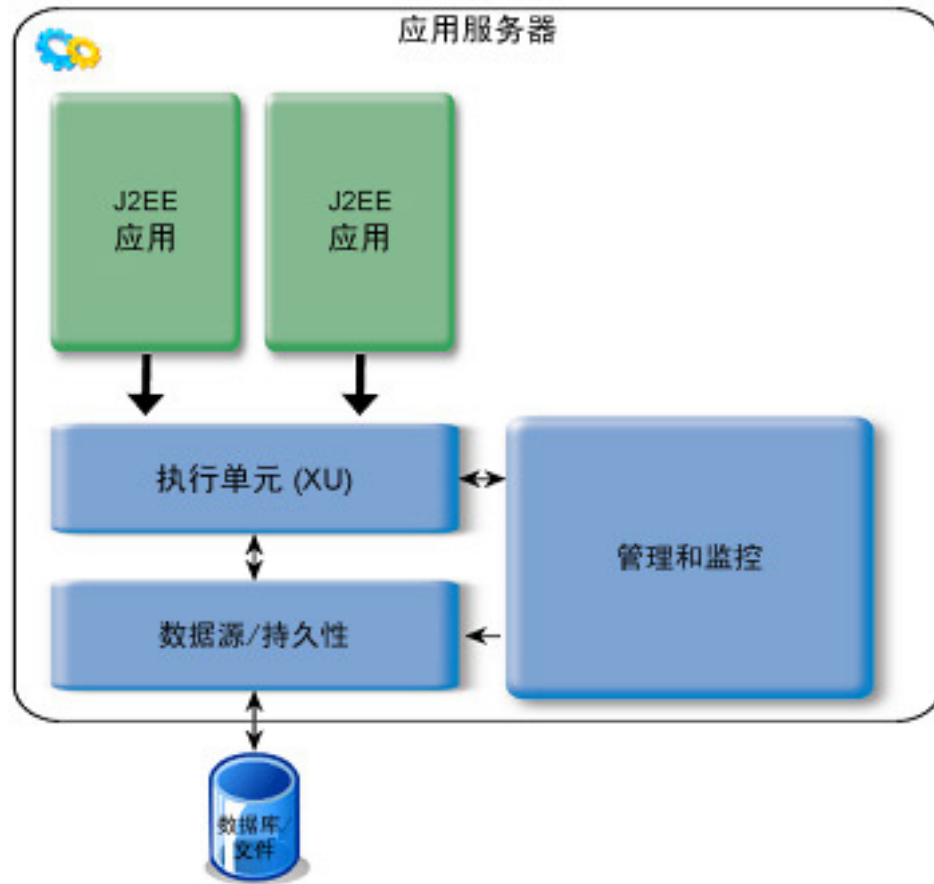


图 3: BRE 服务器架构

XU 可以支持多个业务规则应用，并确保轻松地实现 XU 升级，同时提供 XU 的整体管理和监控的可见性。

在单个业务规则应用中，应用代码调用 BRE 服务器的执行组件，而这些执行组件又调用 XU 来执行业务规则。为此，ILOG JRules 提供了无状态和有状态的会话 EJB 以及标准的 JavaBean 执行组件。应用开发人员利用执行组件的简单应用编程接口 (API)，可轻松地在应用中嵌入对 BRE 服务器的调用。

例如：

```
// acquire a rule session to execute the rules
IlrRuleSessionProvider provider = IlrManagedRuleSessionProvider.getProvider();
IlrStatelessRuleSession ruleSession = provider.createStatelessRuleSession();
// get a helper instance with which to communicate with the session
IlrRuleSessionExecutionHelper helper = new IlrRuleSessionHelper();
// attach the objects needed to evaluate the rules and store the results
helper.addParameter("RuleResult", ruleResult);

helper.addParameter("ShoppingCartFacade", scf);
// invoke the rules
```

```
ruleSession.executeRules("/PetStoreRuleapp/PetStoreRules", null,
    helper.getJavaClassResolver(this), helper.getParameters());
```

通常，BRE 服务器执行组件被打包到应用 EAR 文件中，以确保业务规则使用的应用 Java 类可用于 XU。

图 4 阐释了应用组件、BRE 服务器执行组件和 BRE 服务器执行单元之间的关系。

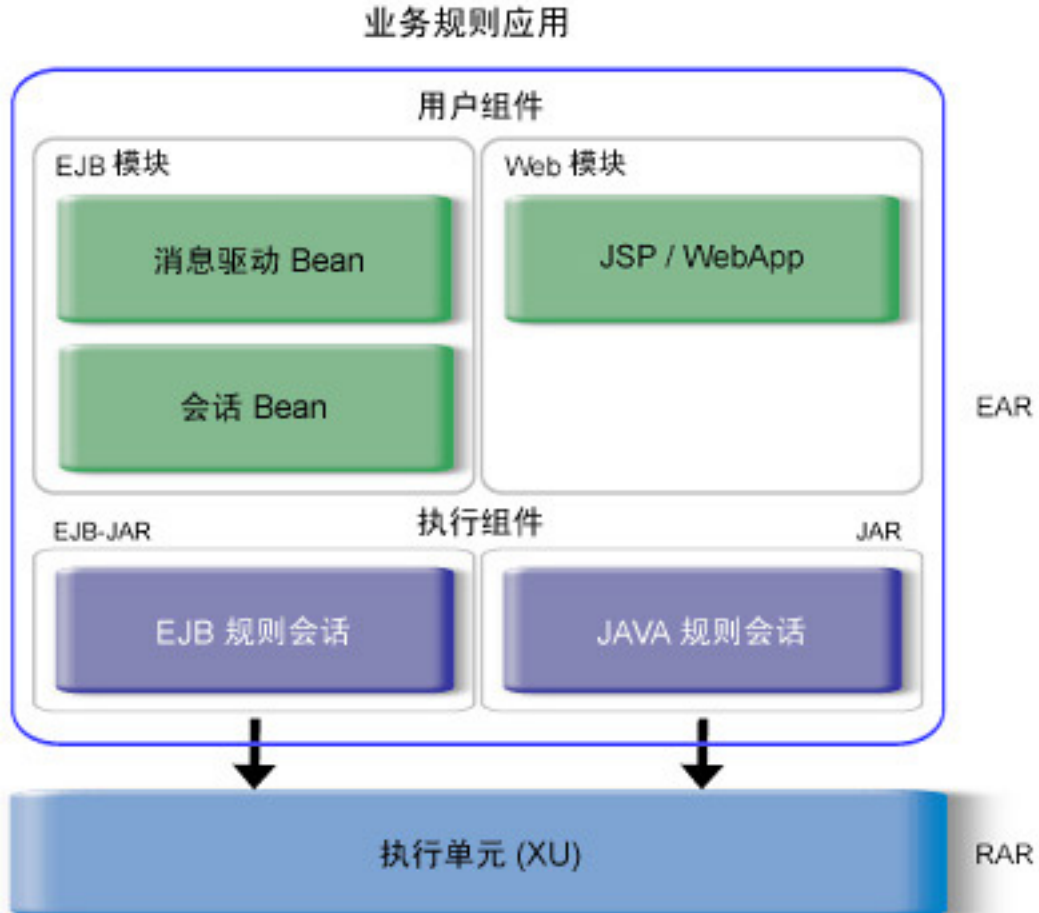


图 4：在业务规则应用中调用 BRE 服务器

除了提供完全受控的执行环境（如 3.6 一节所述），BRE 服务器在架构方面还具有以下主要优势：

- **高效的垂直和水平伸缩性：**
- **高性能的规则集执行：** BRE 服务器充分利用高性能 ILOG JRules 引擎的 RETE 模式和顺序模式。
- **高可用性：** 为最大限度地提高可用性，BRE 服务器支持集群部署，并支持在正在运行的服务器上对规则集进行热部署。
- **详细的技术日志记录**
- **调试：** 可在 ILOG JRules Builder 或基于 Eclipse 的 ILOG Business Rule Studio（开发版）中远程调试正在运行的规则集。

3.2 业务分析人员

在典型的项目中，业务分析人员起着桥梁作用，他们负责对将要自动执行的流程进行完全面向业务的描述，使之与项目的技术规范相衔接。由于传统应用的技术规范中所包含的许多详尽系统功能，现已转由策略管理者直接管理，因此，这种桥梁作用对于现今的业务规则应用尤其重要。分析人员负责创建问题域模型，这种模型对策略管理者来说是智能的、有用的，同时支持开发人员进行高效的实施。

通过 ILOG JRules 5.0，分析人员可以使用以下三个重要工具来实现这种桥梁作用：对象模型规范、业务规则语言规范和规则流。此外，分析人员通常会设计某些高级业务规则工具，供策略管理者以后进行定制。这些工具包括规则模板、决策表和决策树。以下几节将详细介绍。

3.2.1 对象模型规范

ILOG JRules 以业务对象模型 (BOM) 的形式表示业务规则要处理的问题域。与 UML 对象模型类似，BOM 是问题域中所有概念、数据元素和关系的结构化表示形式。

在 ILOG JRules 项目中，对象模型用 Java 类来表示或用表示模型可执行版本的 XML Schema 来表示。对象模型的具体实施通常是开发人员的任务。不过，在零起点的 ILOG JRules 项目中，业务分析人员可能会在定义模型的过程中起到关键作用。分析人员或开发人员可以使用能生成 Java 类或 XML Schema 的任何建模工具来创建基本的对象模型，然后使用 ILOG JRules Builder 将业务概念的语言描述映射到对象模型，以完成类的创建。

图 5 显示了具有汽车保险应用的简单类模型的 Builder。对象模型以树形结构显示在左侧窗格中，并以图形表示形式显示在中间窗格中。

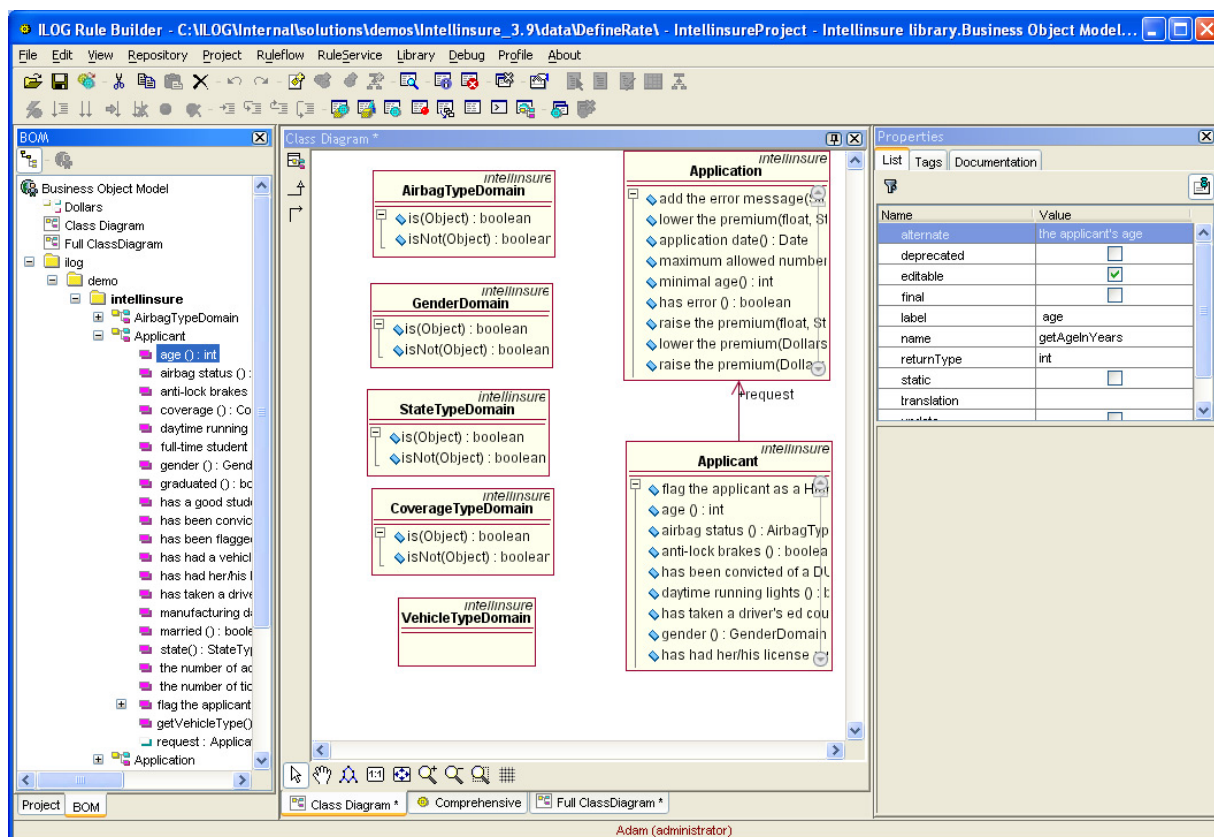


图 5: 显示简单对象模型的 Builder

有关创建对象模型和为对象模式添加注释的更多信息，请参见 3.4.1.1 一节。

3.2.2 业务规则语言规范

ILOG JRules 5.0 将业务对象模型映射为策略管理者熟悉的词语。一旦定义了业务对象模型，分析人员便可使用 ILOG JRules Builder 以适当的短语对模型进行注释，添加策略管理者熟悉的语言。

大多数项目都可以使用 ILOG JRules 业务操作语言 (BAL) 来编写规则。BAL 将策略管理者用来编写规则的一组动词和名词映射到业务对象模型中的类、方法和字段。分析人员使用 Builder 来进行这种映射。此外，分析人员可以和开发人员协作，使用由 BAL 转换为 Java 构件的虚拟方法和字段，扩展对象模型的 Java 对象中的“hardwired”（硬连接）定义。例如，图 6 中的右窗格为树视图中选择的对象 (Applicant) 和方法 (age) 显示一个属性资源管理器。在该属性资源管理器中，分配给 age 属性的短语是“the applicant's age”。中间的窗格显示使用该短语的规则：“the applicant's age is greater than 70”。

在图 6 最右侧的窗格中，Builder 将短语“year in school”映射到 Student 类的 getYear() 方法。中间的窗格显示使用这个 BAL 短语表示的规则。

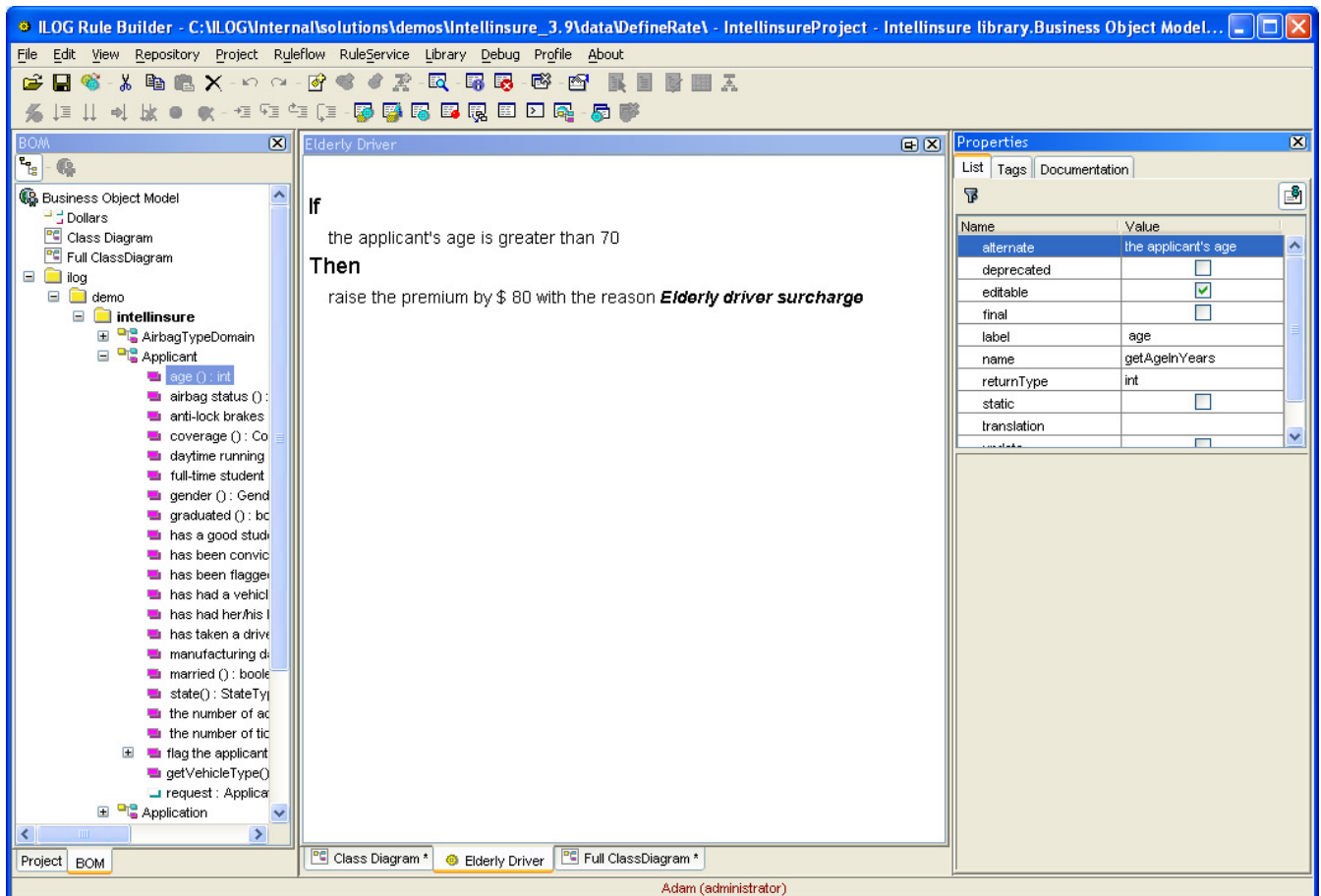


图 6：将业务名词映射到对象模型构件

3.3 规则流

业务规则是业务策略的具体体现。而对于高级规则组织形式而言，通常需要为规则集指定执行顺序和条件。ILOG JRules 5.0 允许分析人员指定用于调用规则集的分组、顺序和条件，从而支持这些规则流。分析人员在 ILOG JRules Builder 中以图形方式创建规则流。图 7 提供了一个示例，其中，规则处理从“Data Validation”（数据验证）开始，然后有条件地流向“Eligibility”（合格）和“Pricing”（定价）。

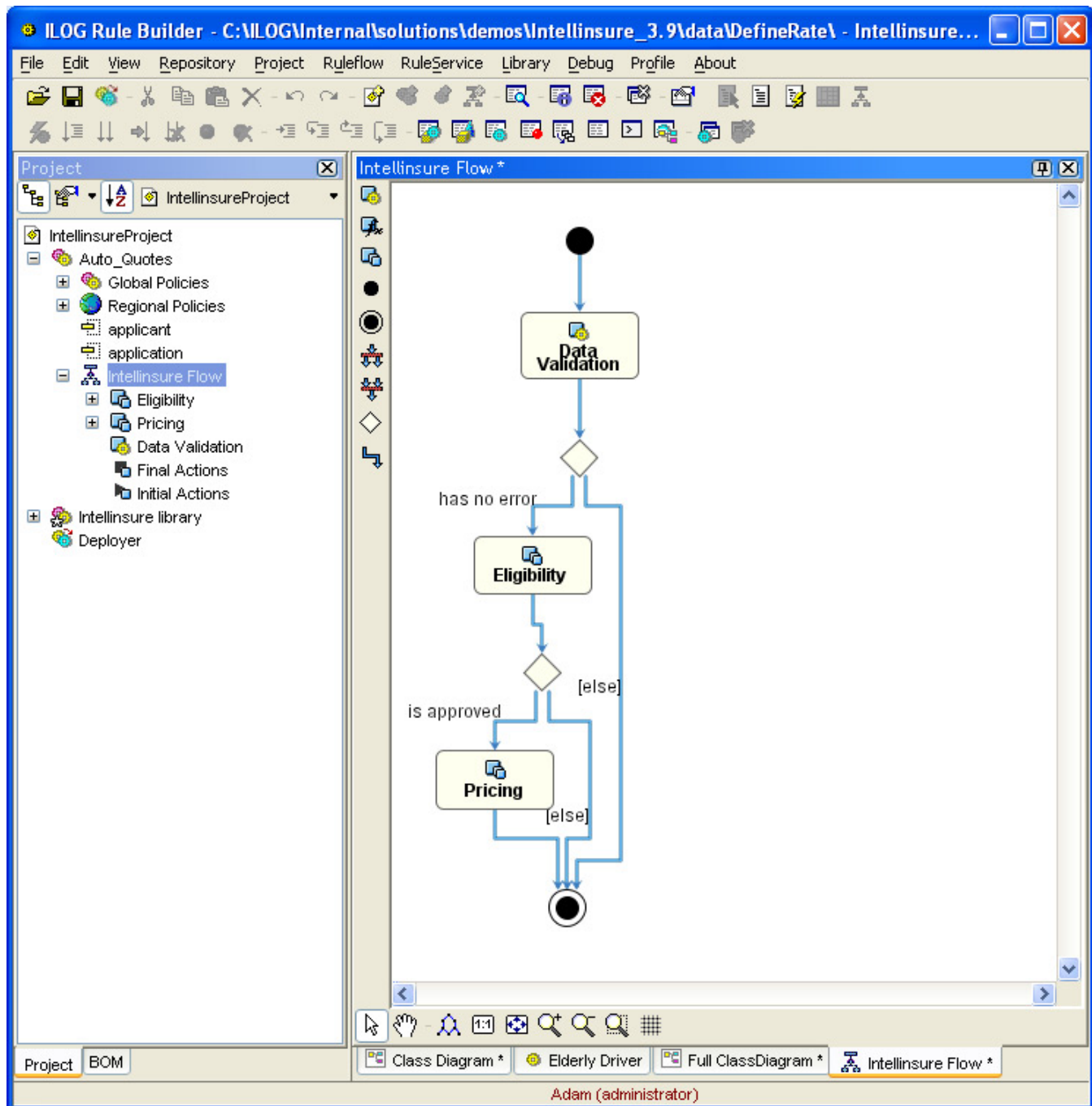


图 7: ILOG JRules Builder 中的规则流

3.3.1 规则模板

在许多应用中，策略管理者编写的所有（或大部分）规则都是在少数通用模式的基础上稍加改变。业务分析人员在了解这些通用模式后，就可以将它们应用于 ILOG JRules 规则模板。策略管理者在以后编写和测试要实际部署的业务规则时，可以对模板进行定制。这种方法简化和加快了生成规则的过程，同时降低了出现人为错误的可能性。

图 8 显示了 ILOG JRules Builder 中的一个模板。该示例模板是为应用折扣策略而部分构建的业务规则。显示为蓝色并带有尖括号的项（如 `<select a condition>`）是业务规则中由使用模板的策略管理者提供的内容。其余内容则是该模板的固有部分。

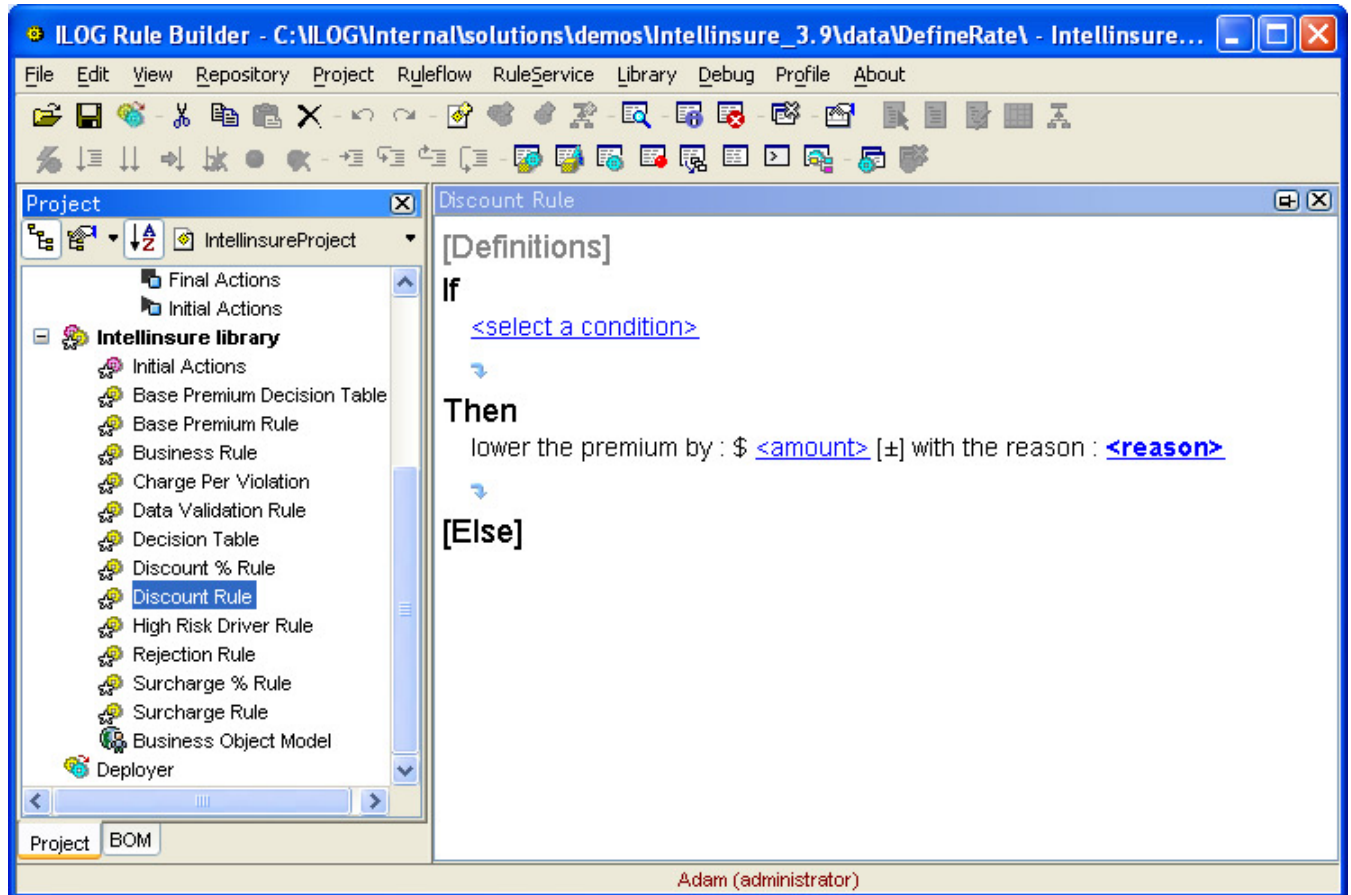


图 8: Rule Builder 中的规则模板

3.3.2 决策表和决策树

有些业务规则最适合采用离散逻辑元素形式编写（图 6）。然而，许多决策过程是高度结构化的，以表格或图形表示可能更有效。ILOG JRules 5.0 为此专门提供了两个规则构件：决策表和决策树。

决策表适于以下情况：多个条件在许多业务规则中有规律地重复出现，每次都产生类似的结果。每个决策表均基于特定“数据格式”（数据格式通常由业务分析人员为特定决策过程创建，并由策略管理者填充）。图 9 显示了 ILOG JRules Builder 决策表编辑器。在此示例中，Builder 为一个表构建数据格式，该表使用 gender、age、marital status 和 state 属性来选择基本保险费金额。图 16 显示了策略管理者所看到的同一个决策表。

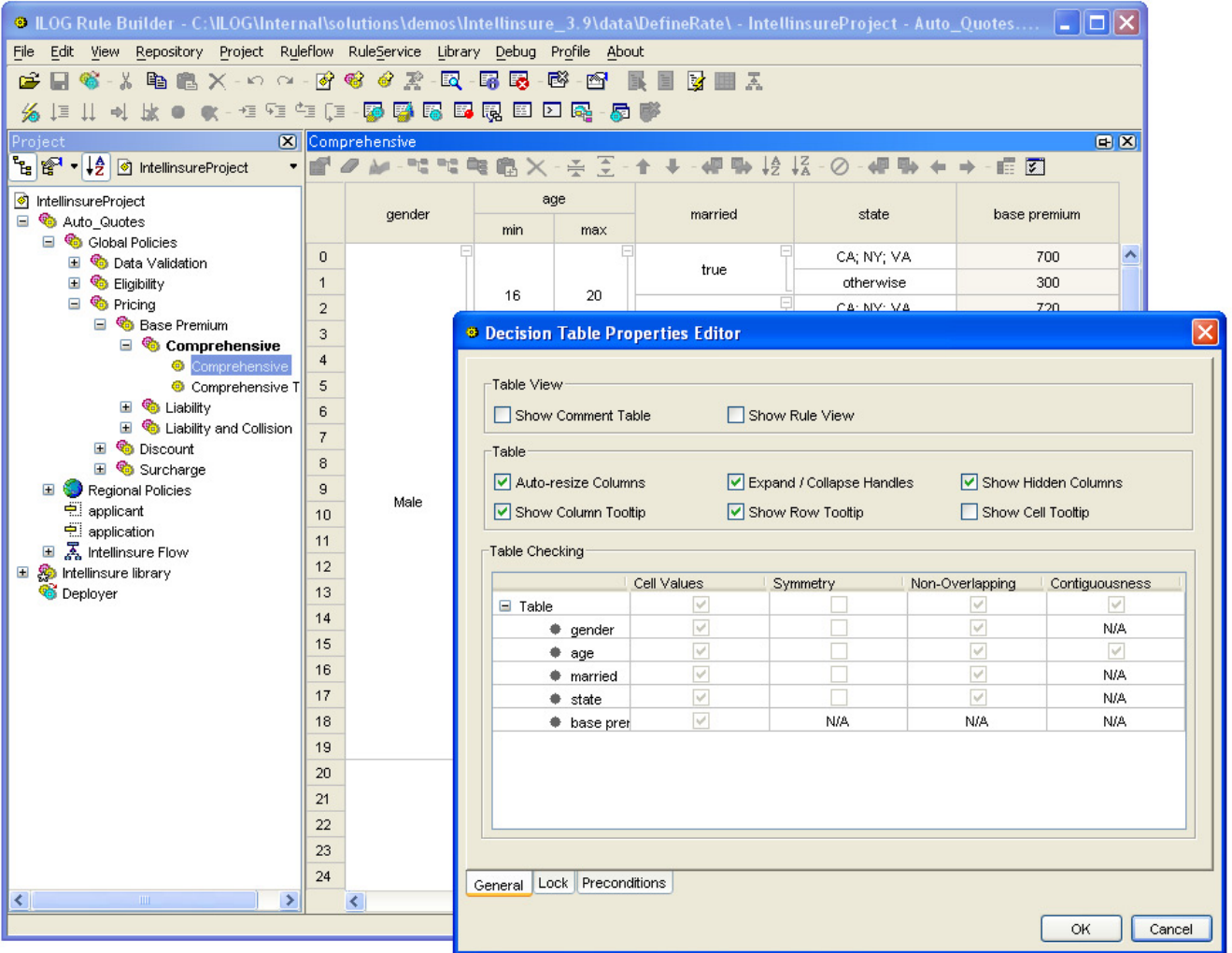


图9：决策表编辑器

决策树与决策表类似，但它适于以下情况：对于某些开始条件，将完全跳过决策表中的特定列。与决策表一样，业务分析人员通常创建决策树的基本流程和布局。可以使用图形化的“拖放”编辑器构建决策树，如图10所示。

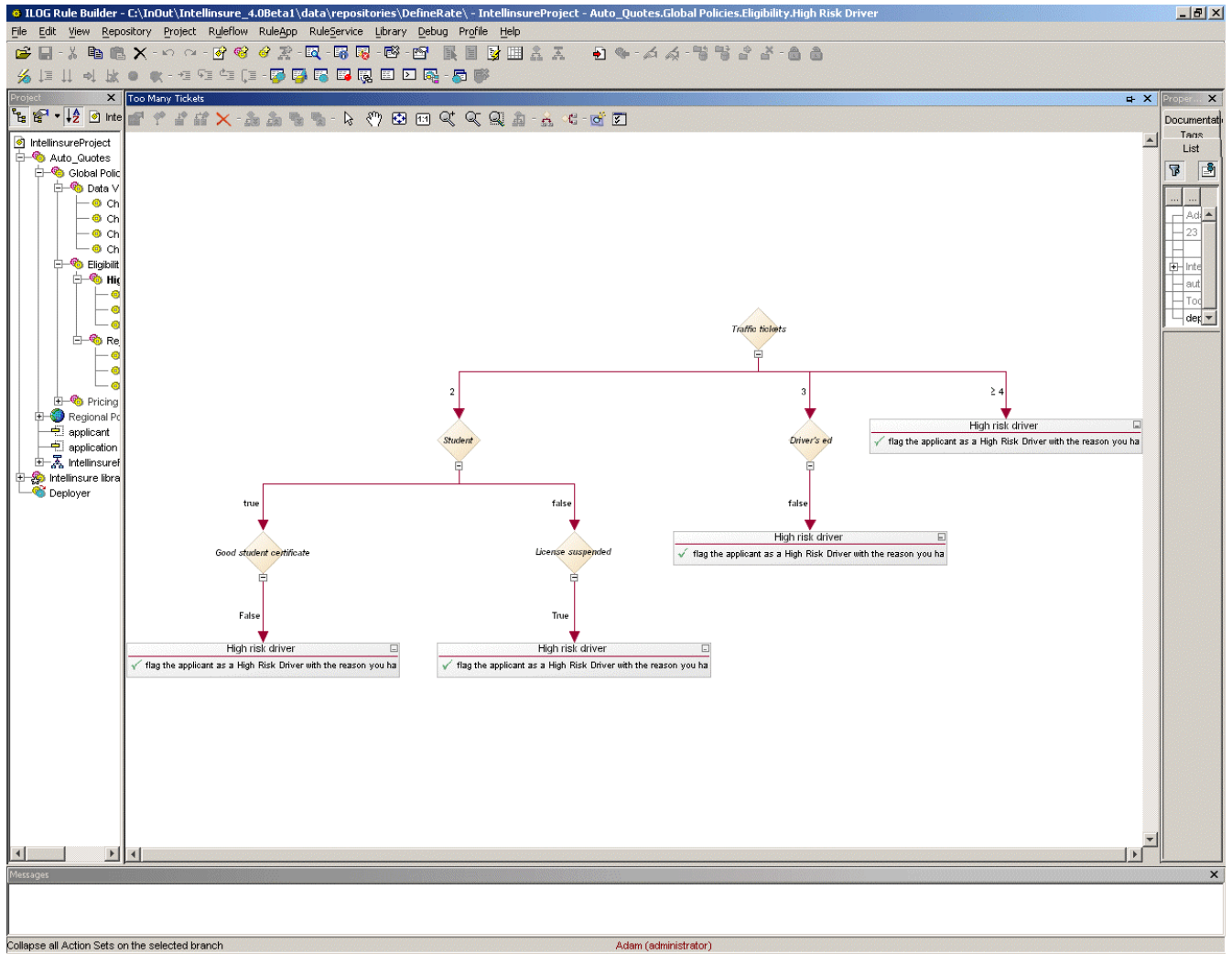


图 10: 决策树编辑器

3.4 业务规则应用开发人员

通常，对于实施应用的固定元素的平台，开发人员将负责编写相应的代码。不过，对于业务规则应用，开发人员的工作还包括一些新的任务：

- 与分析人员协作构建业务对象模型。
- 在交付给策略管理者之前，完成业务规则的编写、测试和调试（尤其在应用开发的早期阶段）。
- 整合利用业务规则实施的可变业务逻辑。这通常包括：
 - 基于上下文收集适用的规则
 - 收集规则必须处理的数据
 - 调用本地或远程规则引擎
 - 整合数据处理结果与业务规则，以处理固定逻辑
- 构建（或扩展和定制）业务规则编写工具

为了有效地执行这些任务，开发人员要求工具不仅能够与现有的开发方法一起使用，还可以用于测试和调试业务规则。为此，ILOG JRules 5.0 提供了基于 ILOG JRules Builder 的一整套开发工具以及全面的应用编程接口 (API)，用于集成业务规则和应用，并构建基于 ILOG JRules Builder 或 ILOG Web Builder 的定制业务规则编写工具。

3.4.1 ILOG JRules Builder

ILOG JRules Builder 是一个具有丰富图形用户界面的 Java 应用，开发人员可用来进行全面的业务规则应用开发，包括：

- 创建业务对象模型
- 管理规则库
- 编写业务规则
- 提取和部署规则集
- 测试和调试业务规则
- 执行分析

3.4.1.1 创建业务对象模型

在许多项目中，分析人员负责提出业务对象模型 (BOM) 的最初设想。不过，业务对象模型最终必须与 Java 或 XML 执行对象模型 (XOM) 相联系。鉴于对象模型的上述要求及其它高级特性和语言规范，开发人员需要具有 Java 开发经验。

开发人员使用 Builder 创建业务对象模型，并使其与 Java XOM 或 XML Schema 相关。在大多数情况下，这是通过导入现有的 Java 类或 XML Schema 实现的。导入后，使用特定于域的短语（构成策略管理者用于编写业务规则的语言）来“修饰”这些元素。

修饰过程具有非常高的灵活性。它允许开发人员为类、属性和方法指定类似于英语的名称，还可以详细指定如何将这些名称转换为对实际 XOM 方法的调用。开发人员还可以通过向业务对象模型添加“虚拟方法”来建立新概念，这些方法能够转换为对 XOM 进行的复杂复合调用。

对于非常复杂的对象模型，开发人员可以为业务对象模型添加注释，以指定以下内容：

- 策略管理者可以采取何种方式以及在什么情况下“浏览”用于编写规则的对象模型。
- 对象在给定的规则调用中出现一次还是出现多次。
- 用于修改对象的方法是否要求对规则执行序表进行重新评估。

图 11 显示了正用于编辑业务对象模型的 Builder。在该示例中，开发人员指定，虚拟方法“is”应被转换为对 Java 对象方法“equals()”的调用。

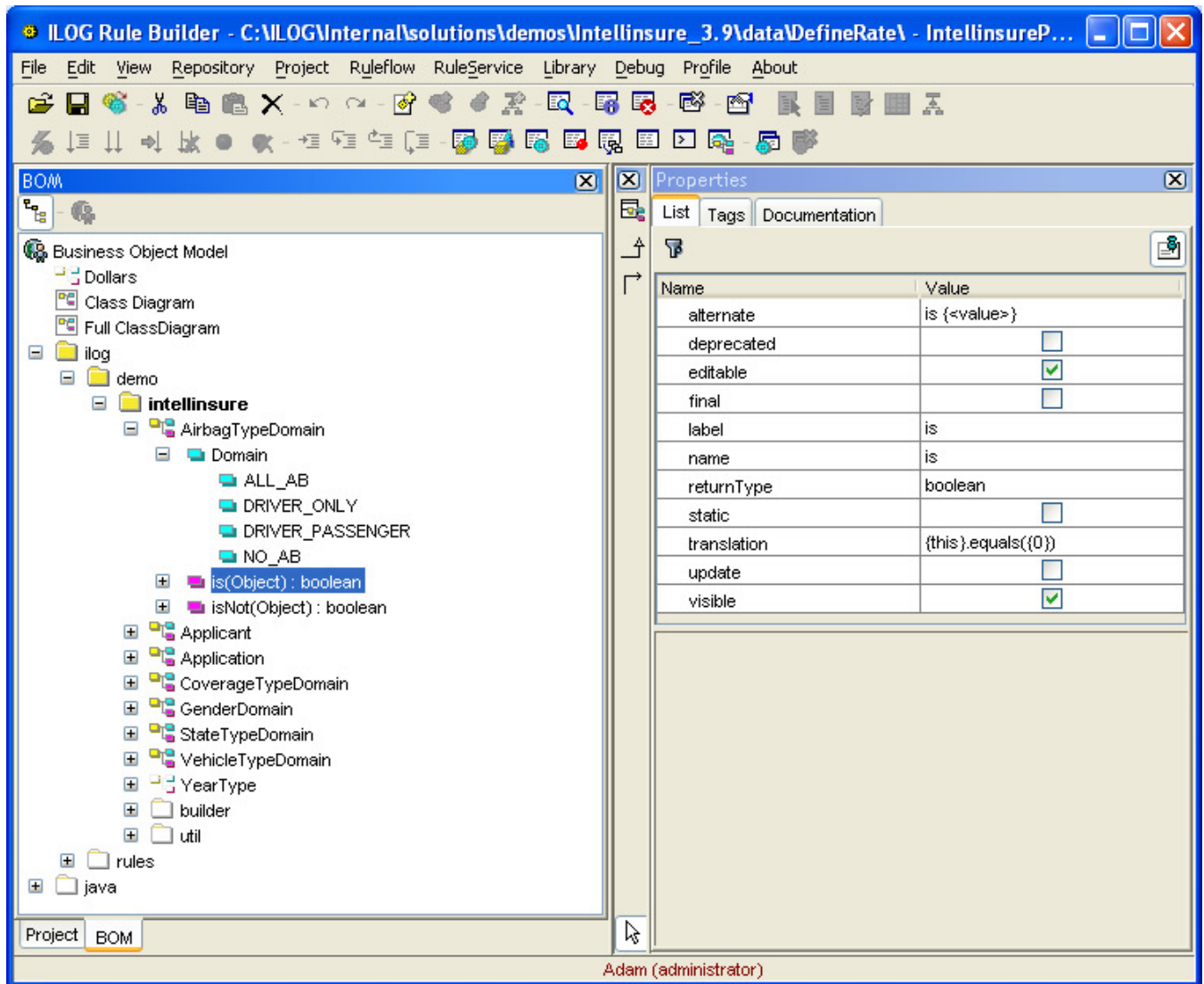


图 11: 在 Builder 中基于 Java 类编辑业务对象模型

3.4.1.2 规则库管理、规则编写和源代码控制集成

为创建业务规则应用，开发人员必须能够编写和管理业务规则，以便将业务规则集成到开发任务中。这意味着，开发人员应始终将一部分注意力放在规则库上。

开发人员通常先处理项目组件的副本，然后使用源代码控件或配置管理系统将所做的更改合并到公共项目代码库中。ILOG JRules 5.0 允许用一种文件系统（类似于存储 Java 软件包和类的文件系统）来表示规则库的内容，从而简化了上述过程。开发人员可以标记或锁定要修改的规则包，随后使用标准的源代码控制机制，将它们合并回项目代码库中。ILOG JRules 为 CVS 源代码控制系统提供了一个插件，可直接将该系统的功能集成到 ILOG JRules Builder 中。还可以通过开放的应用编程接口 (API)，将此集成扩展到其它源代码控制系统。图 12 显示了在 Builder 中调用的 CVS 提交操作。

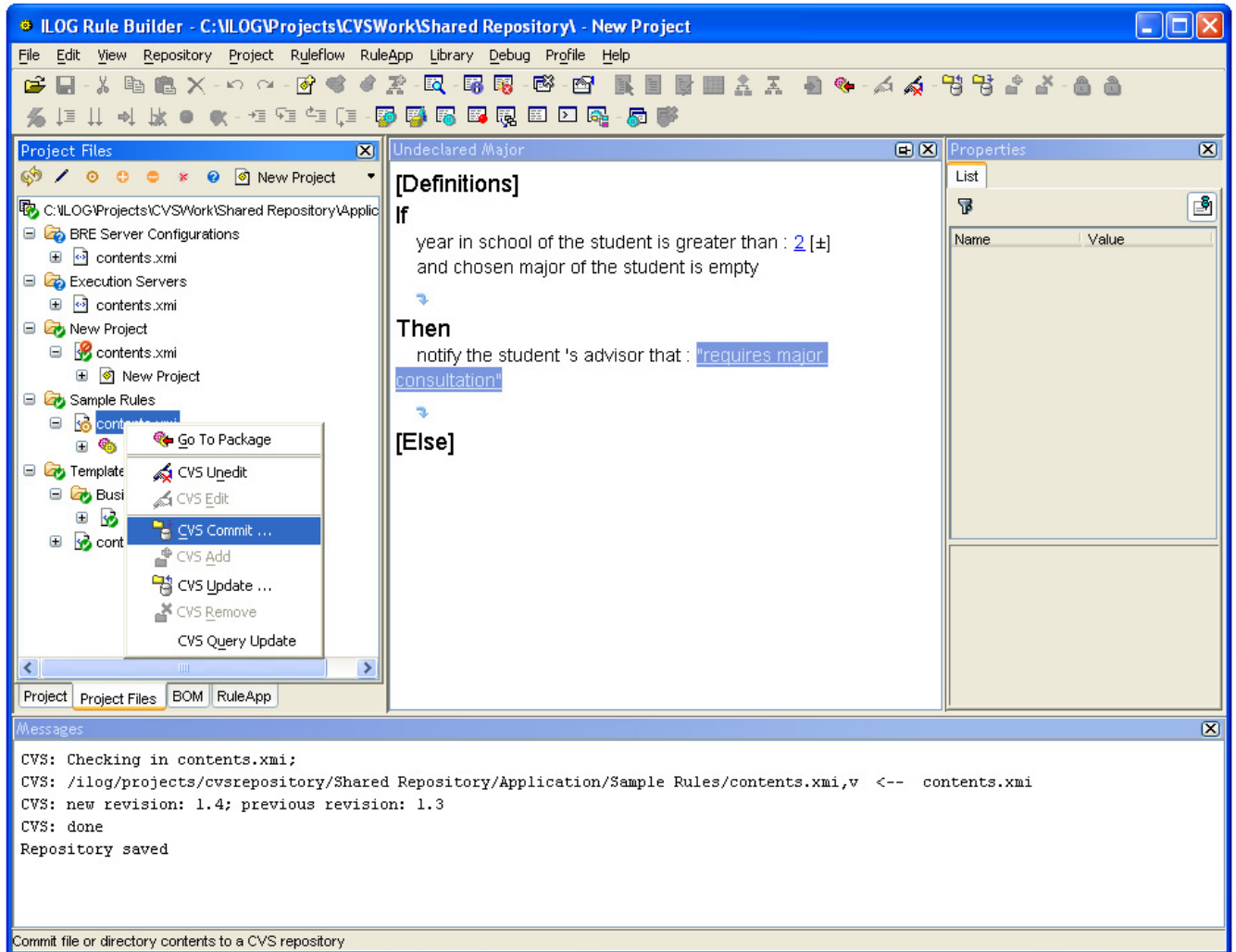


图 12: 向 CVS 规则库提交规则更改

3.4.1.3 规则集提取和部署

开发人员可以使用 ILOG JRules Builder 提取规则集，或从业务规则库中提取规则来构建规则集，并创建自定义的提取程序，通过编程从规则库中提取业务规则。

为了进行部署，Builder 与 BRE 服务器紧密集成。Builder 可用于：

- 从规则库提取“RuleApp”（请参见 3.1.4 一节）
- 在调试或发布模式下，将 RuleApp 远程部署到正在运行的 BRE 服务器
- 在规则库中创建所需数量的 BRE 服务器定义，并将各个定义与项目开发、分段实施、测试和生产等阶段联系起来
- 将多个规则集打包到 RuleApp 中，然后导出，以便在脱机状态下手动部署
- 对 BRE 服务器上运行的业务规则应用进行远程调试

3.4.1.4 规则调试

业务规则应用的许多行为是由其业务规则（而非源代码）决定的。这意味着，开发人员在测试和调试时需要了解业务规则的实际工作情况，就像他们习惯于使用 Java 代码一样。开发人员可以使用 ILOG JRules Builder 来调试业务规则、决策表、决策树和业务对象模型。开发人员可利用 Builder 的调试特性来设置断点、单步执行业务规则，并检查要触发的规则执行序表以及业务规则的作用对象。可以在 Builder 中对执行的规则进行本地调试，也可以通过将 Builder 附加到远程 Java 虚拟机中执行的规则引擎（如 3.1.4 一节中介绍的 BRE 服务器）来进行调试。

图 13 显示了某个规则调试会话在业务规则的某个断点处停止。受业务规则影响的对象显示在底部窗格中。

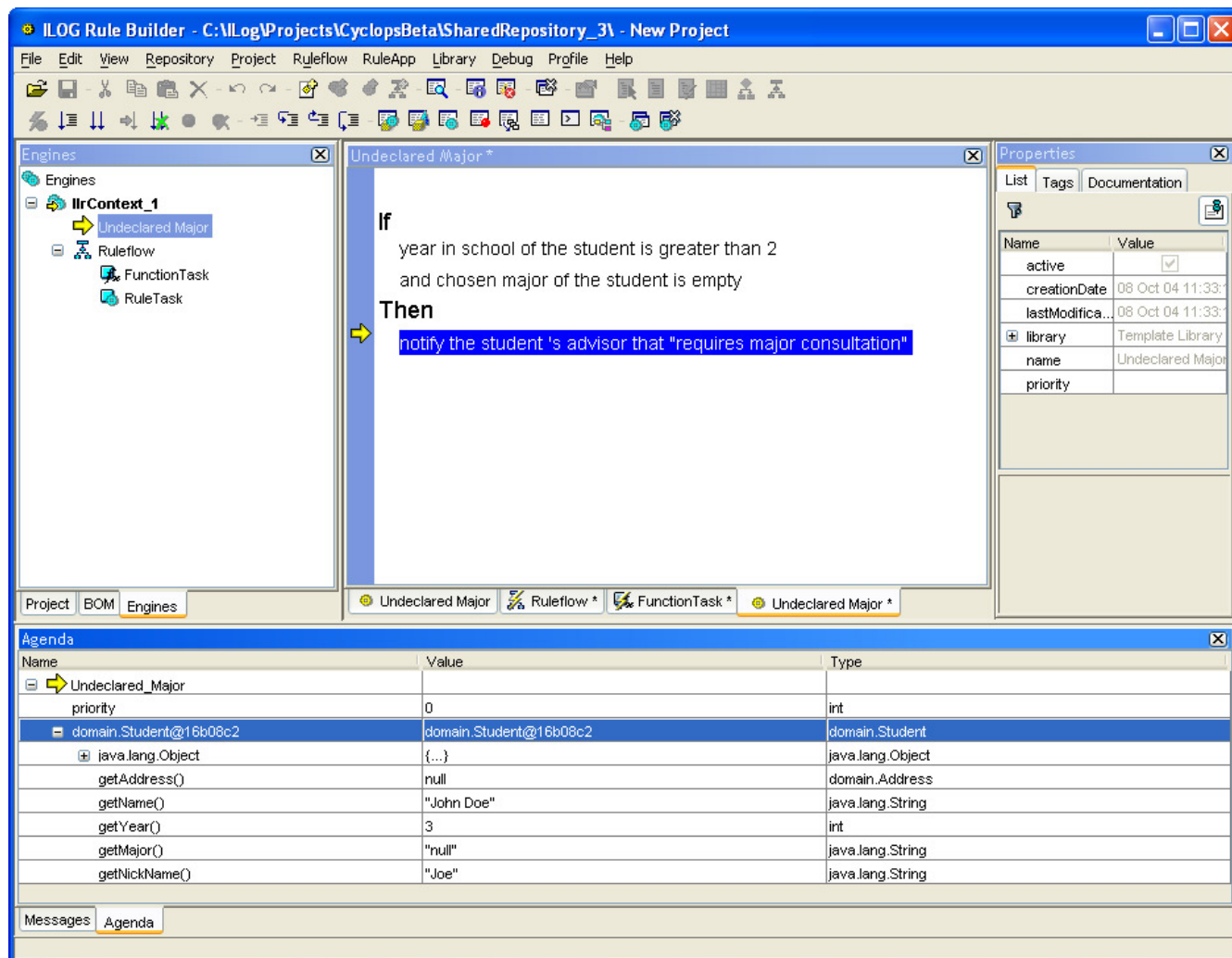


图 13: 规则调试

如果需要对 Java 代码和业务规则进行细微调试，开发人员可以使用 ILOG Business Rule Studio（开发人员版）。ILOG BR Studio 可作为一组插件提供，用于扩展 Eclipse 和 IBM WebSphere Studio 开发环境。使用 ILOG BR Studio，开发人员可以像对待 Java 代码一样处理业务规则，并可以在自己常用的开发环境中执行和协同调试业务规则和应用代码。通过这种“集成的 Java 和业务规则调试”，无需首先了解错误是出在应用的 Java 部分还是出在业务规则中，因此大大简化了错误检测。ILOG 白皮书《ILOG Business Rule Studio（开发人员版）》中详细介绍了 ILOG BR Studio。您可以从 ILOG 的网站 (www.ilog.com) 下载此白皮书及 ILOG BR Studio。

3.4.2 ILOG JRules API

ILOG JRules 通过一个非常灵活的大型应用编程接口 (API) 提供它的几乎全部功能。开发人员可以直接使用此应用编程接口来完成以下任务：

- 定制规则库，包括：
 - 添加自定义的元数据
 - 建立细致的权限管理
- 定制 Builder，包括：
 - 添加自定义的菜单项和操作
 - 添加基于 LDAP 或其它身份验证服务器的身份验证和授权功能
- 从规则库提取和部署规则集
- 在标准应用代码中调用 ILOG JRules 业务规则引擎

3.5 策略管理者

通过业务规则应用，实施业务策略的责任从开发人员转移到策略管理者。策略管理者是业务方面的专家，负责制定和验证执行业务策略所依据的具体业务规则。策略管理者包括产品经理、客户服务经理和其他非技术领域的专业人员。例如，财产及意外事故保险应用的策略管理者可以是负责制定承保决策原则的信用保险单经理。信用保险单经理可以利用业务规则应用，直接编写实施这些原则的业务规则。在这种情况下，策略管理者需要能够编写、测试和管理业务规则的工具，即支持第 2.1 节介绍的业务规则生命周期的工具。

3.5.1 规则编写

策略管理者要想有效地编写业务规则，需要使用易于理解的业务规则表达语言。如 3.2.2 一节中所述，ILOG JRules 提供了一种创建业务规则语言的简单方法，使策略管理者能够使用熟悉的术语和关系编写互不关联的业务规则。对于要求进行瘦客户端部署的应用，可以使用 ILOG JRules Builder 或 ILOG JRules Web Builder 来编辑业务规则。在这两种情况下，都可以充分利用规则模板和“点击式”编辑器来编辑业务规则，这就简化了业务规则语言术语的使用。图 14 显示了一个在 Builder 中编辑规则的示例，可利用其中的下拉提示菜单插入业务规则语言表达。图 15 显示了 Web Builder 中的类似操作。

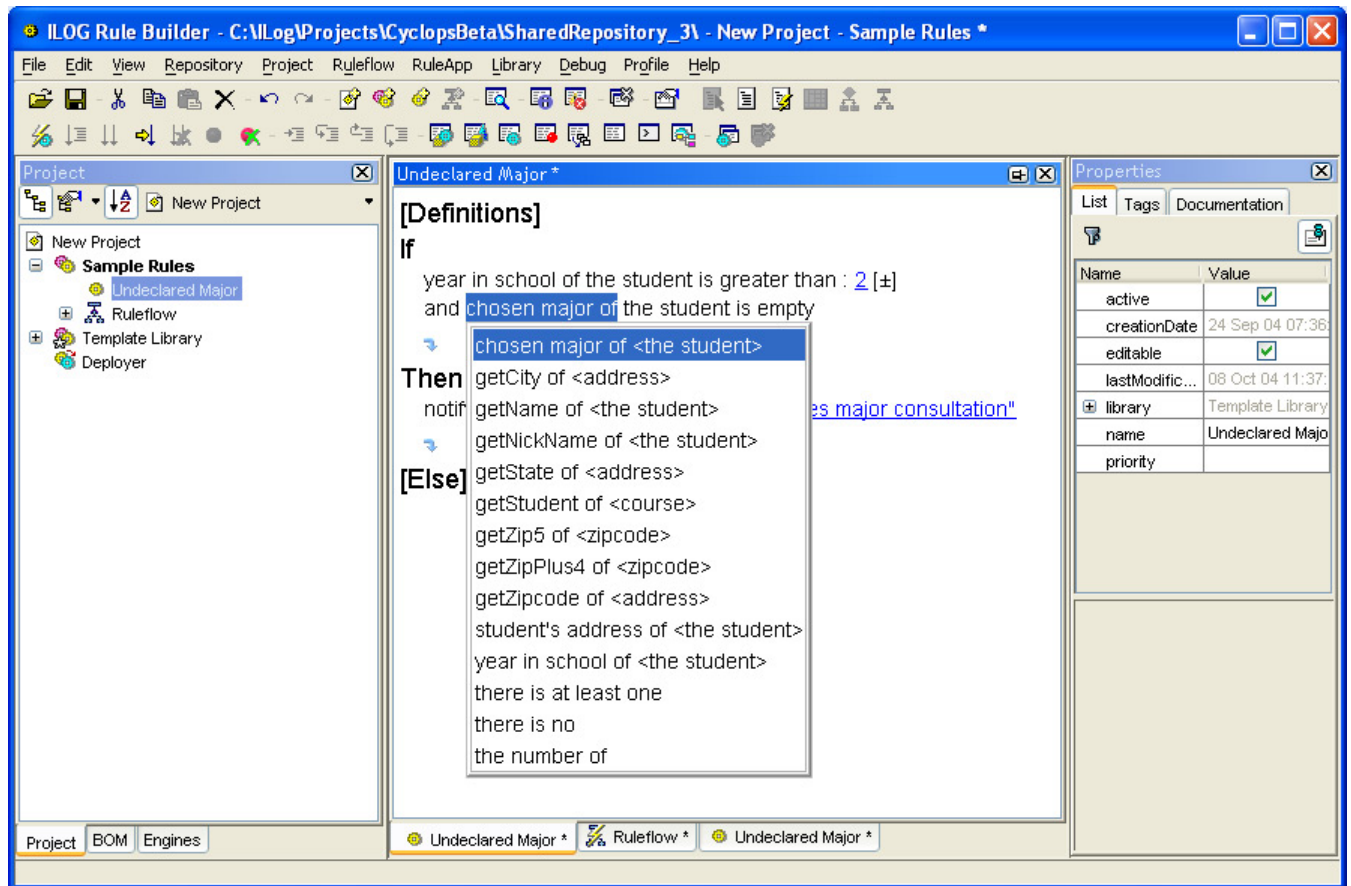


图 14: 在 ILOG JRules Builder 中编辑业务规则

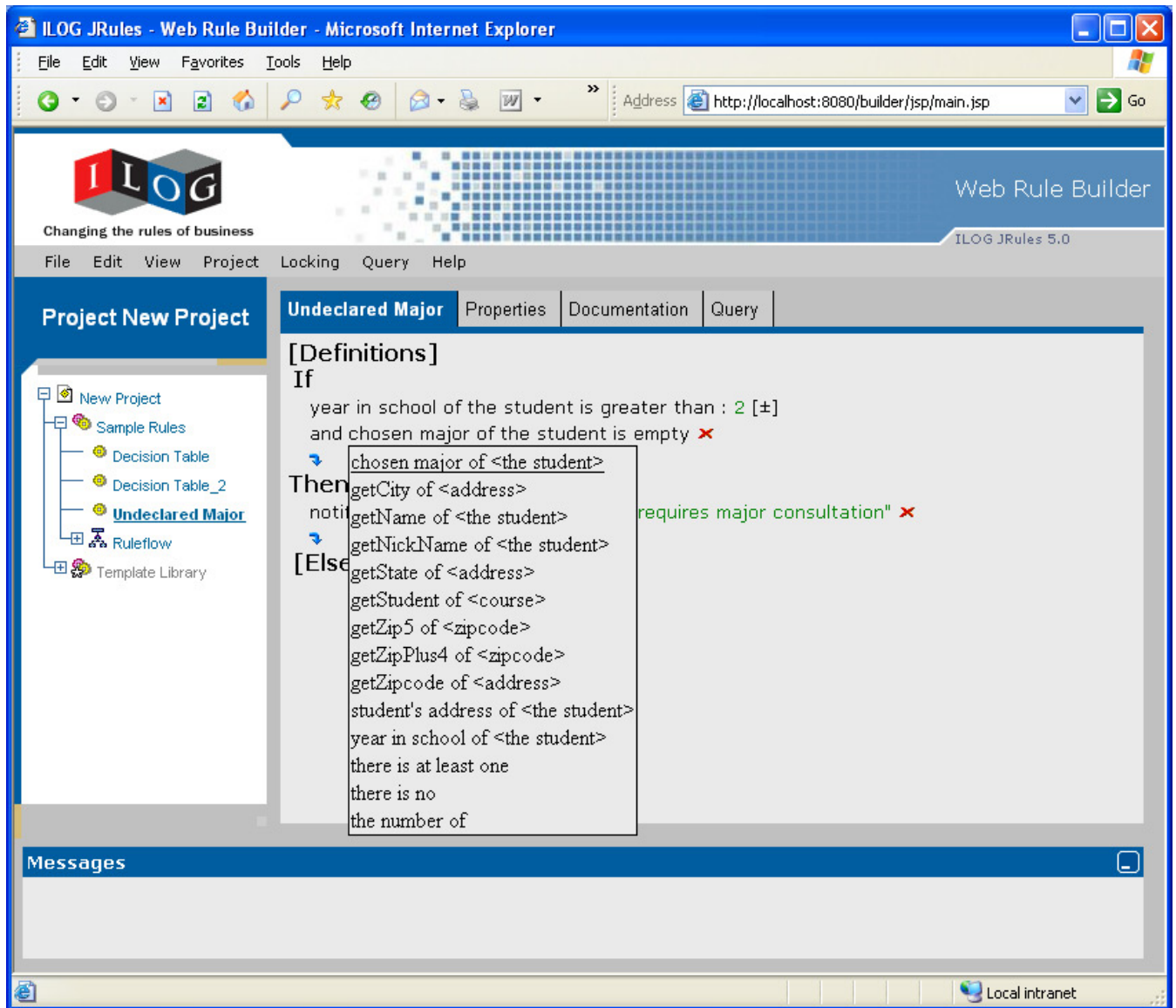


图 15: 在 ILOG JRules Web Builder 中编辑业务规则

ILOG JRules 还支持其它业务规则格式，以更紧凑的方式表示高度结构化的业务规则。这些格式包括决策表（如图 16 中所示）和决策树（如图 17 中所示）。

| | gender | age | | married | state | base premium |
|----|--------|-----|------|---------|------------|--------------|
| | | min | max | | | |
| 0 | | | | | CA, NY, VA | 700 |
| 1 | | | | true | otherwise | 300 |
| 2 | | 16 | 20 | | CA, NY, VA | 720 |
| 3 | | | | false | otherwise | 300 |
| 4 | | | | | CA | 660 |
| 5 | | | | true | NY | 680 |
| 6 | | | | | VA | 640 |
| 7 | | | | | otherwise | 300 |
| 8 | | 21 | 35 | | CA | 680 |
| 9 | Male | | | | NY | 690 |
| 10 | | | | false | VA | 660 |
| 11 | | | | | otherwise | 300 |
| 12 | | | | | CA | 520 |
| 13 | | | | true | NY | 540 |
| 14 | | | | | VA | 520 |
| 15 | | | | | otherwise | 300 |
| 16 | | | ≥ 36 | | CA | 550 |
| 17 | | | | | NY | 570 |
| 18 | | | | false | VA | 540 |

图 16: ILOG JRules Builder 中的决策表

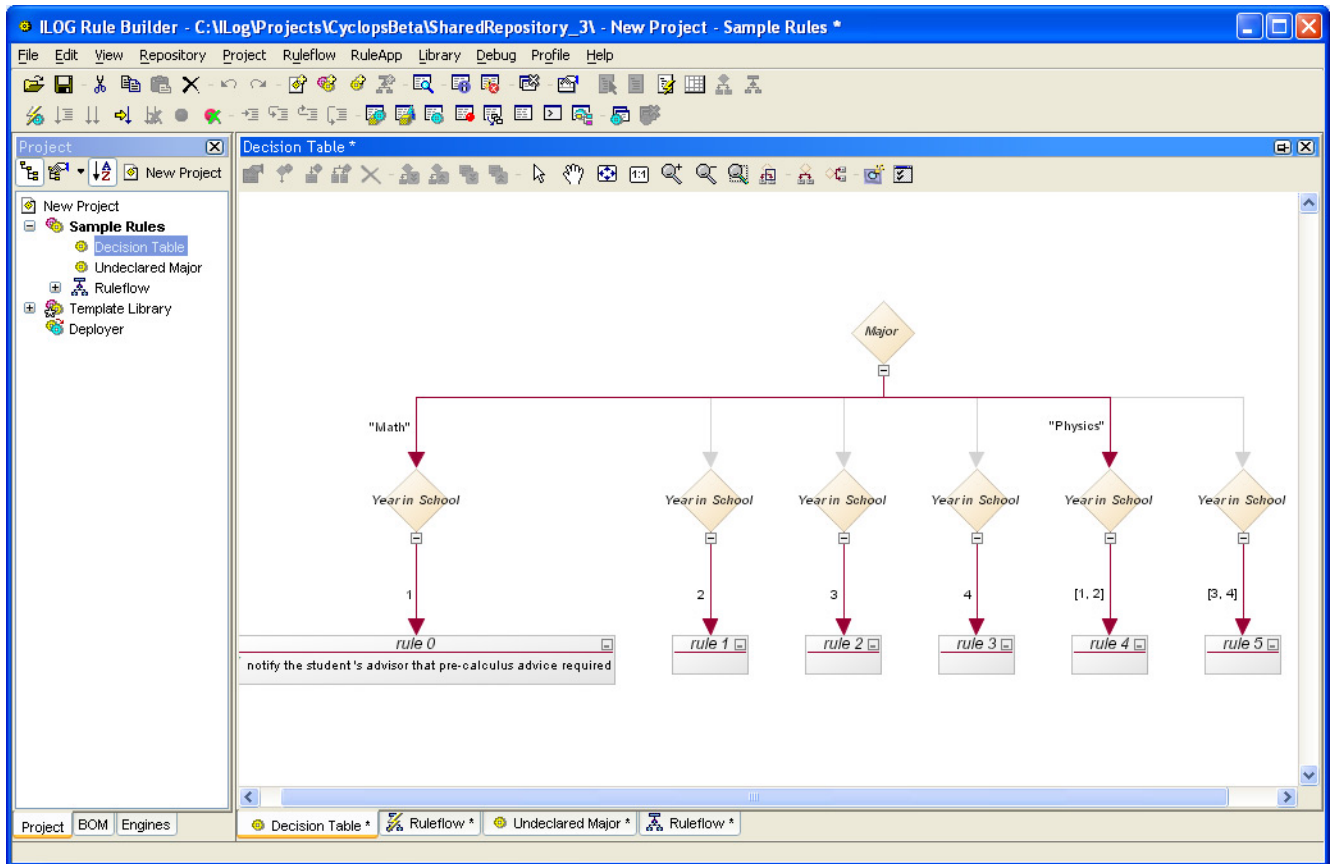


图 17: JRules Builder 中的决策树

3.5.2 规则测试

策略管理者需要验证所编写的业务规则的一致性以及能否正确执行。ILOG JRules 5.0 Builder 提供了业务规则一致性检查功能。策略管理者可利用此特性报告有问题的业务规则，如缺少操作部分、自身不一致、规则包含在其他规则中，以及在相同条件下触发多个规则。

业务规则的功能测试常常依赖于应用。在项目中，可通过 ILOG JRules Builder 应用编程接口来扩展 Builder，从而针对策略管理者直接输入的数据或取自应用数据库的数据进行业务规则功能测试。

3.5.3 规则管理

除了用于基本规则编写和测试的工具，策略管理者还需要用于管理业务规则的功能。例如，策略管理者可能想查看依赖于业务对象模型特定元素的所有规则，或者想要将当前处于“分段实施” (staging) 状态的所有规则的状态都更改为“已部署” (deployed)。ILOG JRules Builder 提供了非常灵活的业务规则查询工具，可实现这些类型的管理功能。例如，图 18 显示了一个查询结构，以查找有效日期为指定日期或指定日期之后的所有业务规则。该示例中的底部窗格显示了查询的结果。

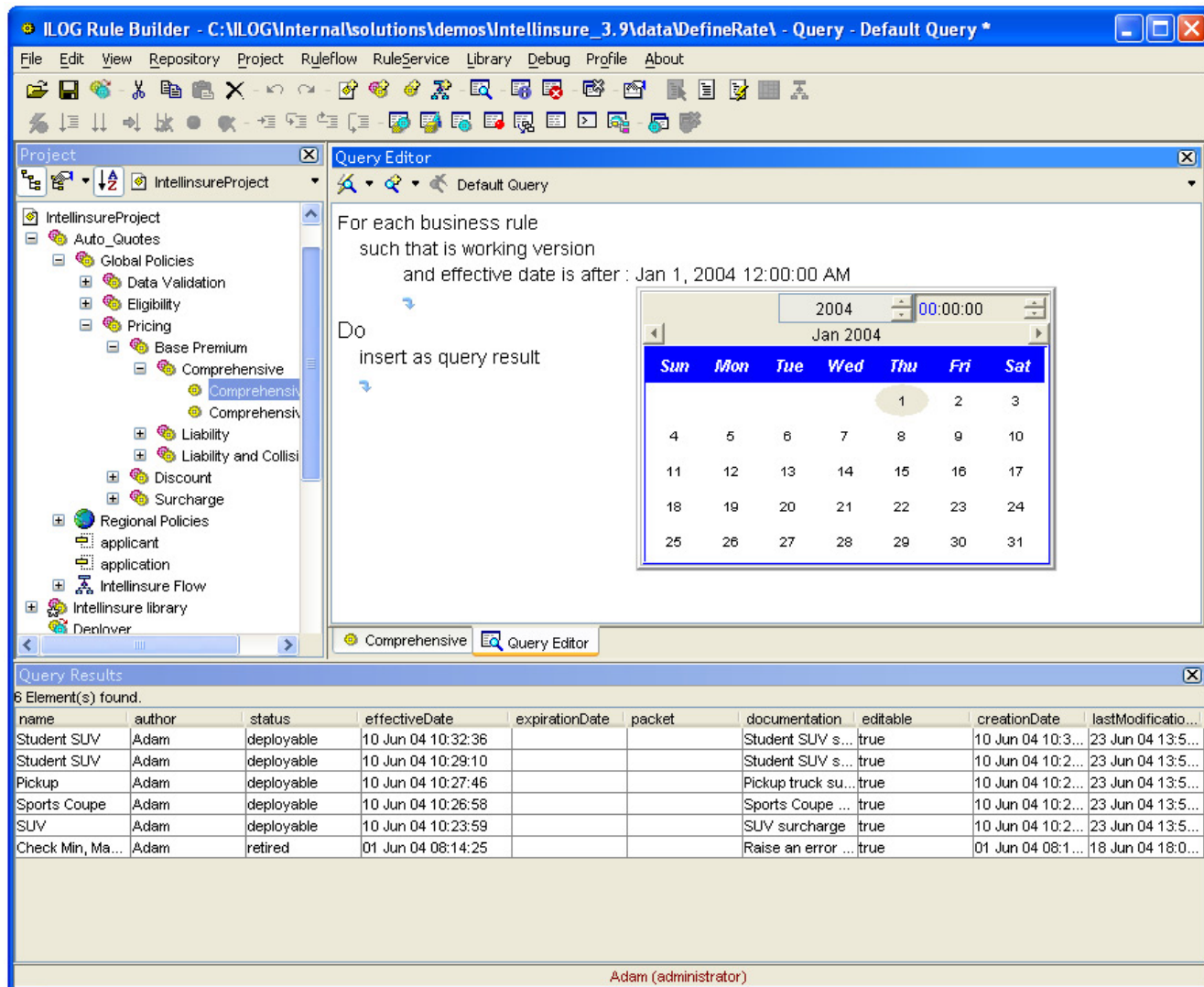


图 18: 使用查询工具

Builder 从规则库结构中选择业务规则，或通过查询工具获得业务规则，并生成相关报告。可以采用 HTML 或 XML 格式导出规则报告。

在适用的情况下，策略管理者还可以直接在 Builder 内为 ILOG JRules BRE 服务器创建部署存档。部署存档的内容可通过开发人员创建的规则集提取程序以静态方式选择，也可以通过查询工具指定。

3.6 系统管理员

系统管理员和业务经理对于业务规则应用的顺利运行起着至关重要的作用。与传统应用不同，在业务规则应用中，业务规则周期可以直接触发对业务行为进行更改。为此，系统管理员需要为业务规则量身定制独立于传统部署工具的部署工具。此外，如果在系统架构中引入规则引擎，则意味着应用还具有其它对性能至关重要的组件，需要系统管理员监控。

ILOG JRules 为系统管理员提供了高效灵活的部署模型。如 3.1.4 一节所述，BRE 服务器是基于 EJB、J2C (JCA) 和 JMX 标准的 J2EE 组件，提供了可管理、可集群化的部署平台。

BRE 服务器中的业务规则部署是由 ILOG JRules “RuleApp” 定义的。RuleApp 是一个容器，其中包含一个或多个规则集以及规则集要求执行的配置资源。RuleApp 被打包到 RuleApp 存档文件中，类似于将 J2EE 应用打包到企业存档 (.ear) 文件中所用的方法。

RuleApp 存档既可以通过 ILOG JRules Builder 以交互方式打包和部署，也可以通过 ANT 脚本，使用 ILOG JRules 提供的自定义 ANT 任务来打包和部署。

RuleApp 存档的部署工具为系统管理员提供了重要的企业特性，包括：

- 对已部署的规则集进行审核和回滚管理
- 支持同时部署和执行单个 RuleApp 的多个版本
- 用于管理多个群集规则引擎的 Web 管理控制台
- 通过 Java 管理扩展 (JMX) 标准与远程管理工具集成
- 为文件系统或数据库选择持久的可执行资源
- 集成用于保护 J2EE 组件的 J2EE 基于角色的安全机制
- 对错误、警告和执行统计信息实施基于 JMX 的实时监控
- 应用编程接口 (API)，可直接访问详细的规则引擎事件

4 小结

将企业业务逻辑从传统的应用代码转移到业务规则管理系统，对于数据量非常大的企业而言，可极大地提高其馈应能力和效率。为此，需要为 IT 专业人员和业务相关人员提供适合各自角色的工具，以执行特定任务。ILOG JRules 5.0 旨在满足系统设计师、业务分析人员、开发人员、策略管理者和系统管理员的业务需要。它是业界首款也是唯一一款业务规则管理系统 (BRMS)，能够满足企业各类相关人员在业务规则应用设计、开发、部署和管理等方面的需要。