Red Hat Process Automation Manager 7.13

Red Hat Process Automation Manager 入门

Last Updated: 2024-03-19

Red Hat Process Automation Manager 7.13 Red Hat Process Automation Manager 入门

法律通告

摘要

本文档论述了如何在 Red Hat Process Automation Manager 中使用决策服务、流程服务和规划解决方案。

目录

前言

作为业务决策和流程的开发人员，您可以使用 Red Hat Process Automation Manager 来使用各种可用资产开发决策服务和流程服务。您还可以使用红帽构建的 OptaPlanner 找到最佳解决方案，以根据一组有限的资源和特定限制来规划问题。

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始： master、slave、blacklist 和 whitelist。这些更改将在即将发行的几个发行本中逐渐实施。详情请查看 *CTO Chris Wright* 的信息。

部分 I. 在 RED HAT PROCESS AUTOMATION MANAGER 中使用决策服务

作为业务规则开发者，您可以在 Red Hat Process Automation Manager 或 VS Code 中的 Red Hat Process Automation Manager DMN 模型者中使用 Business Central 来设计各种决策服务。Red Hat Process Automation Manager 为示例项目提供在 Business Central 中直接作为参考的示例业务资产。本文档论述了如何根据 Business Central 中包含的 traffic **_Violation 示例项目** 创建和测试流量违反示例项目。这个示例项目使用 Decision Model 和 Notation(DMN)模型在流量违反决策服务中定义驱动 penalty 和 suspension 规则。您可以按照本文档中的步骤创建项目及其包含的资产，或者打开并查看现有 traffic **_Violation** 示例项目。

有关 Red Hat Process Automation Manager 中的 DMN 组件和实施的更多信息，*请参阅使用 DMN 模型设计决策服务*。

先决条件

- Red Hat JBoss Enterprise Application Platform 7.4 已安装。有关安装信息，请参阅 *Red Hat JBoss Enterprise Application Platform 7.4 安装指南*。
- Red Hat Process Automation Manager 由 KIE 服务器安装和配置。如需更多信息，请参阅在 *Red Hat JBoss EAP 7.4 上安装和配置 Red Hat Process Automation Manager*。
- Red Hat Process Automation Manager 正在运行，您可以使用 **开发人员** 角色登录到 Business Central。如需更多信息，请参阅 *规划 Red Hat Process Automation Manager 安装* 。

第 1 章 BUSINESS CENTRAL 中的项目和业务资产示例

Business Central 包含具有业务资产的示例项目，您可将其用作您在您自己的 Red Hat Process Automation Manager 项目中所创建的规则、流程或其他资产的参考。每个示例项目都有所不同，旨在展示红帽流程自动化、决策管理或业务优化资产和逻辑。

注意

红帽不提供对 Red Hat Process Automation Manager 发行版本中包含的示例代码的支持。

Business Central 中提供了以下示例项目：

- **courses_Scheduling**:(Business optimization)课程调度和课程决策流程。为房间分配演讲，并根据课程冲突和课程空间容量等因素确定学员的课程。
- **Dinner_ 第三方** ：（业务优化）客户使用指导决策表进行优化。根据每个 guest 的作业类型、自治关系和已知关系来分配客户机名额。
- **Employee_Rostering**:（业务优化）利用决策和解决人员资产提高优化。为员工分配基于技能的转变。
- **评估_流程** ：（流程自动化）使用业务流程资产评估流程。根据性能评估员工。
- **IT_Orders** ：（流程自动化和案例管理）使用业务流程和问题单定义案例。根据需求和批准放置 IT 硬件顺序。
- **mortgages**:（规则管理） Loan 批准过程，使用基于规则的决策资产。根据相关数据和资格确定金级资格。
- **Mortgage_Process**:（流程自动化）使用业务流程和决策资产进行循环审批流程。根据相关数据和资格确定金级资格。
- **OptaCloud** ：（业务优化）使用决策和解决资产进行资源分配优化。为具有有限资源的计算机分配进程。
- **traffic_Violation** ：（使用 DMN 进行 Decision 管理）流量违反决策服务，使用 Decision Model 和 Notation(DMN)模型。根据流量违反情况确定驱动程序 penalty 和 suspension。

1.1. 访问 BUSINESS CENTRAL 中的示例项目和业务资产

您可以使用 Business Central 中的示例项目探索业务资产，作为您在您自己的 Red Hat Process Automation Manager 项目中创建的规则或其他资产的引用。

先决条件

- 业务中心已安装且正在运行。有关安装选项，请参阅 *规划 Red Hat Process Automation Manager 安装*。

流程

1. 在 Business Central 中，转至 **Menu → Design → Projects**。如果已有项目，您可以点击 **MySpace** default 空间并从 **Add Project** 下拉菜单中选择 **Try Samples** 来访问示例。如果没有现有项目，请点击 **Try samples**。

2. 查看每个示例项目的描述，以确定您要浏览的项目。每个示例项目都有所不同，旨在展示红帽流程自动化、决策管理或业务优化资产和逻辑。

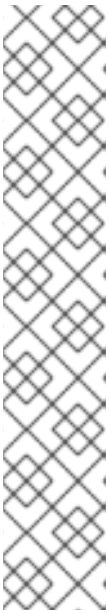
3. 选择一个或多个示例项目，并点击 **Ok** 将项目添加到您的空间中。

4. 在空格的 **Projects** 页面中，选择一个示例项目来查看该项目的资产。

5. 选择每个资产来探索项目如何设计实现指定目标或工作流。某些示例项目包含多个资产页面。点击右上角的左或右箭头查看完整的资产列表。

图 1.1. 资产页面选择

6. 在项目 **资产** 页面右上角，单击 **Build** 来构建示例项目或 **Deploy** 以构建项目，然后将其部署到 KIE 服务器。



注意

您还可以选择 **Build & Install** 选项来构建项目，并将 KJAR 文件发布到配置的 Maven 存储库，而无需部署到 KIE 服务器。在开发环境中，您可以点击 **Deploy** 将构建的 KJAR 文件部署到 KIE 服务器，而无需停止任何正在运行的实例（如果适用），或者点击 **Redeploy** 来部署构建的 KJAR 文件并替换所有实例。下次部署或重新部署构建的 KJAR 时，以前的部署单元（KIE 容器）会在同一目标 KIE 服务器中自动更新。在生产环境中，**Redeploy** 选项被禁用，且只能点击 **Deploy** 将构建的 KJAR 文件部署到 KIE 服务器上的新部署单元（KIE 容器）。

要配置 KIE 服务器环境模式，请将 **org.kie.server.mode** 系统属性设置为 **org.kie.server.mode=development** 或 **org.kie.server.mode=production**。要在 Business Central 中配置对应项目的部署行为，请转至 Project **Settings → General Settings → Version**，切换 **Development Mode** 选项，然后点 **Save**。默认情况下，Business Central 中的 KIE 服务器和所有新项目均为开发模式。您不能部署打开开发模式的项目，或使用手动将 **SNAPSHOT** 版本后缀添加到生产模式中的 KIE 服务器。

要查看项目部署详情，可在屏幕顶部的部署横幅或 Deploy 下拉菜单中点击 View deployment details。这个选项将您定向到 Menu → Deploy → Execution Servers 页面。

第 2 章 RED HAT PROCESS AUTOMATION MANAGER BPMN 和 DMN 型号器

Red Hat Process Automation Manager 提供了以下扩展程序或应用程序，您可以使用它们设计业务流程模型和符号(BPMN)流程模型，以及使用图形模型(DMN)决策模型。

- Business Central ：使您能够在相关嵌入式设计人员中查看和设计 BPMN 模型、DMN 模型和测试方案文件。
  要使用 Business Central，您可以设置一个包含 Business Central 的开发环境，用于设计业务规则和流程，以及 KIE 服务器来执行和测试创建的业务规则和流程。

- Red Hat Process Automation Manager VS Code 扩展：允许您在 Visual Studio Code(VS Code)中查看和设计 BPMN 模型、DMN 模型和测试场景文件。VS Code 扩展需要 VS Code 1.46.0 或更新版本。
  要安装 Red Hat Process Automation Manager VS Code 扩展，请在 VS Code 中选择 Extensions 菜单选项，并搜索并安装 Red Hat Business Automation Bundle扩展。

- 独立 BPMN 和 DMN 编辑器：使您能够查看和设计嵌入在 web 应用程序中的 BPMN 和 DMN 模型。要下载所需的文件，您可以使用 NPM registry 中的 NPM 工件，或直接下载位于 https://<YOUR_PAGE>/dmn/index.js 的 DMN 独立编辑器库的 JavaScript 文件。

2.1. 安装 RED HAT PROCESS AUTOMATION MANAGER VS CODE 扩展捆绑包

Red Hat Process Automation Manager 提供了一个 Red Hat Network Automation Bundle VS Code 扩展，它可让您设计决策模型和符号(DMN)决策模型、业务流程模型和符号(BPMN)2.0 业务流程并直接在 VS Code 中测试场景。VS Code 是开发新业务应用的首选集成开发环境(IDE)。Red Hat Process Automation Manager 还提供单独的 DMN Editor 和 BPMN Editor VS Code 扩展（如果需要）。

重要

VS Code 中的编辑器部分与 Business Central 中的编辑器兼容，并且 VS Code 不支持几个 Business Central 功能。

先决条件

- 安装了 VS Code 的最新稳定版本。

流程

1. 在 VS Code IDE 中，选择 Extensions 菜单选项，并搜索 Red Hat Business Automation Bundle for DMN、Harllse 和 test scenario 文件支持。

   对于 DMN 或 BPMN 文件支持，您还可以搜索单独的 DMN Editor 或 BPMN Editor 扩展。

2. 当 Red Hat Business Automation Bundle 扩展出现在 VS Code 中时，选择它并点 Install。

3. 要获得最佳 VS Code 编辑器行为，请在扩展安装完成后，重新加载或关闭并重新启动 VS Code 实例。

安装 VS Code 扩展捆绑包后，任何 .dmn、.bpmn 或 .bpmn2 文件都会自动显示为图形模型。此外，您打开或创建的 .scesim 文件自动显示为表格测试场景模型，用于测试您的业务决策功能。

如果 DMN、CEP 或测试场景模型器只打开 DMN、Hardb 或 test scenario 文件的 XML 源，并显示错误消息，请查看报告的错误和模型文件，以确保定义所有元素。

注意

对于新的 DMN 或 BPMN 模型，您还可以在网页浏览器中输入 dmn.new 或 BPMn.new，以在在线模型程序中设计 DMN 或 BPMN 模型。完成创建模型后，您可以点击 Download in the online modeler 页面将 DMN 或 BPMN 文件导入到 VS Code 中的 Red Hat Process Automation Manager 项目中。

2.2. 配置 RED HAT PROCESS AUTOMATION MANAGER 独立编辑器

Red Hat Process Automation Manager 提供独立编辑器，这些编辑器在自包含的库中分发，为每个编辑器提供一个一体化 JavaScript 文件。JavaScript 文件使用全面的 API 来设置和控制编辑器。

您可以使用以下方法安装独立编辑器：

- 手动下载每个 JavaScript 文件
- 使用 NPM 软件包

流程

1. 使用以下方法之一安装独立编辑器：

手动下载每个 JavaScript 文件 ： 对于这个方法，请按照以下步骤操作：

a. **下载 JavaScript 文件。**

b. **将下载的 Javascript 文件添加到您的托管应用程序中。**

c. **将以下 &lt ;script& gt; 标签添加到 HTML 页面：**

DMN 编辑器的 HTML 页面标记

```
<script src="https://<YOUR_PAGE>/dmn/index.js"></script>
```

BPMN 编辑器的 HTML 页面的脚本标签

```
<script src="https://<YOUR_PAGE>/bpmn/index.js"></script>
```

使用 NPM 软件包 ： 对于这个方法，请按照以下步骤操作：

a. **在 package.json 文件中添加 NPM 软件包：**

添加 NPM 软件包

```
npm install @kie-tools/kie-editors-standalone
```

b.

将每个编辑器库导入到 TypeScript 文件：

导入每个编辑器

```
import * as DmnEditor from "@kie-tools/kie-editors-standalone/dist/dmn"
import * as BpmnEditor from "@kie-tools/kie-editors-standalone/dist/bpmn"
```

2.

安装独立编辑器后，使用提供的编辑器 API 打开所需的编辑器，如下例所示，打开 DMN 编辑器。每个编辑器的 API 相同。

打开 DMN 独立编辑器

```
const editor = DmnEditor.open({
  container: document.getElementById("dmn-editor-container"),
  initialContent: Promise.resolve(""),
  readOnly: false,
  origin: "",
  resources: new Map([
    [
      "MyIncludedModel.dmn",
      {
        contentType: "text",
        content: Promise.resolve("")
      }
    ]
  ])
});
```

在 editor API 中使用以下参数：

表 2.1. 示例参数

参数	描述
container	附加编辑器的 HTML 元素。

参数	描述
initialContent	对 DMN 模型内容的承诺。这个参数可以为空，如下例所示： • **Promise.resolve("")** • **Promise.resolve("<DIAGRAM_CONTENT_DIRECTLY_HERE>")** • **fetch("MyDmnModel.dmn").then(content ⇒ content.text())**
ReadOnly （可选）	可让您在编辑器中允许更改。在编辑器中将 设置为 **false**（默认）以允许内容编辑和 **true**。
Origin （可选）	仓库的起源。默认值为 **window.location.origin**。
资源 （可选）	编辑器的资源映射。例如，这个参数用于为 DMN 编辑器提供包含的模型，或为 BPMN 编辑器提供工作项目定义。映射中的每个条目都包含一个资源名称和一个对象，它由 **content -type**（**文本 或二进制 ）和内容**组成（与 **初始 Content** 参数类似）。

返回的对象包含操作编辑器所需的方法。

表 2.2. 返回的对象方法

方法	描述
getContent（）：Promise<string>	返回包含编辑器内容的保证。
setContent(path: string, content: string): void	设置编辑器的内容。
getPreview（）：Promise<string>	返回包含当前图表的 SVG 字符串的保证。
subscribeToContentChanges（callback:(isDirty: boolean)IFL void）:(isDirty: boolean)void	当编辑器中的内容更改并返回用于 unsubscription 的回调时，设置要调用的回调。
subscribe toContentChanges（callback: (isDirty: boolean)IFL void）：void	当内容在编辑器中更改时，取消订阅传递的回调。

方法	描述
markAsSaved(): void	重置编辑器状态，这表示已保存编辑器中的内容。另外，它会激活与内容更改相关的订阅回调。
undo（）: void	在编辑器中取消上次更改。另外，它会激活与内容更改相关的订阅回调。
redo（）: void	在编辑器中恢复最近一次撤消的更改。另外，它会激活与内容更改相关的订阅回调。
close（）: void	关闭编辑器。
getElementPosition(selector: string): Promise<Rect>	提供了一种替代方式，可以在可清空或视频组件中的元素中时扩展标准查询选择器。**选择器** 参数必须遵循 <targetNamespaces **>:::<SELECT** > 格式，如 **Canvas:::MySquare** 或 **Video:::PresenterHand**。此方法返回一个代表元素位置的 **Rect**。
envelopeApi: MessageBusClientApi<KogitoEditorEnvelopeApi>	这是高级编辑器 API。有关高级编辑器 API 的更多信息，请参阅 MessageBusClientApi 和 KogitoEditorEnvelopeApi。

第 3 章 使用 MAVEN 创建和执行 DMN 和 BPMN 模型

您可以使用 Maven archetypes 使用 Red Hat Process Automation Manager VS Code 扩展（而非 Business Central）在 VS Code 中开发 DMN 和 BPMN 模型。然后，您可以根据需要将您的 archetypes 与 Red Hat Process Automation Manager 决策和流程服务集成。这种开发 DMN 和 BPMN 模型的方法对使用 Red Hat Process Automation Manager VS Code 扩展构建新的业务应用程序会很有帮助。

流程

1. 在命令终端中，导航到用于存储新 Red Hat Process Automation Manager 项目的本地文件夹。

2. 输入以下命令使用 Maven archtype 在定义的文件夹中生成项目：

使用 Maven archetype 生成项目

```
mvn archetype:generate \
    -DarchetypeGroupId=org.kie \
    -DarchetypeArtifactId=kie-kjar-archetype \
    -DarchetypeVersion=7.67.0.Final-redhat-00024
```

此命令生成包含所需依赖项的 Maven 项目，并生成所需的目录和文件以构建您的业务应用程序。在开发项目时，您可以使用 Git 版本控制系统（推荐）。

如果要在同一目录中生成多个项目，请通过将 -DgroupId= -D artifactId ={> -DartifactId= <artifactId> 添加到 上一个命令来指定生成的业务应用程序的 artifactId 和 groupId。

3. 在 VS Code IDE 中，单击 File，选择 Open Folder，再导航到使用上一命令生成的文件夹。

4. 在创建第一个资产前，请为您的业务应用程序设置一个软件包，例如 org.kie.Businessapp，并在以下路径中创建相应的目录：

- PROJECT_HOME/src/main/java

- PROJECT_HOME/src/main/resources
- PROJECT_HOME/src/test/resources

例如，您可以为 org.kie . Businessapp 创建 PROJECT_HOME/src/main/java /org/kie/businessapp 软件包。

5. 使用 VS Code 为您的业务应用程序创建资产。您可以使用以下方法创建由 Red Hat Process Automation Manager VS Code 扩展支持的资产：

- 要创建业务流程，请在 PROJECT_HOME/src/main/resources/org/kie/businessapp 目录（如 Process .bpmn ）中使用 .bpmn 或 .bpmn 创建新文件。
- 要创建 DMN 模型，请在 PROJECT_HOME/src/main/resources/org/kie/businessapp 目录中创建一个使用 .dmn 的新文件，如 AgeDecision.dmn。
- 要创建测试场景模拟模型，请在 PROJECT_HOME/src/test/resources/org/kie/businessapp 目录中创建一个带有 .scesim 的新文件，如 TestAgeScenario.scesim。

6. 在 Maven archetype 中创建资产后，导航到命令行中项目的 root 目录（包含 pom.xml），再运行以下命令来构建项目的知识库文章(KJAR)：

```
mvn clean install
```

如果构建失败，请解决命令行错误消息中描述的任何问题，并尝试验证项目，直到构建成功为止。但是，如果构建成功，您可以在 PROJECT_HOME/target 目录中找到业务应用程序的工件。

注意

在开发的每一主要更改后，经常使用 mvn clean install 命令验证您的项目。

您可以使用 REST API 在运行的 KIE 服务器上部署您业务应用程序生成的知识 JAR(KJAR)。有关使用 REST API 的更多信息，请参阅使用 *KIE API 与 Red Hat Process Automation Manager 交互*。

第 4 章 在 BUSINESS CENTRAL 中创建流量违反项目

在本例中，创建一个名为 traffic-violation 的新项目。项目是数据对象、DMN 资产和测试场景等资产的容器。创建的示例项目与 Business Central 中的现有 traffic_Vi olation 示例项目类似。

流程

1. 在 Business Central 中，转至 Menu → Design → Projects。

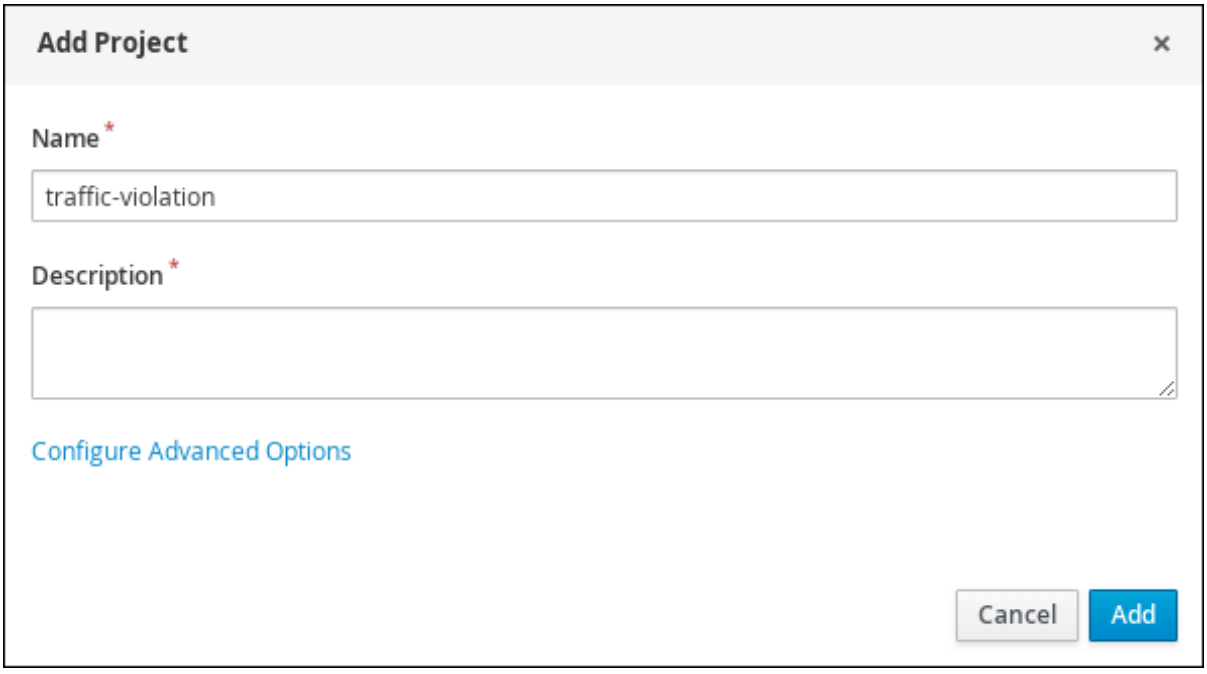
   Red Hat Process Automation Manager 提供了一个名为 MySpace 的默认空间。您可以使用默认空间来创建和测试示例项目。

2. 单击 Add Project。

3. 在 Name 字段中输入 traffic-violation。

4. 点击 Add。

图 4.1. 添加项目窗口



项目 的资源视图将打开。

第 5 章 决策模型和符号(DMN)

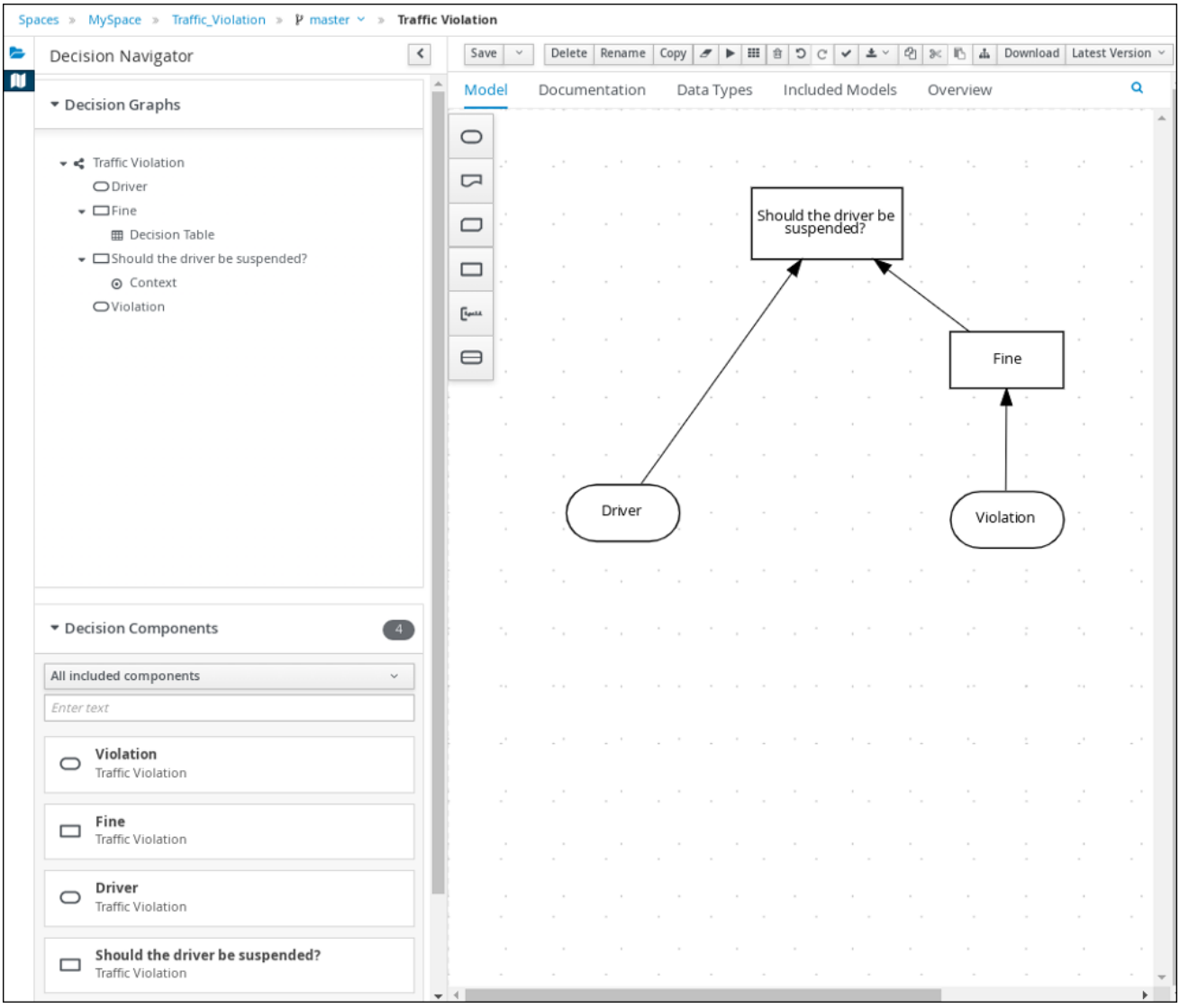
决策模型和符号(DMN)是对象管理组(OMG)制定的标准，用于描述和建模操作决策。DMN 定义了一个 XML 模式，使 DMN 模型可以在 DMN 兼容平台和机构中共享，以便业务分析者和业务规则开发人员能够合作设计和实施 DMN 决策服务。DMN 标准与 相似，可与业务流程建模和符号(BPMN)标准一起使用，以设计和建模业务流程。

有关 DMN 后台和应用的详情，请查看 OMG Decision Model 和 Notation 规格。

5.1. 创建流量违反 DMN 决策要求图(DRD)

决策要求图(DRD)是 DMN 模型的可视化表示。使用 Business Central 中的 DMN 设计器为流量违反项目设计 DRD 并定义 DRD 组件的决策逻辑。

图 5.1. 流量冲突示例 DRD



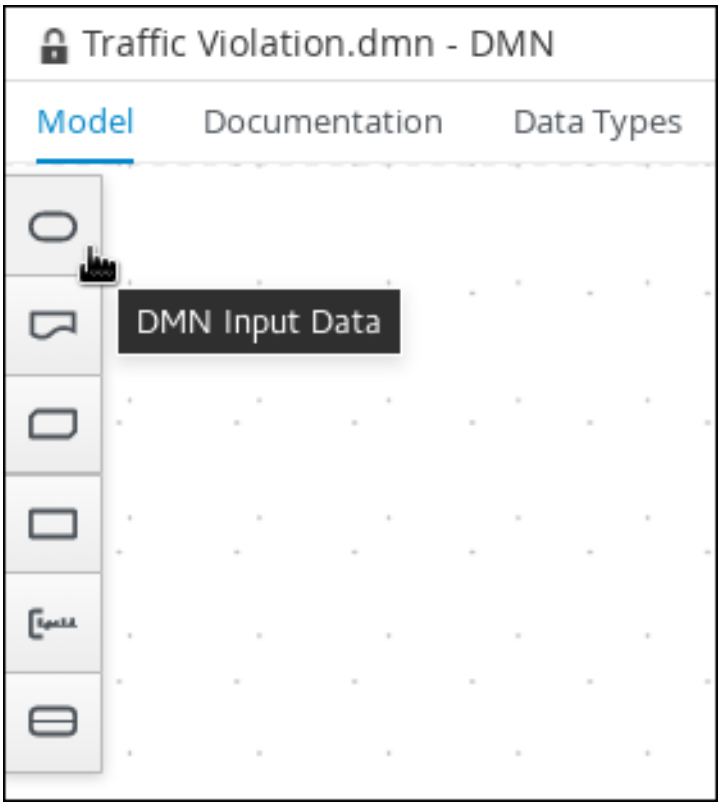
先决条件

- 您已在 Business Central 中创建流量违反项目。

流程

1. 在 traffic-violation 项目的主页上，单击 Add Asset。
2. 在 Add Asset 页面上，点 DMN。此时会打开 Create new DMN 窗口。
   a. 在 Create new DMN 窗口中，在 DMN 名称字段中输入 流量冲突。
   b. 从 Package 列表中，选择 com.myspace.traffic_violation。
   c. 点 确定。DMN 设计器中的 DMN 资产被打开。
3. 在 DMN 设计器 canvas 中，将两个 DMN 输入 数据输入节点拖到 canvas 中。

图 5.2. DMN 输入数据节点



4. 在右上角点击

图标。

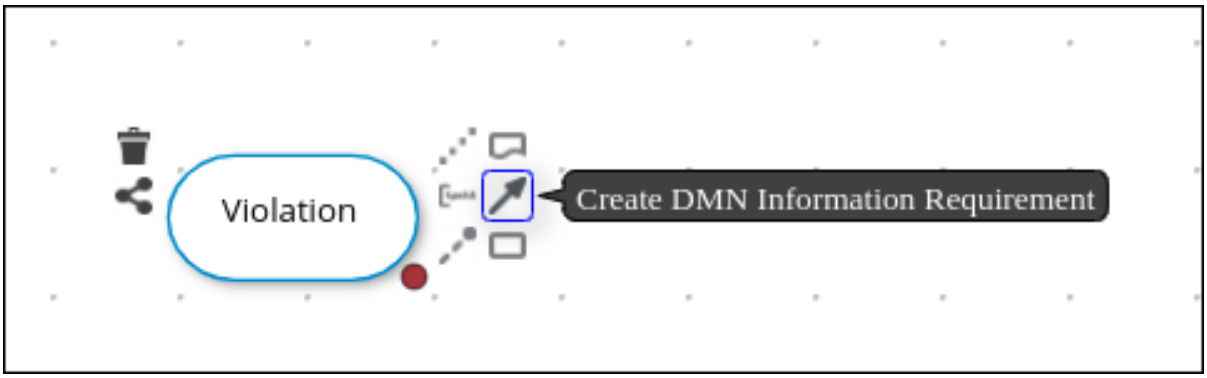
5. 双击输入节点并将 a to Driver 重命名为 Violation。

6. 将 DMN Decision 决策节点拖到 canvas 中。

7. 双击决策节点并将其重命名为 Fine 。

8. 点 Violation 输入节点，选择 Create DMN Information Requirement 图标，然后点击"填充决策节点"链接两个节点。

图 5.3. 创建 DMN 信息要求图标



9. 将 DMN Decision 决策节点拖到 canvas 中。

10. 双击决策节点并将其重命名为 应暂停驱动程序？

11. 单击 Driver 输入节点，选择 Create DMN Information Requirement 图标，然后单击 应该暂停驱动程序？ 决定节点连接两个节点。

12. 点 Fine 决策节点，选择 Create DMN Information Requirement 图标，然后选择 应该暂停驱动程序？ 决策节点。

13. 点击 Save。

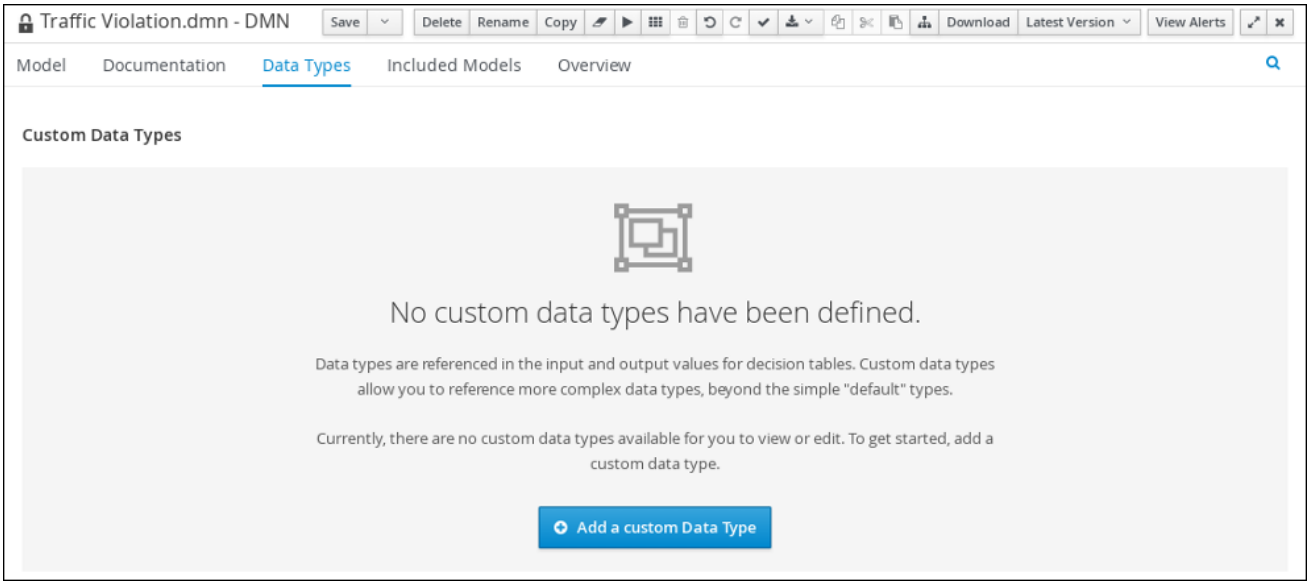
注意

当您定期保存 DRD 时，DMN 设计程序会对 DMN 模型执行静态验证，并可能会生成错误消息，直到模型被完全定义为止。在完全定义 DMN 模型后，如果任何错误保留，请相应地对指定问题进行故障排除。

5.2. 创建流量违反 DMN 自定义数据类型

DMN 数据类型决定了在 DMN 框表达式中用来定义决策逻辑的表、列或字段中使用的数据结构。您可以使用默认 DMN 数据类型（如字符串、数字或布尔值）或者您可以创建自定义数据类型来指定您要为框式表达式值实施的其他字段和约束。在 Business Central 中使用 DMN 设计器 的数据类型 选项卡，为流量违反项目定义自定义数据类型。

图 5.4. 自定义数据类型标签页



下表列出了 tDriver、tViolation 和 tFine 自定义数据类型。

表 5.1. tDriver 自定义数据类型

Name	类型
tDriver	结构
Name	字符串
age	number
状态	字符串
City	字符串

Name	类型
点	number

表 5.2. tViolation 自定义数据类型

Name	类型
tViolation	结构
代码	字符串
Date	date
类型	字符串
速度限制	number
actual Speed	number

表 5.3. TFine 自定义数据类型

Name	类型
tFine	结构
数量	number
点	number

先决条件

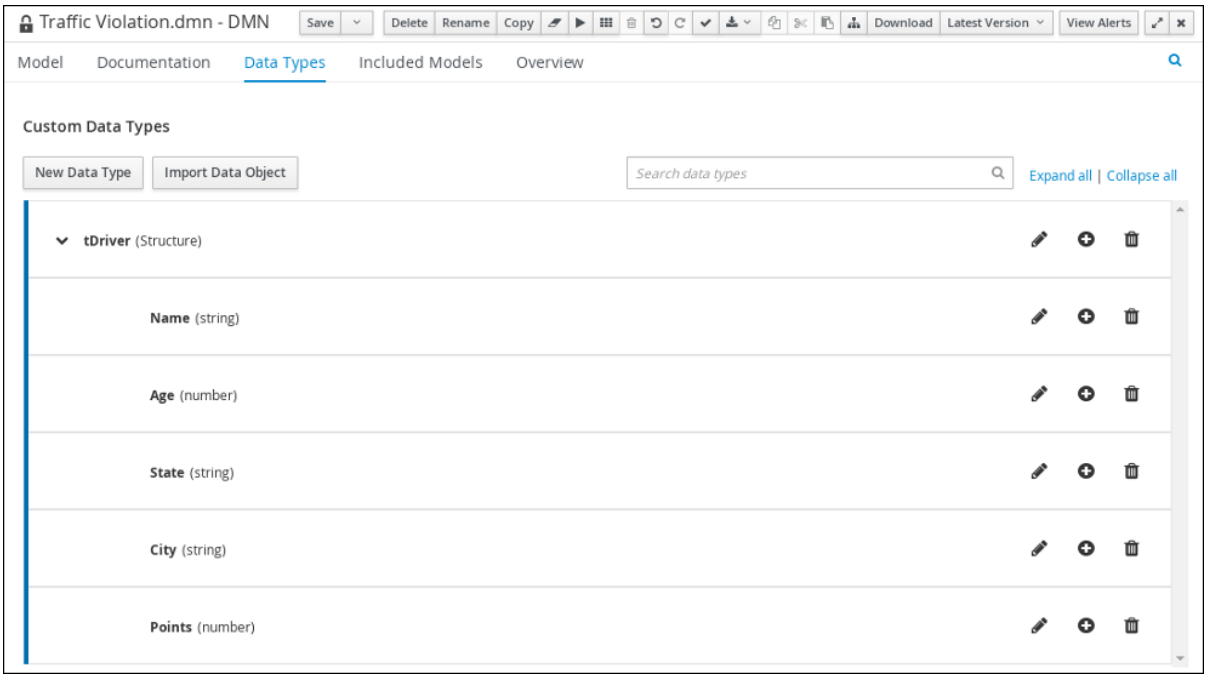
- **您在 Business Central 中创建了流量违反 DMN 决策要求图(DRD)。**

流程

1. **要创建 tDriver custom 数据类型，在 Data Types 标签页中点 Add a custom Data Type，在 Name 字段中输入 tDriver，然后从 Type 列表中选择 Structure。**

2. **点击新数据类型右侧的检查标记保存您的更改。**

图 5.5. tDriver 自定义数据类型



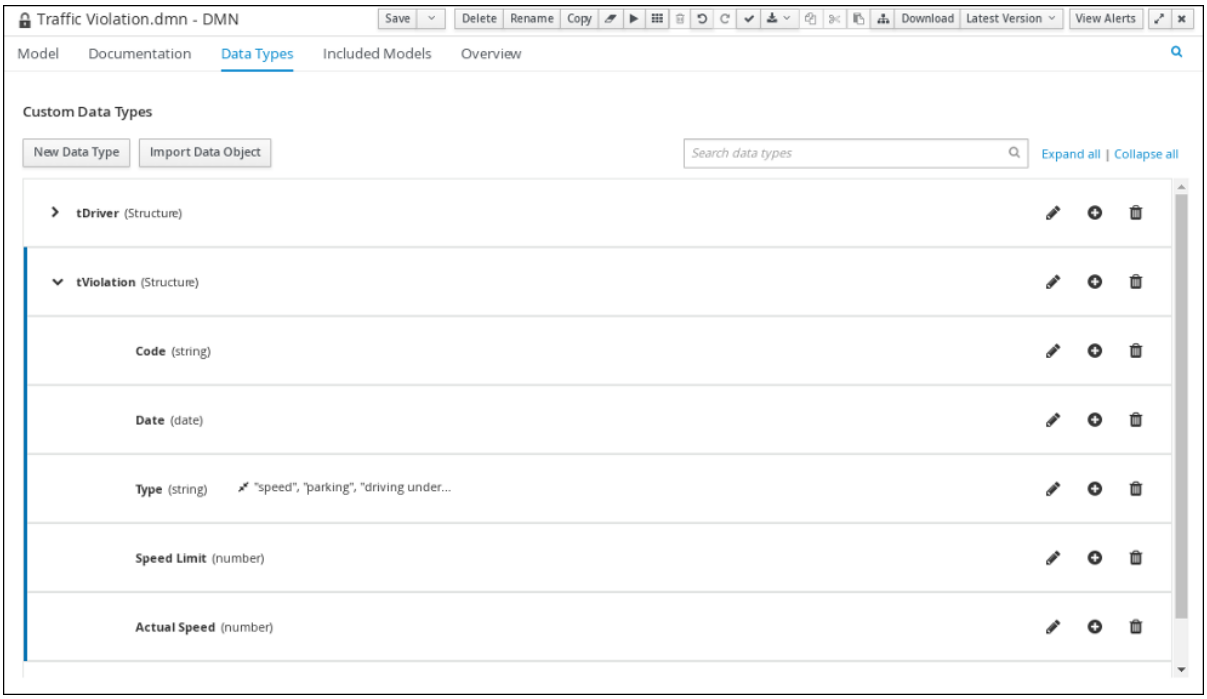
3. 通过单击每个新嵌套的数据类型的 tDriver 旁边的加号，将以下嵌套数据类型添加到 tDriver 中。点击每个新数据类型右侧的检查标记保存您的更改。

- 名称 （字符串）
- age （数字）
- 状态 （字符串）
- City （字符串）
- 点 （数字）

4. 要创建 tViolation custom 数据类型，请单击 New Data Type，在 Name 字段中输入 tViolation，然后从 Type 列表中选择 Structure。

5. 点击新数据类型右侧的检查标记保存您的更改。

图 5.6. tViolation 自定义数据类型



6. 通过单击每个新嵌套数据类型旁边的加号，将以下嵌套数据类型添加到 tViolation 结构化数据类型中。点击每个新数据类型右侧的检查标记保存您的更改。

- Code （字符串）
- 日期 （日期）
- 类型 （字符串）
- 速度限制 （数字）
- 实际 Speed （数字）

7. 要在 Type 嵌套数据类型中添加以下限制，点编辑图标，点 Add Constraints，然后从 Select constraint type 下拉菜单中选择 Enumeration。

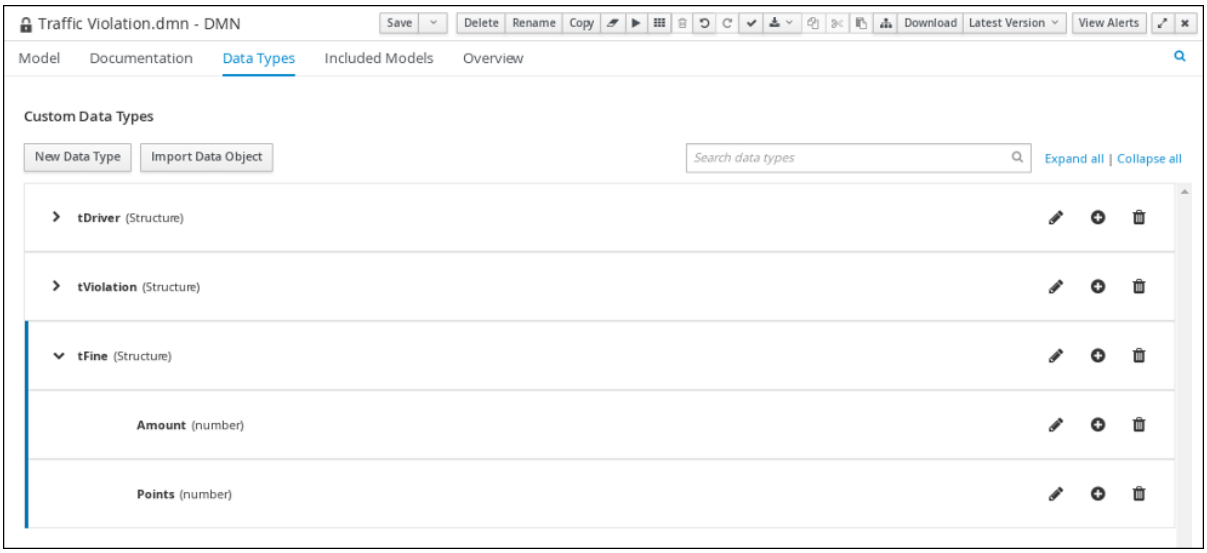
- 速度

- parking
- 驱动影响

8. 单击确定，然后单击 Type 数据类型右侧的检查标记来保存您的更改。

9. 要创建 tFine 自定义数据类型，请单击 New Data Type，在 Name 字段中输入 tFine，从 Type 列表中选择 Structure，然后单击 Save。

图 5.7. tFine 自定义数据类型



10. 通过单击每个新嵌套数据类型旁边的加号，将以下嵌套数据类型添加到 tFine 结构化数据类型中。点击每个新数据类型右侧的检查标记保存您的更改。

- 数量 （数字）
- 点 （数字）

11. 点击 Save。

5.3. 为 DRD 输入和决定节点分配自定义数据类型

创建 DMN 自定义数据类型后，将它们分配到流量违反 DRD 中的相应 DMN 输入 数据和 DMN Decision 节点。

先决条件

- 您已在 Business Central 中创建流量违反 DMN 自定义数据类型。

流程

1. 点 DMN Designer 上的 Model 选项卡，然后点击 DMN 设计器右上角的 Properties 图标公开 DRD 属性。

2. 在 DRD 中，选择 Driver 输入数据节点并在 Properties 面板中，从 Data type 下拉菜单中选择 tDriver。

3. 选择 Violation 输入数据节点，然后从 Data type 下拉菜单中选择 tViolation。

4. 选择 Fine 决策节点，然后从 数据 类型下拉菜单中选择 tFine。

5. 选择 应该暂停驱动程序？ 决定节点并设置以下属性：

   - 数据类型:字符串
   - 问题 ：由于驱动程序许可证的点，应该挂起驱动程序？
   - 允许的回答 ：是、否

6. 点击 Save。

您已将自定义数据类型分配给您的 DRD 的输入和决策节点。

5.4. 定义流量违反 DMN 决策逻辑

要计算细调并决定是否要暂停驱动程序，您可以使用 DMN 决策表和上下文框的表达式定义流量违反 DMN 决策逻辑。

图 5.8. 精细表达式

Fine *(Decision Table)*

U	Violation.Type *(string)*	Violation.Actual Speed - Violation.Speed Limit *(number)*	Fine *(tFine)*		Enter Text
			Amount *(number)*	Points *(number)*	
1	"speed"	[10..30)	500	3	
2	"speed"	>= 30	1000	7	
3	"parking"	-	100	1	
4	"driving under the influence"	-	1000	5	

图 5.9. 应暂停驱动程序的表达式

Should the driver be suspended? *(Context)*

#	Should the driver be suspended? *(string)*	
1	Total Points *(number)*	Driver.Points + Fine.Points
	<result>	if Total Points >= 20 then "Yes" else "No"

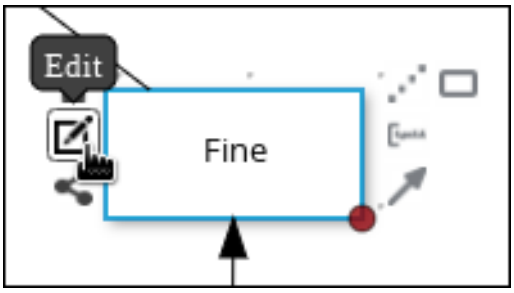
先决条件

- 您已将 DMN 自定义数据类型分配给 Business Central 中流量违反 DRD 中的相应决定和输入节点。

流程

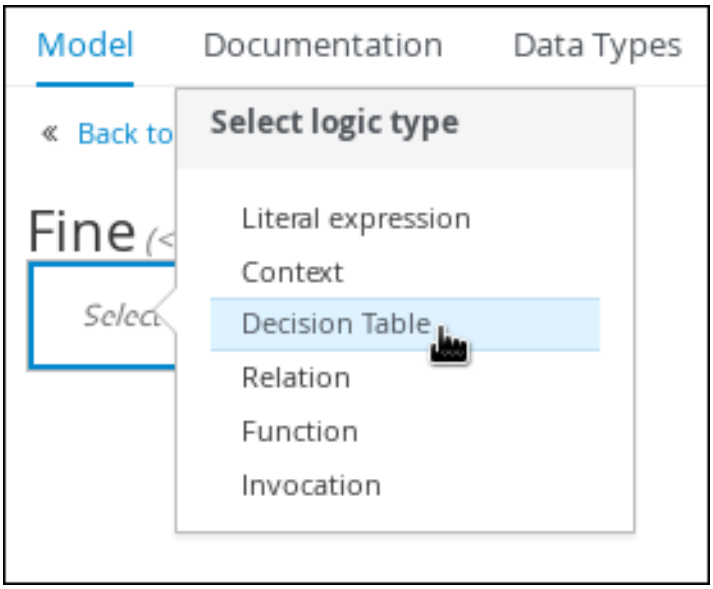
1. 要计算此情况，请在 DMN 设计器 canvas 中选择 Fine 决策节点，然后点击 Edit 图标打开 DMN 框的表达式设计程序。

图 5.10. 决策节点编辑图标



2. 点 Select expression → Decision Table。

图 5.11. 选择 Decisiong Table 逻辑类型



3. 对于 Violation.Date、Violation.Code 和 Violation.Speed Limit 列，请右键为每个字段选择 Delete。

4. 点 Violation.Actual Speed 列标题，然后在 Expression 字段中输入表达式 Violation.Actual Speed - Violation.Speed Limit "。

5. 在决策表的第一行中输入以下值：

- violation.Type:"speed"
- violation.Actual Speed - Violation.Speed Limit:[10..30）

- 计数 ：500
- 点 ：3

右键单击第一行，再选择 Insert 以 添加另一个行。

6. 在决策表的第二行中输入以下值：

- violation.Type:"speed"
- violation.Actual Speed - Violation.Speed Limit：>= 30
- 数量:1000
- 点:7

右击第二行，再选择 Insert 以 添加另一个行。

7. 在决策表的第三个行中输入以下值：

- violation.Type:"parking"
- violation.Actual Speed - Violation.Speed Limit:-
- 数量 ：100
- 点:1

右击第三行并选择 下面的 Insert 以添加另一个行。

8. 在决策表的第四个行中输入以下值：

- violation.Type : "指定影响"
- violation.Actual Speed - Violation.Speed Limit:-
- 数量:1000
- 点 ：5

9. 点击 Save。

10. 要定义驱动程序挂起规则，返回 DMN 设计器 canvas，选择 应暂停驱动程序？ 决策节点，并点击 Edit 图标打开 DMN 框式表达式设计程序。

11. 点 Select expression → Context。

12. 点 ContextEntry-1，输入 Total Points 作为 Name，然后从 Data Type 下拉菜单中选择 数字。

13. 单击 Total Points 旁边的单元，从上下文菜单中选择 Literal 表达式，并输入 Driver.Points + Fine.Points 作为表达式。

14. 在下面的单元格中，指出 + Fine.Points，从上下文菜单中选择 Literal Expression，并输入 if Total Points >= 20 then "Yes" other "No"。

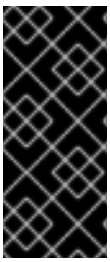
15. 点击 Save。

您已定义了如何计算细和上下文，以决定何时挂起驱动程序。您可以导航到 traffic-violation 项目页面，再单击 Build 来构建 example 项目，并解决 Alerts 面板中提到的任何错误。

第 6 章 测试场景

通过测试 Red Hat Process Automation Manager 中的场景，您可以在将其部署到生产环境前验证业务规则和业务规则数据的功能（适用于基于 DMN 的测试场景）。通过测试场景，您可以使用项目中的数据根据一个或多个定义的业务规则来设置给定条件和预期结果。当您运行该场景时，会比较规则实例的预期结果和实际结果。如果预期的结果与实际结果匹配，则测试会成功。如果预期结果与实际结果不匹配，则测试会失败。

Red Hat Process Automation Manager 包括新的 测试场景 设计人员和之前的 测试场景(Legacy) 设计人员。默认设计器是新的测试场景设计器，它支持测试规则和 DMN 模型，并提供测试场景的增强整体用户体验。如果需要，您可以继续使用旧的测试场景程序，该设计只支持基于规则的测试场景。



重要

旧的测试场景设计程序从 Red Hat Process Automation Manager 版本 7.3.0 中弃用。在以后的 Red Hat Process Automation Manager 发行版本中会删除它。改为使用新的测试场景设计程序。

您可以使用多种方法运行定义的测试场景，例如，您可以在项目级别或特定测试场景资产中运行可用的测试场景。测试场景是独立的，不会影响或修改其他测试场景。您可以在 Business Central 项目开发过程中随时运行测试场景。您不必编译或部署决定服务来运行测试场景。

您可以将数据对象从不同软件包导入到与测试场景相同的项目软件包。默认导入同一软件包中的资产。创建必要的数据对象和测试场景后，您可以使用测试场景设计器的 Data Objects 选项卡来验证所有必需的数据对象是否已通过 添加新项 来导入其他现有数据对象。

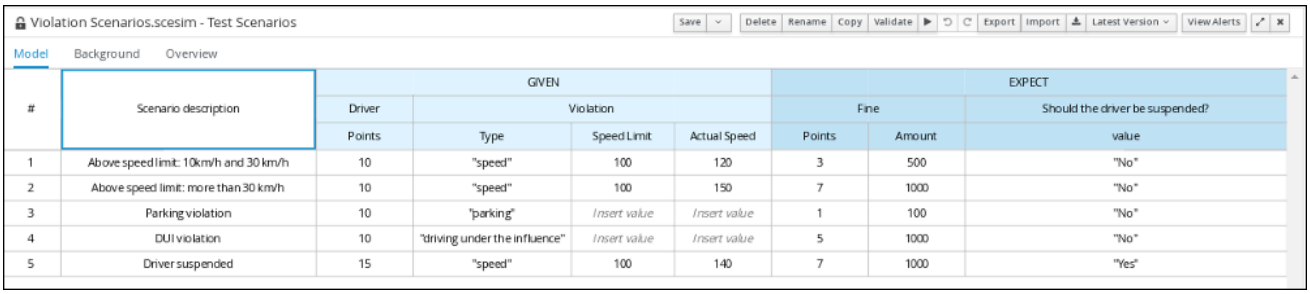
重要

在测试场景文档中，*测试场景* 和测试 *场景设计人员* 的所有引用都适用于新版本，除非明确声明为旧版本。

6.1. 使用测试场景测试流量违反情况

使用 Business Central 中的测试测试 DMN 决策要求图(DRD)的测试，并为流量违反项目定义决策逻辑。

图 6.1. 测试流量违反情况示例



Violation Scenarios.scesim - Test Scenarios

Save | Delete | Rename | Copy | Validate | Export | Import | Latest Version | View Alerts

Model | Background | Overview

#	Scenario description	GIVEN				EXPECT		
		Driver	Violation			Fine		Should the driver be suspended?
		Points	Type	Speed Limit	Actual Speed	Points	Amount	value
1	Above speed limit: 10km/h and 30 km/h	10	"speed"	100	120	3	500	"No"
2	Above speed limit: more than 30 km/h	10	"speed"	100	150	7	1000	"No"
3	Parking violation	10	"parking"	*Insert value*	*Insert value*	1	100	"No"
4	DUI violation	10	"driving under the influence"	*Insert value*	*Insert value*	5	1000	"No"
5	Driver suspended	15	"speed"	100	140	7	1000	"Yes"

先决条件

- 您已成功构建了 Business Central 中的流量违反项目。

流程

1. 在 流量概览 项目主页上，单击 Add Asset 以打开 Add Asset 屏幕。
2. 点 Test Scenario 打开 Create new Test Scenario 对话框。
   a. 在 Test Scenario 字段中输入 Violation Scenarios。
   b. 从 Package 列表中，选择 com.myspace.traffic_violation。
   c. 选择 DMN 作为 Source 类型。
   d. 在 Choose a DMN asset 列表中，选择 DMN 资产的路径。
   e. 单击 Ok 以在 Test Scenarios 设计器中打开 Violation Scenarios 测试场景。
3. 在 Driver 列 sub-header 下，右键单击 状态、城市、Age 和 Name 值单元，然后从上下文菜单中选择 Delete 列 以删除它们。
4. 在 Violation column sub-header 下，右键点击 Date 和 Code value cells，然后选择 Delete 列来删除它们。

5. 在测试场景的第一行中输入以下信息：

   - 场景描述：Above speed limit: 10km/h 和 30 km/h
   - 点 （ 在列标题 下） ：10
   - 类型 ："快速"
   - 速度限制 ：100
   - 实际 Speed:120
   - 点 ：3
   - 计数 ：500
   - 应暂停驱动程序？" 否"

   右击第一行并选择 下面的 Insert 行，以添加另一个行。

6. 在测试场景的第二行中输入以下信息：

   - 场景描述:Above speed limit: more than 30 km/h
   - 点 （ 在列标题 下） ：10
   - 类型 ："快速"

- 速度限制 ：100
- 实际 Speed:150
- 点:7
- 数量:1000
- 应暂停驱动程序？" 否"

  右键单击第二行并选择 下面的 Insert 行，以添加另一个行。

7. 在测试场景的第三行中输入以下信息：

- 场景描述：违反情况
- 点 （ 在列标题 下） ：10
- 键入:"parking"
- speed Limit: 留空
- actual Speed ：留空
- 点:1
- 数量 ：100

- 应暂停驱动程序？" 否"

  右击第三行并选择 下面的 Insert 行，以添加另一个行。

8. 在测试场景的第四个行中输入以下信息：

   - 场景描述:DUI 违反情况
   - 点 （ 在列标题 下） ：10
   - 键入:"driving in the influence"
   - speed Limit: 留空
   - actual Speed ：留空
   - 点 ：5
   - 数量:1000
   - 应暂停驱动程序？" 否"

     右键点击第四行并选择 下面的 Insert 行 来添加另一个行。

9. 在测试场景的第五个行中输入以下信息：

   - 场景描述:暂停驱动程序
   - 点 （ 在列标题 下） ：15

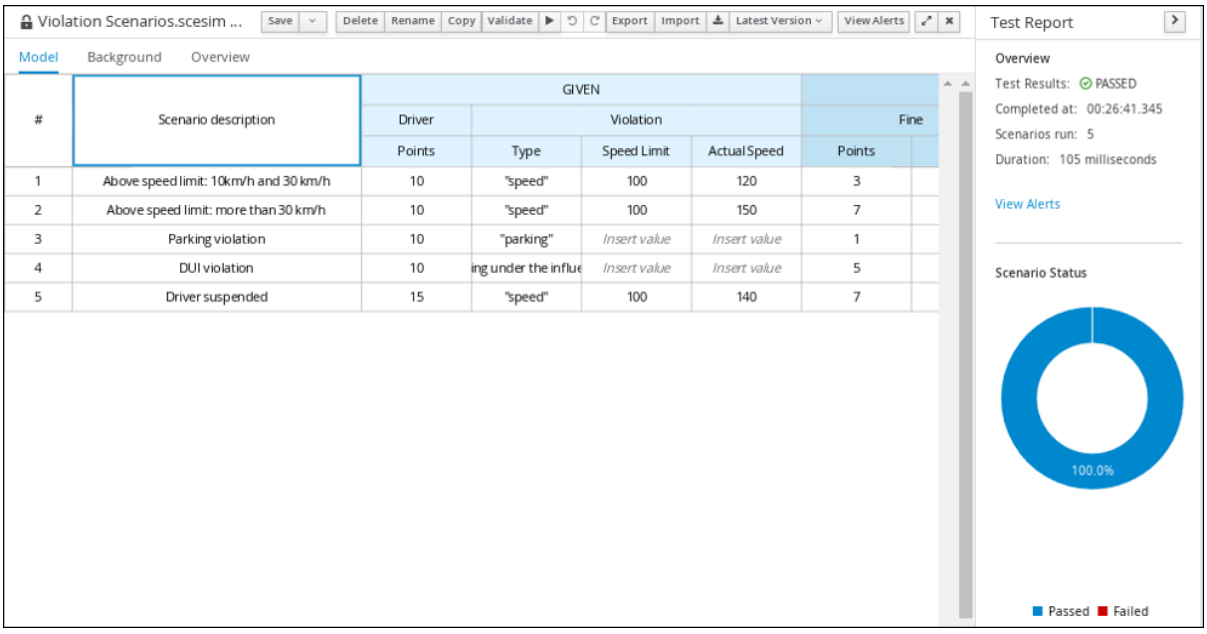
- 类型 ："快速"
- 速度限制 ：100
- 实际 Speed:140
- 点:7
- 数量:1000
- 应暂停驱动程序吗？" 是"

10. 点击 **Save**。

11. 点 **Play** 图标

检查测试场景通过或失败。

图 6.2. 测试流量违反情况示例的场景执行结果



如果出现故障，请更正错误，然后再次运行测试场景。

第 7 章 DMN 模型执行

您可以使用 Business Central 在 Red Hat Process Automation Manager 项目中创建或导入 DMN 文件，或者在没有 Business Central 的情况下将 DMN 文件打包为项目知识 JAR(KJAR)文件的一部分。在 Red Hat Process Automation Manager 项目中实现 DMN 文件后，您可以通过部署包含 KIE Server 的 KIE 服务器并使用 KIE 服务器 REST API 与容器交互来执行 DMN 决策服务。

有关在项目打包和部署方法中包含外部 DMN 资产的信息，请参阅打包和部署 *Red Hat Process Automation Manager 项目*。

7.1. 使用 KIE 服务器 REST API 执行 DMN 服务

与 KIE 服务器的 REST 端点直接交互，提供调用代码和决策逻辑定义之间的最分离。调用代码完全没有直接依赖项，您可以在一个完全不同的开发平台（如 Node.js 或 .NET ）中实施。本节中的示例演示 Nix 风格的 curl 命令，但提供相关信息来适应任何 REST 客户端。

当您使用 KIE 服务器的 REST 端点时，最佳实践是定义一个域对象 POJO Java 类，使用标准 KIE 服务器 marshalling 注解标注。例如，以下代码使用正确注解的域对象 Person 类：

POJO Java 类示例

```
@javax.xml.bind.annotation.XmlAccessorType(javax.xml.bind.annotation.XmlAccessType.FIELD)
public class Person implements java.io.Serializable {

    static final long serialVersionUID = 1L;

    private java.lang.String id;
    private java.lang.String name;

@javax.xml.bind.annotation.adapters.XmlJavaTypeAdapter(org.kie.internal.jaxb.LocalDateXmlAdapter.class)
    private java.time.LocalDate dojoining;

    public Person() {
    }

    public java.lang.String getId() {
        return this.id;
    }

    public void setId(java.lang.String id) {
        this.id = id;
    }
```

```
    public java.lang.String getName() {
        return this.name;
    }

    public void setName(java.lang.String name) {
        this.name = name;
    }

    public java.time.LocalDate getDojoining() {
        return this.dojoining;
    }

    public void setDojoining(java.time.LocalDate dojoining) {
        this.dojoining = dojoining;
    }

    public Person(java.lang.String id, java.lang.String name,
            java.time.LocalDate dojoining) {
        this.id = id;
        this.name = name;
        this.dojoining = dojoining;
    }

}
```

有关 KIE 服务器 REST API 的更多信息，请参阅使用 *KIE API 与 Red Hat Process Automation Manager 交互*。

先决条件

- KIE 服务器是安装和配置的，包括具有 kie-server 角色的用户的已知用户名和凭证。有关安装选项，请参阅 *规划 Red Hat Process Automation Manager 安装*。
- 您已将 DMN 项目构建为 KJAR 工件并将其部署到 KIE 服务器：

```
mvn clean install
```

  有关项目打包和部署以及可执行模型的更多信息，请参阅打包和部署 *Red Hat Process Automation Manager 项目*。

- 您有包含 DMN 模型的 KIE 容器的 ID。如果存在多个模型，还必须知道相关模型的型号命名空间和型号名称。

流程

1. **确定用于访问 KIE 服务器 REST API 端点的基础 URL。这需要了解以下值（使用默认本地部署值作为示例）：**

   - **主机（本地主机）**
   - **端口(8080)**
   - **根上下文(kie-server)**
   - **基本 REST 路径(services/rest/)**

   流量违反项目的本地部署中的基本 URL 示例：

   http://localhost:8080/kie-server/services/rest/server/containers/traffic-violation_1.0.0-SNAPSHOT

2. **确定用户身份验证要求。**

   当在 KIE 服务器配置中直接定义用户时，使用 HTTP 基本身份验证并需要用户名和密码。成功请求需要该用户具有 kie-server 角色。

   以下示例演示了如何在 curl 请求中添加凭证：

   ```
   curl -u username:password <request>
   ```

   如果使用 Red Hat Single Sign-On 配置 KIE 服务器，则请求必须包含 bearer 令牌：

   ```
   curl -H "Authorization: bearer $TOKEN" <request>
   ```

3. **指定请求和响应的格式。REST API 端点同时使用 JSON 和 XML 格式，并使用请求标头来设**

置：

JSON

```
curl -H "accept: application/json" -H "content-type: application/json"
```

XML

```
curl -H "accept: application/xml" -H "content-type: application/xml"
```

4. **可选：查询容器以获取部署决策模型列表：**

   [GET] server/containers/{containerId}/dmn

   curl 请求示例：

```
curl -u wbadmin:wbadmin -H "accept: application/xml" -X GET "http://localhost:8080/kie-server/services/rest/server/containers/traffic-violation_1.0.0-SNAPSHOT/dmn"
```

   XML 输出示例：

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response type="SUCCESS" msg="Ok models successfully retrieved from container 'traffic-violation_1.0.0-SNAPSHOT'">
   <dmn-model-info-list>
      <model>
         <model-namespace>https://kiegroup.org/dmn/_60b01f4d-e407-43f7-848e-258723b5fac8</model-namespace>
         <model-name>Traffic Violation</model-name>
         <model-id>_2CD7D1AA-BD84-4B43-AD21-B0342ADE655A</model-id>
         <decisions>
            <dmn-decision-info>
               <decision-id>_23428EE8-DC8B-4067-8E67-9D7C53EC975F</decision-id>
```

```
            <decision-name>Fine</decision-name>
          </dmn-decision-info>
          <dmn-decision-info>
            <decision-id>_B5EEE2B1-915C-44DC-BE43-C244DC066FD8</decision-id>
            <decision-name>Should the driver be suspended?</decision-name>
          </dmn-decision-info>
        </decisions>
        <inputs>
          <dmn-inputdata-info>
            <inputdata-id>_CEB959CD-3638-4A87-93BA-03CD0FB63AE3</inputdata-id>
            <inputdata-name>Violation</inputdata-name>
            <inputdata-typeref>
              <namespace-uri>https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8</namespace-uri>
              <local-part>tViolation</local-part>
              <prefix></prefix>
            </inputdata-typeref>
          </dmn-inputdata-info>
          <dmn-inputdata-info>
            <inputdata-id>_B0E810E6-7596-430A-B5CF-67CE16863B6C</inputdata-id>
            <inputdata-name>Driver</inputdata-name>
            <inputdata-typeref>
              <namespace-uri>https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8</namespace-uri>
              <local-part>tDriver</local-part>
              <prefix></prefix>
            </inputdata-typeref>
          </dmn-inputdata-info>
        </inputs>
        <itemdefinitions>
          <dmn-itemdefinition-info>
            <itemdefinition-id>_9C758F4A-7D72-4D0F-B63F-
2F5B8405980E</itemdefinition-id>
            <itemdefinition-name>tViolation</itemdefinition-name>
            <itemdefinition-itemcomponent>
              <dmn-itemdefinition-info>
                <itemdefinition-id>_0B6FF1E2-ACE9-4FB3-876B-
5BB30B88009B</itemdefinition-id>
                <itemdefinition-name>Code</itemdefinition-name>
                <itemdefinition-typeref>
                  <namespace-uri>https://kiegroup.org/dmn/_60b01f4d-e407-43f7-848e-
258723b5fac8</namespace-uri>
                  <local-part>string</local-part>
                  <prefix></prefix>
                </itemdefinition-typeref>
                <itemdefinition-itemcomponent/>
                <itemdefinition-iscollection>false</itemdefinition-iscollection>
              </dmn-itemdefinition-info>
              <dmn-itemdefinition-info>
                <itemdefinition-id>_27A5DA18-3CA7-4C06-81B7-
CF7F2F050E29</itemdefinition-id>
                <itemdefinition-name>date</itemdefinition-name>
                <itemdefinition-typeref>
                  <namespace-uri>https://kiegroup.org/dmn/_60B01F4D-E407-43F7-
848E-258723B5FAC8</namespace-uri>
                  <local-part>date</local-part>
```

```
                    <prefix></prefix>
                  </itemdefinition-typeref>
                  <itemdefinition-itemcomponent/>
                  <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
                <dmn-itemdefinition-info>
                  <itemdefinition-id>_8961969A-8A80-4F12-B568-
346920C0F038</itemdefinition-id>
                  <itemdefinition-name>type</itemdefinition-name>
                  <itemdefinition-typeref>
                    <namespace-uri>https://kiegroup.org/dmn/_60B01F4D-E407-43F7-
848E-258723B5FAC8</namespace-uri>
                    <local-part>string</local-part>
                    <prefix></prefix>
                  </itemdefinition-typeref>
                  <itemdefinition-itemcomponent/>
                  <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
                <dmn-itemdefinition-info>
                  <itemdefinition-id>_7450F12A-3E95-4D5E-8DCE-
2CB1FAC2BDD4</itemdefinition-id>
                  <itemdefinition-name>speed limit</itemdefinition-name>
                  <itemdefinition-typeref>
                    <namespace-uri>https://kiegroup.org/dmn/_60b01f4d-e407-43f7-848e-
258723b5fac8</namespace-uri>
                    <local-part>number</local-part>
                    <prefix></prefix>
                  </itemdefinition-typeref>
                  <itemdefinition-itemcomponent/>
                  <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
                <dmn-itemdefinition-info>
                  <itemdefinition-id>_0A9A6F26-6C14-414D-A9BF-
765E5850429A</itemdefinition-id>
                  <itemdefinition-name>Actual Speed</itemdefinition-name>
                  <itemdefinition-typeref>
                    <namespace-uri>https://kiegroup.org/dmn/_60B01F4D-E407-43F7-
848E-258723B5FAC8</namespace-uri>
                    <local-part>number</local-part>
                    <prefix></prefix>
                  </itemdefinition-typeref>
                  <itemdefinition-itemcomponent/>
                  <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
              </itemdefinition-itemcomponent>
              <itemdefinition-iscollection>false</itemdefinition-iscollection>
            </dmn-itemdefinition-info>
            <dmn-itemdefinition-info>
              <itemdefinition-id>_13C7EFD8-B85C-43BF-94D3-
14FABE39A4A0</itemdefinition-id>
              <itemdefinition-name>tDriver</itemdefinition-name>
              <itemdefinition-itemcomponent>
                <dmn-itemdefinition-info>
                  <itemdefinition-id>_EC11744C-4160-4549-9610-
2C757F40DFE8</itemdefinition-id>
                  <itemdefinition-name>Name</itemdefinition-name>
```

```
                    <itemdefinition-typeref>
                        <namespace-uri>https://kiegroup.org/dmn/_60B01F4D-E407-43F7-
848E-258723B5FAC8</namespace-uri>
                        <local-part>string</local-part>
                        <prefix></prefix>
                    </itemdefinition-typeref>
                    <itemdefinition-itemcomponent/>
                    <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
                <dmn-itemdefinition-info>
                    <itemdefinition-id>_E95BE3DB-4A51-4658-A166-
02493EAAC9D2</itemdefinition-id>
                    <itemdefinition-name>Age</itemdefinition-name>
                    <itemdefinition-typeref>
                        <namespace-uri>https://kiegroup.org/dmn/_60B01F4D-E407-43F7-
848E-258723B5FAC8</namespace-uri>
                        <local-part>number</local-part>
                        <prefix></prefix>
                    </itemdefinition-typeref>
                    <itemdefinition-itemcomponent/>
                    <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
                <dmn-itemdefinition-info>
                    <itemdefinition-id>_7B3023E2-BC44-4BF3-BF7E-
773C240FB9AD</itemdefinition-id>
                    <itemdefinition-name>State</itemdefinition-name>
                    <itemdefinition-typeref>
                        <namespace-uri>https://kiegroup.org/dmn/_60B01F4D-E407-43F7-
848E-258723B5FAC8</namespace-uri>
                        <local-part>string</local-part>
                        <prefix></prefix>
                    </itemdefinition-typeref>
                    <itemdefinition-itemcomponent/>
                    <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
                <dmn-itemdefinition-info>
                    <itemdefinition-id>_3D4B49DD-700C-4925-99A7-
3B2B873F7800</itemdefinition-id>
                    <itemdefinition-name>city</itemdefinition-name>
                    <itemdefinition-typeref>
                        <namespace-uri>https://kiegroup.org/dmn/_60B01F4D-E407-43F7-
848E-258723B5FAC8</namespace-uri>
                        <local-part>string</local-part>
                        <prefix></prefix>
                    </itemdefinition-typeref>
                    <itemdefinition-itemcomponent/>
                    <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
                <dmn-itemdefinition-info>
                    <itemdefinition-id>_B37C49E8-B0D9-4B20-9DC6-
D655BB1CA7B1</itemdefinition-id>
                    <itemdefinition-name>Points</itemdefinition-name>
                    <itemdefinition-typeref>
                        <namespace-uri>https://kiegroup.org/dmn/_60B01F4D-E407-43F7-
848E-258723B5FAC8</namespace-uri>
                        <local-part>number</local-part>
```

```
                        <prefix></prefix>
                    </itemdefinition-typeref>
                    <itemdefinition-itemcomponent/>
                    <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
            </itemdefinition-itemcomponent>
            <itemdefinition-iscollection>false</itemdefinition-iscollection>
        </dmn-itemdefinition-info>
        <dmn-itemdefinition-info>
            <itemdefinition-id>_A4077C7E-B57A-4DEE-9C65-
7769636316F3</itemdefinition-id>
            <itemdefinition-name>tFine</itemdefinition-name>
            <itemdefinition-itemcomponent>
                <dmn-itemdefinition-info>
                    <itemdefinition-id>_79B152A8-DE83-4001-B88B-
52DFF0D73B2D</itemdefinition-id>
                    <itemdefinition-name>Amount</itemdefinition-name>
                    <itemdefinition-typeref>
                        <namespace-uri>https://kiegroup.org/dmn/_60B01F4D-E407-43F7-
848E-258723B5FAC8</namespace-uri>
                        <local-part>number</local-part>
                        <prefix></prefix>
                    </itemdefinition-typeref>
                    <itemdefinition-itemcomponent/>
                    <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
                <dmn-itemdefinition-info>
                    <itemdefinition-id>_D7CB5F9C-9D55-48C2-83EE-
D47045EC90D0</itemdefinition-id>
                    <itemdefinition-name>Points</itemdefinition-name>
                    <itemdefinition-typeref>
                        <namespace-uri>https://kiegroup.org/dmn/_60B01F4D-E407-43F7-
848E-258723B5FAC8</namespace-uri>
                        <local-part>number</local-part>
                        <prefix></prefix>
                    </itemdefinition-typeref>
                    <itemdefinition-itemcomponent/>
                    <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
            </itemdefinition-itemcomponent>
            <itemdefinition-iscollection>false</itemdefinition-iscollection>
        </dmn-itemdefinition-info>
    </itemdefinitions>
    <decisionservices/>
  </model>
 </dmn-model-info-list>
</response>
```

JSON 输出示例：

```
{
  "type" : "SUCCESS",
  "msg" : "OK models successfully retrieved from container 'Traffic-Violation_1.0.0-
SNAPSHOT'",
```

```
 "result" : {
  "dmn-model-info-list" : {
   "models" : [ {
    "model-namespace" : "https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8",
    "model-name" : "Traffic Violation",
    "model-id" : "_2CD7D1AA-BD84-4B43-AD21-B0342ADE655A",
    "decisions" : [ {
     "decision-id" : "_23428EE8-DC8B-4067-8E67-9D7C53EC975F",
     "decision-name" : "Fine"
    }, {
     "decision-id" : "_B5EEE2B1-915C-44DC-BE43-C244DC066FD8",
     "decision-name" : "Should the driver be suspended?"
    } ],
    "inputs" : [ {
     "inputdata-id" : "_CEB959CD-3638-4A87-93BA-03CD0FB63AE3",
     "inputdata-name" : "Violation",
     "inputdata-typeRef" : {
      "namespace-uri" : "https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8",
      "local-part" : "tViolation",
      "prefix" : ""
     }
    }, {
     "inputdata-id" : "_B0E810E6-7596-430A-B5CF-67CE16863B6C",
     "inputdata-name" : "Driver",
     "inputdata-typeRef" : {
      "namespace-uri" : "https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8",
      "local-part" : "tDriver",
      "prefix" : ""
     }
    } ],
    "itemDefinitions" : [ {
     "itemdefinition-id" : "_13C7EFD8-B85C-43BF-94D3-14FABE39A4A0",
     "itemdefinition-name" : "tDriver",
     "itemdefinition-typeRef" : null,
     "itemdefinition-itemComponent" : [ {
      "itemdefinition-id" : "_EC11744C-4160-4549-9610-2C757F40DFE8",
      "itemdefinition-name" : "Name",
      "itemdefinition-typeRef" : {
       "namespace-uri" : "https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8",
       "local-part" : "string",
       "prefix" : ""
      },
      "itemdefinition-itemComponent" : [ ],
      "itemdefinition-isCollection" : false
     }, {
      "itemdefinition-id" : "_E95BE3DB-4A51-4658-A166-02493EAAC9D2",
      "itemdefinition-name" : "Age",
      "itemdefinition-typeRef" : {
       "namespace-uri" : "https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8",
       "local-part" : "number",
       "prefix" : ""
```

```
        },
        "itemdefinition-itemComponent" : [ ],
        "itemdefinition-isCollection" : false
      }, {
        "itemdefinition-id" : "_7B3023E2-BC44-4BF3-BF7E-773C240FB9AD",
        "itemdefinition-name" : "State",
        "itemdefinition-typeRef" : {
          "namespace-uri" : "https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8",
          "local-part" : "string",
          "prefix" : ""
        },
        "itemdefinition-itemComponent" : [ ],
        "itemdefinition-isCollection" : false
      }, {
        "itemdefinition-id" : "_3D4B49DD-700C-4925-99A7-3B2B873F7800",
        "itemdefinition-name" : "City",
        "itemdefinition-typeRef" : {
          "namespace-uri" : "https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8",
          "local-part" : "string",
          "prefix" : ""
        },
        "itemdefinition-itemComponent" : [ ],
        "itemdefinition-isCollection" : false
      }, {
        "itemdefinition-id" : "_B37C49E8-B0D9-4B20-9DC6-D655BB1CA7B1",
        "itemdefinition-name" : "Points",
        "itemdefinition-typeRef" : {
          "namespace-uri" : "https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8",
          "local-part" : "number",
          "prefix" : ""
        },
        "itemdefinition-itemComponent" : [ ],
        "itemdefinition-isCollection" : false
      } ],
      "itemdefinition-isCollection" : false
    }, {
      "itemdefinition-id" : "_A4077C7E-B57A-4DEE-9C65-7769636316F3",
      "itemdefinition-name" : "tFine",
      "itemdefinition-typeRef" : null,
      "itemdefinition-itemComponent" : [ {
        "itemdefinition-id" : "_79B152A8-DE83-4001-B88B-52DFF0D73B2D",
        "itemdefinition-name" : "Amount",
        "itemdefinition-typeRef" : {
          "namespace-uri" : "https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8",
          "local-part" : "number",
          "prefix" : ""
        },
        "itemdefinition-itemComponent" : [ ],
        "itemdefinition-isCollection" : false
      }, {
        "itemdefinition-id" : "_D7CB5F9C-9D55-48C2-83EE-D47045EC90D0",
        "itemdefinition-name" : "Points",
```

```
          "itemdefinition-typeRef" : {
            "namespace-uri" : "https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8",
            "local-part" : "number",
            "prefix" : ""
          },
          "itemdefinition-itemComponent" : [ ],
          "itemdefinition-isCollection" : false
        } ],
        "itemdefinition-isCollection" : false
      }, {
        "itemdefinition-id" : "_9C758F4A-7D72-4D0F-B63F-2F5B8405980E",
        "itemdefinition-name" : "tViolation",
        "itemdefinition-typeRef" : null,
        "itemdefinition-itemComponent" : [ {
          "itemdefinition-id" : "_0B6FF1E2-ACE9-4FB3-876B-5BB30B88009B",
          "itemdefinition-name" : "Code",
          "itemdefinition-typeRef" : {
            "namespace-uri" : "https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8",
            "local-part" : "string",
            "prefix" : ""
          },
          "itemdefinition-itemComponent" : [ ],
          "itemdefinition-isCollection" : false
        }, {
          "itemdefinition-id" : "_27A5DA18-3CA7-4C06-81B7-CF7F2F050E29",
          "itemdefinition-name" : "Date",
          "itemdefinition-typeRef" : {
            "namespace-uri" : "https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8",
            "local-part" : "date",
            "prefix" : ""
          },
          "itemdefinition-itemComponent" : [ ],
          "itemdefinition-isCollection" : false
        }, {
          "itemdefinition-id" : "_8961969A-8A80-4F12-B568-346920C0F038",
          "itemdefinition-name" : "Type",
          "itemdefinition-typeRef" : {
            "namespace-uri" : "https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8",
            "local-part" : "string",
            "prefix" : ""
          },
          "itemdefinition-itemComponent" : [ ],
          "itemdefinition-isCollection" : false
        }, {
          "itemdefinition-id" : "_7450F12A-3E95-4D5E-8DCE-2CB1FAC2BDD4",
          "itemdefinition-name" : "Speed Limit",
          "itemdefinition-typeRef" : {
            "namespace-uri" : "https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8",
            "local-part" : "number",
            "prefix" : ""
          },
```

```
          "itemdefinition-itemComponent" : [ ],
          "itemdefinition-isCollection" : false
        }, {
          "itemdefinition-id" : "_0A9A6F26-6C14-414D-A9BF-765E5850429A",
          "itemdefinition-name" : "Actual Speed",
          "itemdefinition-typeRef" : {
            "namespace-uri" : "https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8",
            "local-part" : "number",
            "prefix" : ""
          },
          "itemdefinition-itemComponent" : [ ],
          "itemdefinition-isCollection" : false
        } ],
        "itemdefinition-isCollection" : false
      } ],
      "decisionServices" : [ ]
    } ]
  }
 }
}
```

5. 执行模型：

   [POST] server/containers/{containerId}/dmn

注意

属性 **model-namespace** 会自动生成，每个用户都是不同的。确保使用的 **model-namespace** 和 **model-name** 属性与部署的模型匹配。

curl 请求示例：

```
curl -u wbadmin:wbadmin -H "accept: application/json" -H "content-type: application/json" -X
POST "http://localhost:8080/kie-server/services/rest/server/containers/traffic-violation_1.0.0-
SNAPSHOT/dmn" -d "{ \"model-namespace\" : \"https://kiegroup.org/dmn/_60B01F4D-E407-
43F7-848E-258723B5FAC8\", \"model-name\" : \"Traffic Violation\", \"dmn-context\" :
{\"Driver\" : {\"Points\" : 15}, \"Violation\" : {\"Type\" : \"speed\", \"Actual Speed\" : 135, \"Speed
Limit\" : 100}}}"
```

JSON 请求示例：

```
{
  "model-namespace" : "https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8",
```

```
 "model-name" : "Traffic Violation",
 "dmn-context" :
 {
  "Driver" :
  {
   "Points" : 15
  },
 "Violation" :
  {
  "Type" : "speed",
  "Actual Speed" : 135,
  "Speed Limit" : 100
 }
 }
}
```

XML 请求示例（JAXB 格式）：

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dmn-evaluation-context>
   <dmn-context xsi:type="jaxbListWrapper" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
      <type>MAP</type>
      <element xsi:type="jaxbStringObjectPair" key="Violation">
         <value xsi:type="jaxbListWrapper">
            <type>MAP</type>
            <element xsi:type="jaxbStringObjectPair" key="Type">
               <value xsi:type="xs:string"
xmlns:xs="http://www.w3.org/2001/XMLSchema">speed</value>
            </element>
            <element xsi:type="jaxbStringObjectPair" key="Speed Limit">
               <value xsi:type="xs:decimal"
xmlns:xs="http://www.w3.org/2001/XMLSchema">100</value>
            </element>
            <element xsi:type="jaxbStringObjectPair" key="Actual Speed">
               <value xsi:type="xs:decimal"
xmlns:xs="http://www.w3.org/2001/XMLSchema">135</value>
            </element>
         </value>
      </element>
      <element xsi:type="jaxbStringObjectPair" key="Driver">
         <value xsi:type="jaxbListWrapper">
            <type>MAP</type>
            <element xsi:type="jaxbStringObjectPair" key="Points">
               <value xsi:type="xs:decimal"
xmlns:xs="http://www.w3.org/2001/XMLSchema">15</value>
            </element>
         </value>
      </element>
   </dmn-context>
</dmn-evaluation-context>
```

注意

无论请求格式如何，请求都需要以下元素：

- 型号命名空间
- 型号名称
- 包含输入值的上下文对象

JSON 响应示例：

```
{
  "type": "SUCCESS",
  "msg": "OK from container 'Traffic-Violation_1.0.0-SNAPSHOT'",
  "result": {
     "dmn-evaluation-result": {
        "messages": [],
        "model-namespace": "https://kiegroup.org/dmn/_7D8116DE-ADF5-4560-A116-
FE1A2EAFFF48",
        "model-name": "Traffic Violation",
        "decision-name": [],
        "dmn-context": {
           "Violation": {
             "Type": "speed",
             "Speed Limit": 100,
             "Actual Speed": 135
           },
           "Should Driver be Suspended?": "Yes",
             "Driver": {
               "Points": 15
             },
             "Fine": {
               "Points": 7,
               "Amount": 1000
             }
         },
     "decision-results": {
        "_E1AF5AC2-E259-455C-96E4-596E30D3BC86": {
           "messages": [],
           "decision-id": "_E1AF5AC2-E259-455C-96E4-596E30D3BC86",
           "decision-name": "Should the Driver be Suspended?",
           "result": "Yes",
           "status": "SUCCEEDED"
         },
        "_D7F02CE0-AF50-4505-AB80-C7D6DE257920": {
```

```
          "messages": [],
          "decision-id": "_D7F02CE0-AF50-4505-AB80-C7D6DE257920",
          "decision-name": "Fine",
          "result": {
            "Points": 7,
            "Amount": 1000
          },
        "status": "SUCCEEDED"
      }
    }
  }
 }
}
```

XML（JAXB 格式）响应示例：

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response type="SUCCESS" msg="OK from container 'Traffic_1.0.0-SNAPSHOT'">
   <dmn-evaluation-result>
      <model-namespace>https://kiegroup.org/dmn/_A4BCA8B8-CF08-433F-93B2-
A2598F19ECFF</model-namespace>
      <model-name>Traffic Violation</model-name>
      <dmn-context xsi:type="jaxbListWrapper"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
         <type>MAP</type>
         <element xsi:type="jaxbStringObjectPair" key="Violation">
            <value xsi:type="jaxbListWrapper">
               <type>MAP</type>
               <element xsi:type="jaxbStringObjectPair" key="Type">
                  <value xsi:type="xs:string"
xmlns:xs="http://www.w3.org/2001/XMLSchema">speed</value>
               </element>
               <element xsi:type="jaxbStringObjectPair" key="Speed Limit">
                  <value xsi:type="xs:decimal"
xmlns:xs="http://www.w3.org/2001/XMLSchema">100</value>
               </element>
               <element xsi:type="jaxbStringObjectPair" key="Actual Speed">
                  <value xsi:type="xs:decimal"
xmlns:xs="http://www.w3.org/2001/XMLSchema">135</value>
               </element>
            </value>
         </element>
         <element xsi:type="jaxbStringObjectPair" key="Driver">
            <value xsi:type="jaxbListWrapper">
               <type>MAP</type>
               <element xsi:type="jaxbStringObjectPair" key="Points">
                  <value xsi:type="xs:decimal"
xmlns:xs="http://www.w3.org/2001/XMLSchema">15</value>
               </element>
            </value>
         </element>
         <element xsi:type="jaxbStringObjectPair" key="Fine">
            <value xsi:type="jaxbListWrapper">
               <type>MAP</type>
```

```
                <element xsi:type="jaxbStringObjectPair" key="Points">
                  <value xsi:type="xs:decimal"
xmlns:xs="http://www.w3.org/2001/XMLSchema">7</value>
                </element>
                <element xsi:type="jaxbStringObjectPair" key="Amount">
                  <value xsi:type="xs:decimal"
xmlns:xs="http://www.w3.org/2001/XMLSchema">1000</value>
                </element>
              </value>
            </element>
            <element xsi:type="jaxbStringObjectPair" key="Should the driver be suspended?">
              <value xsi:type="xs:string"
xmlns:xs="http://www.w3.org/2001/XMLSchema">Yes</value>
            </element>
          </dmn-context>
          <messages/>
          <decisionResults>
            <entry>
              <key>_4055D956-1C47-479C-B3F4-BAEB61F1C929</key>
              <value>
                <decision-id>_4055D956-1C47-479C-B3F4-BAEB61F1C929</decision-id>
                <decision-name>Fine</decision-name>
                <result xsi:type="jaxbListWrapper"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                  <type>MAP</type>
                  <element xsi:type="jaxbStringObjectPair" key="Points">
                    <value xsi:type="xs:decimal"
xmlns:xs="http://www.w3.org/2001/XMLSchema">7</value>
                  </element>
                  <element xsi:type="jaxbStringObjectPair" key="Amount">
                    <value xsi:type="xs:decimal"
xmlns:xs="http://www.w3.org/2001/XMLSchema">1000</value>
                  </element>
                </result>
                <messages/>
                <status>SUCCEEDED</status>
              </value>
            </entry>
            <entry>
              <key>_8A408366-D8E9-4626-ABF3-5F69AA01F880</key>
              <value>
                <decision-id>_8A408366-D8E9-4626-ABF3-5F69AA01F880</decision-id>
                <decision-name>Should the driver be suspended?</decision-name>
                <result xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">Yes</result>
                <messages/>
                <status>SUCCEEDED</status>
              </value>
            </entry>
          </decisionResults>
        </dmn-evaluation-result>
</response>
```

第 8 章 其他资源

- *使用 DMN 模型设计决策服务*
- *使用测试场景测试决策服务*
- *在 Business Central 中管理项目*
- *使用 KIE API 与 Red Hat Process Automation Manager 交互*

部分 II. 在 RED HAT PROCESS AUTOMATION MANAGER 中使用进程服务

作为业务规则和流程开发人员，您可以在红帽流程自动化管理器或 VS Code 中的 Red Hat Process Automation Manager BPMN 模型器中使用 Business Central 来设计业务流程以满足特定的业务要求。Red Hat Process Automation Manager 提供 Business Central 中的示例项目，其中包含用于参考的业务资产。本文档论述了如何创建新的影片流程项目、数据对象和业务流程，以熟悉 Business Central 和流程设计人员。

然后，您将参考 Business Central 中包含的 Mortgage_Process 示例项目，以检查项目的业务规则、决策表和表单。您将构建和部署 Mortgage_Process 示例项目，并执行项目的定义功能。

先决条件

- Red Hat JBoss Enterprise Application Platform 7.4 已安装。详情请查看 *Red Hat JBoss Enterprise Application Platform 7.4 安装指南*。
- Red Hat Process Automation Manager 由 KIE 服务器安装和配置。如需更多信息，*请参阅在 Red Hat JBoss EAP 7.4 上安装和配置 Red Hat Process Automation Manager*。
- Red Hat Process Automation Manager 正在运行，您可以使用 开发人员 角色登录到 Business Central。如需更多信息，请参阅 *规划 Red Hat Process Automation Manager 安装*。

第 9 章 概述

Business Central 使您能够自动化您的业务流程。业务流程是一种图，描述必须执行一系列步骤的顺序，并且由预定义的节点和连接组成。每个节点代表进程中的一个步骤，同时连接指定如何从一个节点转换到另一个节点。

例如，银行提供大片金服务。使用 Business Central，银行住住部门为抵押贷款创建了完整的业务流程。

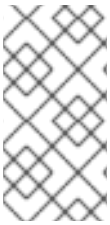
当客户想要使用学者购买新属性时，将采取以下步骤：

1. 客户联系该代理商，该代理协助转销贷款。
2. 该代理收集有关属性和客户的信息，如客户、社交安全号码、属性销售价格和所请求金金。
3. 然后，代理代表客户提交请求。

每当客户提交请求时，都会创建一个新进程实例。这样可保证评估每个请求的质量一致性，并对每个请求的状态提供全面的可见性，并使进程高效运行。

第 10 章 BUSINESS CENTRAL 中的项目和业务资产示例

Business Central 包含具有业务资产的示例项目，您可将其用作您在您自己的 Red Hat Process Automation Manager 项目中所创建的规则、流程或其他资产的参考。每个示例项目都有所不同，旨在展示红帽流程自动化、决策管理或业务优化资产和逻辑。



注意

红帽不提供对 Red Hat Process Automation Manager 发行版本中包含的示例代码的支持。

Business Central 中提供了以下示例项目：

- courses_Scheduling:(Business optimization)课程调度和课程决策流程。为房间分配演讲，并根据课程冲突和课程空间容量等因素确定学员的课程。
- Dinner_ 第三方 ：（业务优化）客户使用指导决策表进行优化。根据每个 guest 的作业类型、自治关系和已知关系来分配客户机名额。
- Employee_Rostering:（业务优化）利用决策和解决人员资产提高优化。为员工分配基于技能的转变。
- 评估_流程 ：（流程自动化）使用业务流程资产评估流程。根据性能评估员工。
- IT_Orders ：（流程自动化和案例管理）使用业务流程和问题单定义案例。根据需求和批准放置 IT 硬件顺序。
- mortgages:（规则管理） Loan 批准过程，使用基于规则的决策资产。根据相关数据和资格确定金级资格。
- Mortgage_Process:（流程自动化）使用业务流程和决策资产进行循环审批流程。根据相关数据和资格确定金级资格。
- OptaCloud ：（业务优化）使用决策和解决资产进行资源分配优化。为具有有限资源的计算

机分配进程。

- traffic_Violation ：（使用 DMN 进行 Decision 管理）流量违反决策服务，使用 Decision Model 和 Notation(DMN)模型。根据流量违反情况确定驱动程序 penalty 和 suspension。

10.1. 访问 BUSINESS CENTRAL 中的示例项目和业务资产

您可以使用 Business Central 中的示例项目探索业务资产，作为您在您自己的 Red Hat Process Automation Manager 项目中创建的规则或其他资产的引用。

先决条件

- 业务中心已安装且正在运行。有关安装选项，请参阅 *规划 Red Hat Process Automation Manager 安装*。

流程

1. 在 Business Central 中，转至 Menu → Design → Projects。如果已有项目，您可以点击 MySpace default 空间并从 Add Project 下拉菜单中选择 Try Samples 来访问示例。如果没有现有项目，请点击 Try samples。

2. 查看每个示例项目的描述，以确定您要浏览的项目。每个示例项目都有所不同，旨在展示红帽流程自动化、决策管理或业务优化资产和逻辑。

3. 选择一个或多个示例项目，并点击 Ok 将项目添加到您的空间中。

4. 在空格的 Projects 页面中，选择一个示例项目来查看该项目的资产。

5. 选择每个资产来探索项目如何设计实现指定目标或工作流。某些示例项目包含多个资产页面。点击右上角的左或右箭头查看完整的资产列表。

图 10.1. 资产页面选择

6. 在项目 资产 页面右上角，单击 Build 来构建示例项目或 Deploy 以构建项目，然后将其部署到 KIE 服务器。

注意

您还可以选择 **Build & Install** 选项来构建项目，并将 **KJAR** 文件发布到配置的 **Maven** 存储库，而无需部署到 **KIE** 服务器。在开发环境中，您可以点击 **Deploy** 将构建的 **KJAR** 文件部署到 **KIE** 服务器，而无需停止任何正在运行的实例（如果适用），或者点击 **Redeploy** 来部署构建的 **KJAR** 文件并替换所有实例。下次部署或重新部署构建的 **KJAR** 时，以前的部署单元（**KIE** 容器）会在同一目标 **KIE** 服务器中自动更新。在生产环境中，**Redeploy** 选项被禁用，且只能点击 **Deploy** 将构建的 **KJAR** 文件部署到 **KIE** 服务器上的新部署单元（**KIE** 容器）。

要配置 **KIE** 服务器环境模式，请将 **org.kie.server.mode** 系统属性设置为 **org.kie.server.mode=development** 或 **org.kie.server.mode=production**。要在 **Business Central** 中配置对应项目的部署行为，请转至 **Project Settings → General Settings → Version**，切换 **Development Mode** 选项，然后点 **Save**。默认情况下，**Business Central** 中的 **KIE** 服务器和所有新项目均为开发模式。您不能部署打开开发模式的项目，或使用手动将 **SNAPSHOT** 版本后缀添加到生产模式中的 **KIE** 服务器。

要查看项目部署详情，可在屏幕顶部的部署横幅或 **Deploy** 下拉菜单中点击 **View deployment details**。这个选项将您定向到 **Menu → Deploy → Execution Servers** 页面。

第 11 章 RED HAT PROCESS AUTOMATION MANAGER BPMN 和 DMN 型号器

Red Hat Process Automation Manager 提供了以下扩展程序或应用程序，您可以使用它们设计业务流程模型和符号(BPMN)流程模型，以及使用图形模型(DMN)决策模型。

- Business Central ：使您能够在相关嵌入式设计人员中查看和设计 BPMN 模型、DMN 模型和测试方案文件。

  要使用 Business Central，您可以设置一个包含 Business Central 的开发环境，用于设计业务规则和流程，以及 KIE 服务器来执行和测试创建的业务规则和流程。

- Red Hat Process Automation Manager VS Code 扩展 ：允许您在 Visual Studio Code(VS Code)中查看和设计 BPMN 模型、DMN 模型和测试场景文件。VS Code 扩展需要 VS Code 1.46.0 或更新版本。

  要安装 Red Hat Process Automation Manager VS Code 扩展，请在 VS Code 中选择 Extensions 菜单选项，并搜索并安装 Red Hat Business Automation Bundle 扩展。

- 独立 BPMN 和 DMN 编辑器 ：使您能够查看和设计嵌入在 web 应用程序中的 BPMN 和 DMN 模型。要下载所需的文件，您可以使用 NPM registry 中的 NPM 工件，或直接下载位于 https://<YOUR_PAGE>/dmn/index.js 的 DMN 独立编辑器库的 JavaScript 文件。

11.1. 安装 RED HAT PROCESS AUTOMATION MANAGER VS CODE 扩展捆绑包

Red Hat Process Automation Manager 提供了一个 Red Hat Network Automation Bundle VS Code 扩展，它可让您设计决策模型和符号(DMN)决策模型、业务流程模型和符号(BPMN)2.0 业务流程并直接在 VS Code 中测试场景。VS Code 是开发新业务应用的首选集成开发环境(IDE)。Red Hat Process Automation Manager 还提供单独的 DMN Editor 和 BPMN Editor VS Code 扩展（如果需要）。

重要

VS Code 中的编辑器部分与 Business Central 中的编辑器兼容，并且 VS Code 不支持几个 Business Central 功能。

先决条件

- 安装了 VS Code 的最新稳定版本。

流程

1. 在 VS Code IDE 中，选择 Extensions 菜单选项，并搜索 Red Hat Business Automation Bundle for DMN、Harllse 和 test scenario 文件支持。

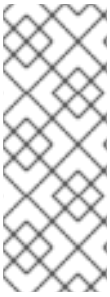
   对于 DMN 或 BPMN 文件支持，您还可以搜索单独的 DMN Editor 或 BPMN Editor 扩展。

2. 当 Red Hat Business Automation Bundle 扩展出现在 VS Code 中时，选择它并点 Install。

3. 要获得最佳 VS Code 编辑器行为，请在扩展安装完成后，重新加载或关闭并重新启动 VS Code 实例。

安装 VS Code 扩展捆绑包后，任何 .dmn、.bpmn 或 .bpmn2 文件都会自动显示为图形模型。此外，您打开或创建的 .scesim 文件自动显示为表格测试场景模型，用于测试您的业务决策功能。

如果 DMN、CEP 或测试场景模型器只打开 DMN、Hardb 或 test scenario 文件的 XML 源，并显示错误消息，请查看报告的错误和模型文件，以确保定义所有元素。



注意

对于新的 DMN 或 BPMN 模型，您还可以在网页浏览器中输入 dmn.new 或 BPMn.new，以在在线模型程序中设计 DMN 或 BPMN 模型。完成创建模型后，您可以点击 Download in the online modeler 页面将 DMN 或 BPMN 文件导入到 VS Code 中的 Red Hat Process Automation Manager 项目中。

11.2. 配置 RED HAT PROCESS AUTOMATION MANAGER 独立编辑器

Red Hat Process Automation Manager 提供独立编辑器，这些编辑器在自包含的库中分发，为每个编辑器提供一个一体化 JavaScript 文件。JavaScript 文件使用全面的 API 来设置和控制编辑器。

您可以使用以下方法安装独立编辑器：

- 手动下载每个 JavaScript 文件

- 使用 NPM 软件包

流程

1. 使用以下方法之一安装独立编辑器：

   手动下载每个 JavaScript 文件 ： 对于这个方法，请按照以下步骤操作：

   a. 下载 JavaScript 文件。

   b. 将下载的 Javascript 文件添加到您的托管应用程序中。

   c. 将以下 &lt ;script& gt; 标签添加到 HTML 页面：

   DMN 编辑器的 HTML 页面标记

   ```
   <script src="https://<YOUR_PAGE>/dmn/index.js"></script>
   ```

   BPMN 编辑器的 HTML 页面的脚本标签

   ```
   <script src="https://<YOUR_PAGE>/bpmn/index.js"></script>
   ```

   使用 NPM 软件包 ： 对于这个方法，请按照以下步骤操作：

   a. 在 package.json 文件中添加 NPM 软件包：

   添加 NPM 软件包

```
npm install @kie-tools/kie-editors-standalone
```

b. **将每个编辑器库导入到 TypeScript 文件：**

导入每个编辑器

```
import * as DmnEditor from "@kie-tools/kie-editors-standalone/dist/dmn"
import * as BpmnEditor from "@kie-tools/kie-editors-standalone/dist/bpmn"
```

2. **安装独立编辑器后，使用提供的编辑器 API 打开所需的编辑器，如下例所示，打开 DMN 编辑器。每个编辑器的 API 相同。**

打开 DMN 独立编辑器

```
const editor = DmnEditor.open({
  container: document.getElementById("dmn-editor-container"),
  initialContent: Promise.resolve(""),
  readOnly: false,
  origin: "",
  resources: new Map([
    [
      "MyIncludedModel.dmn",
      {
        contentType: "text",
        content: Promise.resolve("")
      }
    ]
  ])
});
```

在 editor API 中使用以下参数：

表 11.1. 示例参数

参数	描述
container	附加编辑器的 HTML 元素。
initialContent	对 DMN 模型内容的承诺。这个参数可以为空，如下例所示： • **Promise.resolve("")** • **Promise.resolve("<DIAGRAM_CONTENT_DIRECTLY_HERE>")** • **fetch("MyDmnModel.dmn").then(content ⇒ content.text())**
ReadOnly（可选）	可让您在编辑器中允许更改。在编辑器中将 设置为 **false**（默认）以允许内容编辑和 **true**。
Origin（可选）	仓库的起源。默认值为 **window.location.origin**。
资源（可选）	编辑器的资源映射。例如，这个参数用于为 DMN 编辑器提供包含的模型，或为 BPMN 编辑器提供工作项目定义。映射中的每个条目都包含一个资源名称和一个对象，它由 **content -type**（**文本 或二进制**）**和内容**组成（与 **初始 Content** 参数类似）。

返回的对象包含操作编辑器所需的方法。

表 11.2. 返回的对象方法

方法	描述
getContent（）：Promise<string>	返回包含编辑器内容的保证。
setContent(path: string, content: string): void	设置编辑器的内容。
getPreview（）：Promise<string>	返回包含当前图表的 SVG 字符串的保证。
subscribeToContentChanges（callback:(isDirty: boolean)IFL void）:(isDirty: boolean)void	当编辑器中的内容更改并返回用于 unsubscription 的回调时，设置要调用的回调。

方法	描述
subscribe toContentChanges（callback: (isDirty: boolean)lFL void）: void	当内容在编辑器中更改时，取消订阅传递的回调。
markAsSaved(): void	重置编辑器状态，这表示已保存编辑器中的内容。另外，它会激活与内容更改相关的订阅回调。
undo（）: void	在编辑器中取消上次更改。另外，它会激活与内容更改相关的订阅回调。
redo（）: void	在编辑器中恢复最近一次撤消的更改。另外，它会激活与内容更改相关的订阅回调。
close（）: void	关闭编辑器。
getElementPosition(selector: string): Promise<Rect>	提供了一种替代方式，可以在可清空或视频组件中的元素中时扩展标准查询选择器。**选择器** 参数必须遵循 <targetNamespaces **>:::<SELECT** > 格式，如 **Canvas:::MySquare** 或 **Video:::PresenterHand**。此方法返回一个代表元素位置的 **Rect**。
envelopeApi: MessageBusClientApi<KogitoEditorEnvelopeApi>	这是高级编辑器 API。有关高级编辑器 API 的更多信息，请参阅 MessageBusClientApi 和 KogitoEditorEnvelopeApi。

第 12 章 使用 MAVEN 创建和执行 DMN 和 BPMN 模型

您可以使用 Maven archetypes 使用 Red Hat Process Automation Manager VS Code 扩展（而非 Business Central）在 VS Code 中开发 DMN 和 BPMN 模型。然后，您可以根据需要将您的 archetypes 与 Red Hat Process Automation Manager 决策和流程服务集成。这种开发 DMN 和 BPMN 模型的方法对使用 Red Hat Process Automation Manager VS Code 扩展构建新的业务应用程序会很有帮助。

流程

1. 在命令终端中，导航到用于存储新 Red Hat Process Automation Manager 项目的本地文件夹。

2. 输入以下命令使用 Maven archtype 在定义的文件夹中生成项目：

使用 Maven archetype 生成项目

```
mvn archetype:generate \
    -DarchetypeGroupId=org.kie \
    -DarchetypeArtifactId=kie-kjar-archetype \
    -DarchetypeVersion=7.67.0.Final-redhat-00024
```

此命令生成包含所需依赖项的 Maven 项目，并生成所需的目录和文件以构建您的业务应用程序。在开发项目时，您可以使用 Git 版本控制系统（推荐）。

如果要在同一目录中生成多个项目，请通过将 -DgroupId= -D artifactId ={> -DartifactId= <artifactId> 添加到 上一个命令来指定生成的业务应用程序的 artifactId 和 groupId。

3. 在 VS Code IDE 中，单击 File，选择 Open Folder，再导航到使用上一命令生成的文件夹。

4. 在创建第一个资产前，请为您的业务应用程序设置一个软件包，例如 org.kie.Businessapp，并在以下路径中创建相应的目录：

- PROJECT_HOME/src/main/java

- PROJECT_HOME/src/main/resources
- PROJECT_HOME/src/test/resources

例如，您可以为 org.kie . Businessapp 创建 PROJECT_HOME/src/main/java /org/kie/businessapp 软件包。

5. 使用 VS Code 为您的业务应用程序创建资产。您可以使用以下方法创建由 Red Hat Process Automation Manager VS Code 扩展支持的资产：

- 要创建业务流程，请在 PROJECT_HOME/src/main/resources/org/kie/businessapp 目录（如 Process .bpmn ）中使用 .bpmn 或 .bpmn 创建新文件。
- 要创建 DMN 模型，请在 PROJECT_HOME/src/main/resources/org/kie/businessapp 目录中创建一个使用 .dmn 的新文件，如 AgeDecision.dmn。
- 要创建测试场景模拟模型，请在 PROJECT_HOME/src/test/resources/org/kie/businessapp 目录中创建一个带有 .scesim 的新文件，如 TestAgeScenario.scesim。

6. 在 Maven archetype 中创建资产后，导航到命令行中项目的 root 目录（包含 pom.xml），再运行以下命令来构建项目的知识库文章(KJAR)：

```
mvn clean install
```

如果构建失败，请解决命令行错误消息中描述的任何问题，并尝试验证项目，直到构建成功为止。但是，如果构建成功，您可以在 PROJECT_HOME/target 目录中找到业务应用程序的工件。

注意

在开发的每一主要更改后，经常使用 mvn clean install 命令验证您的项目。

您可以使用 REST API 在运行的 KIE 服务器上部署您业务应用程序生成的知识 JAR(KJAR)。有关使用 REST API 的更多信息，请参阅使用 *KIE API 与 Red Hat Process Automation Manager 交互*。

第 13 章 创建用户

您可以根据需要创建多个 Business Central 用户。用户特权和设置由分配给用户的角色以及用户所属的组控制。在本示例中，您必须创建两个新用户：Kty，谁将担任银行的 loan manager 和 approver，而 Bill 人将充当请求 loans 的代理。有关创建用户的更多信息，请参阅在 Red Hat *JBoss EAP 7.4 上安装和配置 Red Hat Process Automation Manager 的* 创建用户 章节。

在 Business Central 中，您可以使用组和角色来控制用户集合的权限。您可以根据需要创建多个组和角色，但组必须至少有一个用户。

- 在本例中，处理任务的用户或用户必须分配给以下一个或多个组和角色：
  - approver 组： 用于 Qualify 任务
  - broker 组： 用于检测数据 并增加故障支付 任务
  - Manager 角色：对于 最后批准 任务

流程

1. 点击右上角的齿轮图标

并点 Users。

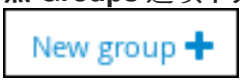
2. 点


，输入 Katy，点 Next，然后点 Create。

3. 点 Yes 设置密码，在这两个字段中输入 Katy，然后点 Change。

4. 输入 Bill，单击 Next，再单击 Create。

5. 单击 **Yes** 以设置密码，在这两个字段中输入 **Bill**，然后单击 **Change**。

6. 点 **Groups** 选项卡并点击 

，输入 **approver**，然后点 **Next**。

7. 从用户列表选择 **Katy**，点 **Add selected users**。

8. 输入 **broker**，然后单击 **Next**。

9. 从用户列表选择 **Bill**，然后单击 **Add selected users**。

10. 点 **Users** 选项卡，选择 **Katy**，然后点 **Edit** → **Roles** → **Add roles**。

11. 选择 **manager**，单击 **Add to selected roles**，然后单击 **Save**。

12. 点 **Groups** 标签页，点 **Edit** → **Groups** → **Add to groups**。

13. 选择 **批准 人和 kie-server**，然后单击 **Add to selected groups**。

14. 点击 **Save**。

15. 点 **Users** 选项卡，从用户列表选择 **Bill**，然后点击 **Edit** → **Roles** → **Add roles**。

16. 选择用户，然后单击 **Add to selected roles**。

17. 点 **Groups** 选项卡，点 

，选择 **kie-server**，然后点 **Add to selected groups**。

18. **点击 Save。**

第 14 章 创建 MORTGAGE-PROCESS 项目

项目是数据对象、业务流程、指导规则、决策表和表单等资产的容器。您创建的项目与 Business Central 中的现有 Mortgage_Process 示例项目类似。

流程

1. 在 Business Central 中，转至 Menu → Design → Projects。

   Red Hat Process Automation Manager 提供了一个名为 MySpace 的默认空间，如以下镜像所示。您可以使用默认空间来创建和测试示例项目。

   图 14.1. 默认空间

2. 单击 Add Project。

3. 在 Name 字段中输入 mortgage-process。

4. 点 Configure Advanced Options，使用以下值修改 GAV 字段：

   - 组 ID:com.myspace
   - 工件 ID:rtgage-process
   - Version:1.0.0

5. 点击 Add。

项目 的资源视图将打开。

第 15 章 创建 MORTGAGE-PROCESS DATA 对象

数据对象是您创建的规则资产的构建块。数据对象是在项目指定软件包中作为 Java 类实施的自定义数据类型。这些自定义数据类型决定了您的资产和您的决策服务所基于的数据。

mortgage 进程项目使用以下数据对象：

- applicant
- 属性
- ValidationErrorDO
- Application（应用程序）

15.1. 创建 APPLICANT 数据对象

创建 Applicant 数据对象，其中包含有关 applicant 的信息。这是适用于本教程中借款所需的基本信息。

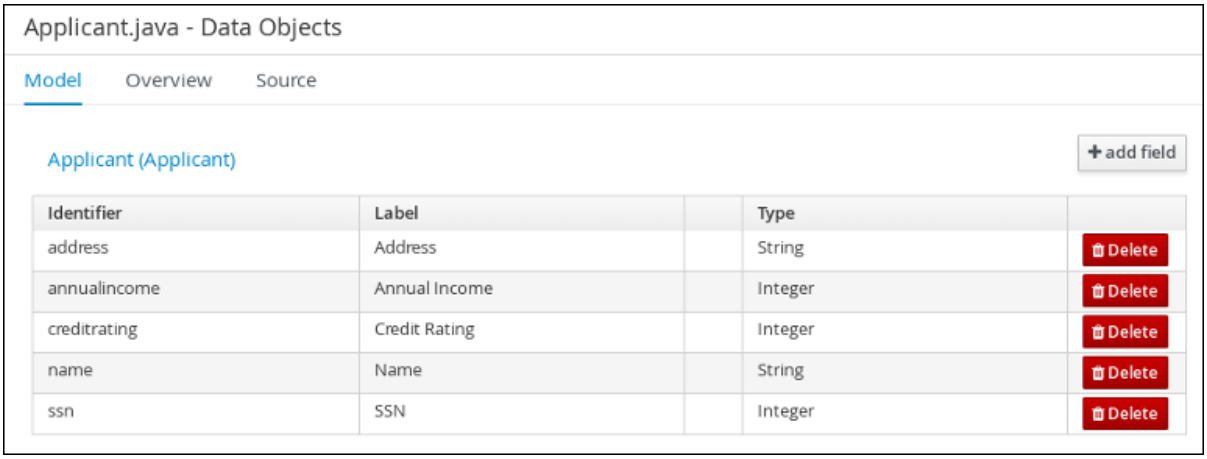
流程

1. 在 Business Central 中，单击 MySpace 默认空间。
2. 点 Menu → Design → Projects，然后点击 mortgage-process。
3. 单击 Add Asset 并选择 Data Object。
4. 在 Create new Data Object 窗口的 Data Object 字段中输入 Applicant。
5. 从 Package 下拉菜单中选择 com.myspace.mortgage_app，然后点击 Ok。

6. 在 'Applicant'- General properties 部分的 Label 字段中输入 Applica nt。

7. 点 +add 字段 并输入以下 Applicant 数据对象值。点 Create，并在 添加后继续。对于最后添加的，请点击 Create。

图 15.1. applicant 数据对象字段值



8. 点击 Save。

15.2. 创建 PROPERTY 数据对象

创建 Property 数据对象，其中包含有关属性详细信息的信息，如属性年龄和价格。

流程

1. 在 Business Central 中，单击 MySpace 默认空间。

2. 点 Menu → Design → Projects，然后点击 mortgage-process。

3. 单击 Add Asset 并选择 Data Object。

4. 在 Create new Data Object 窗口的 Data Object 字段中输入 Property。

5. 从 Package 下拉菜单中选择 com.myspace.mortgage_app，然后点击 Ok。

6. 在 ' Property '- General properties 部分的 Label 字段中输入 Property。

7. 点 +add 字段 并输入以下 Property data 对象值。点 Create，并在 添加后继续。对于最后添加的，请点击 Create。

图 15.2. 属性数据对象字段值

8. 点击 Save。

15.3. 创建 VALIDATIONERRORDO 数据对象

创建 ValidationErrorDO 数据对象，它指定应用程序错误的原因。

流程

1. 在 Business Central 中，单击 MySpace 默认空间。

2. 点 Menu → Design → Projects，然后点击 mortgage-process。

3. 单击 Add Asset 并选择 Data Object。

4. 在 Create new Data Object 窗口的 Data Object 字段中输入 ValidationErrorDO。

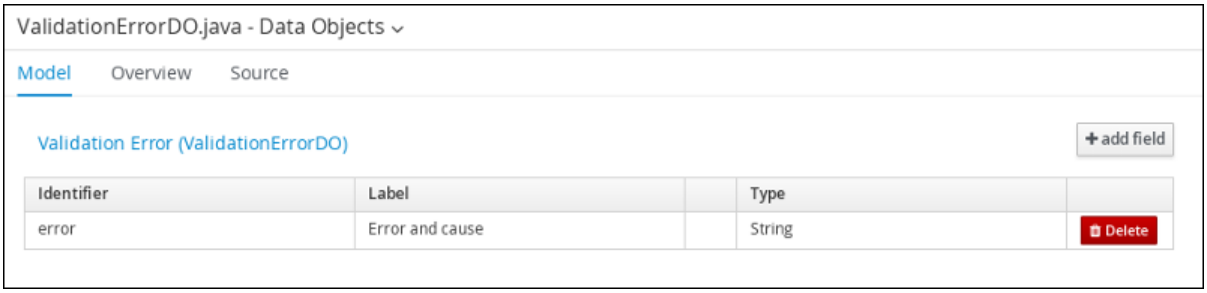
5. 从 Package 下拉菜单中选择 com.myspace.mortgage_app，然后点击 Ok。

6.

在 ' ValidationErrorDO '- General properties 项中的 Label 字段中输入 ValidationErrorDO。

7. 单击 +add 字段，并输入以下 ValidationErrorDO data 对象值。点 Create，并在 添加后继续。对于最后添加的，请点击 Create。

图 15.3. ValidationErrorDO data object field 值



8. 点击 Save。

15.4. 创建应用程序数据对象

创建 Application data 对象，其中包含有关 mortgage 详细信息的信息，如 down 支付和抵制数量。

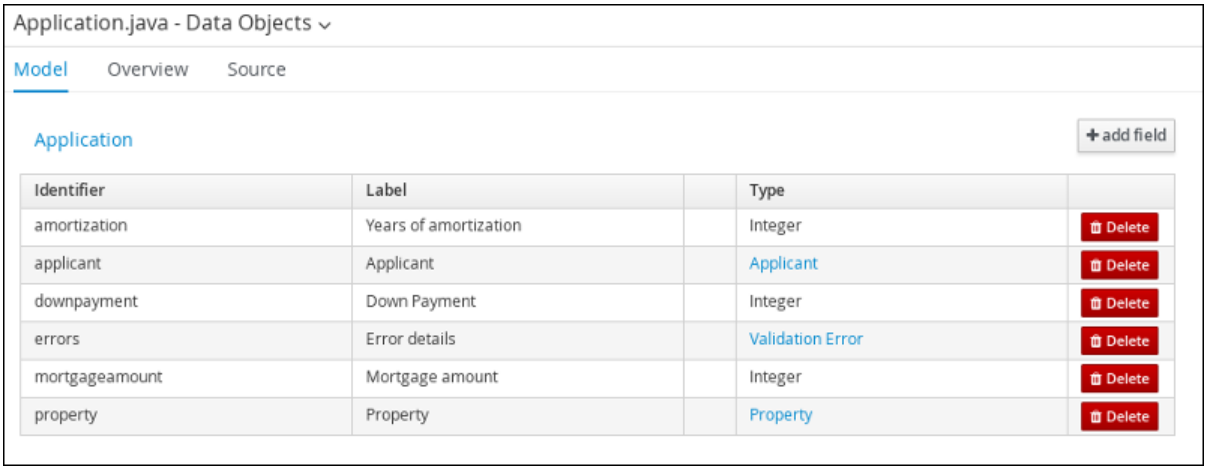
流程

1. 在 Business Central 中，单击 MySpace 默认空间。
2. 点 Menu → Design → Projects，然后点击 mortgage-process。
3. 单击 Add Asset 并选择 Data Object。
4. 在 Create new Data Object 窗口的 Data Object 字段中输入 Application。
5. 从 Package 下拉菜单中选择 com.myspace.mortgage_app，然后点击 Ok。
6. 在 ' Application '- General properties 部分的 Label 字段中输入 Application。

7.

点 +add 字段 并输入以下 Application data 对象值。点 Create，并在 添加后继续。对于最后添加的，请点击 Create。

图 15.4. 应用程序数据对象字段值



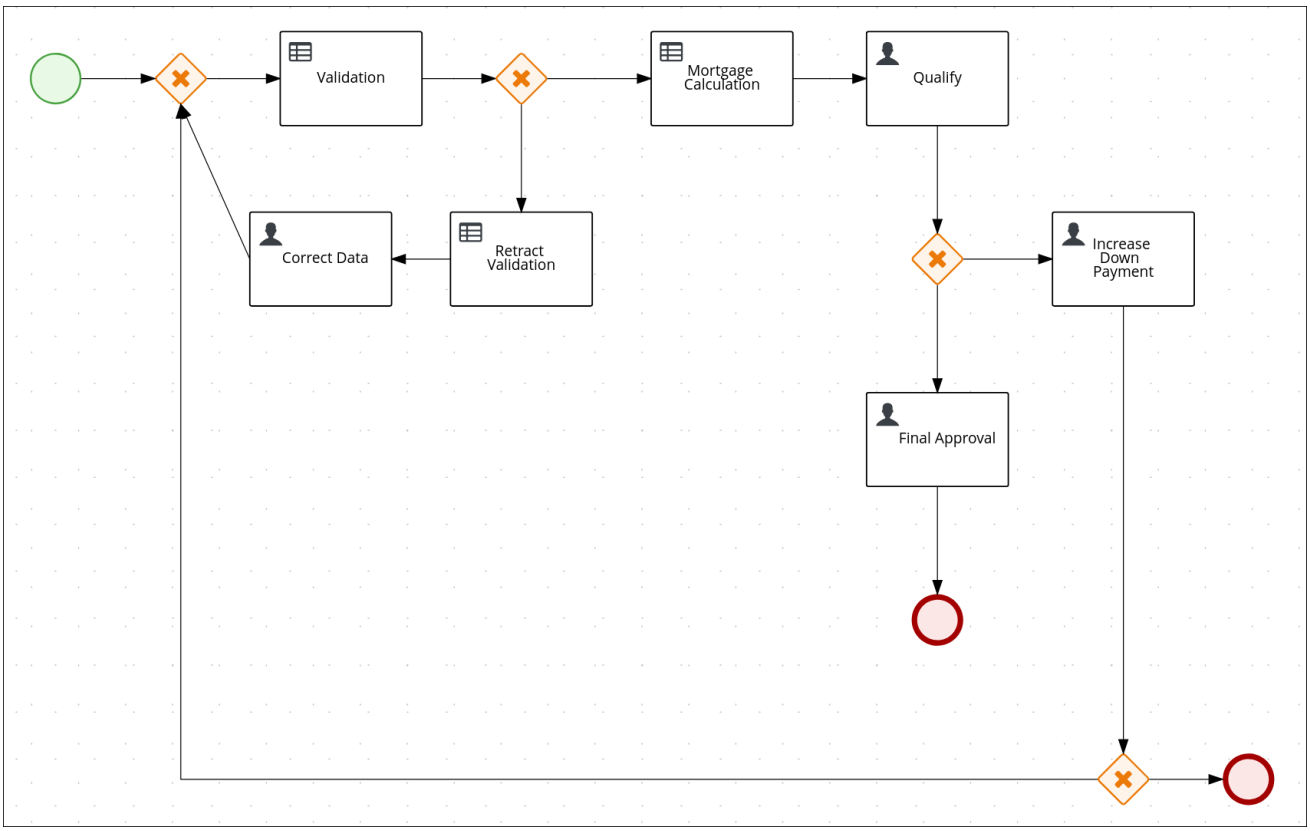
8.

点击 Save。

第 16 章 BUSINESS CENTRAL 中的业务流程

业务流程是一个图，描述必须使用流图表执行一系列步骤的顺序。业务流程包含一组节点，它们通过连接相互连接。每个节点代表整个过程中的一个步骤，而连接指定了如何从一个节点转换到另一个节点。

Mortgage_Process 示例包含以下预定义的 MortgageApprovalProcess 业务流程。



16.1. 创建业务流程

以下流程逐步指导您完成组成 MortgageApprovalProcess 业务流程的任务、连接和网关。mortgage 验证业务流程确定了分流应用程序是否包含所有必需数据。如果满足指定的数据要求，则应用程序将继续进入分流计算业务流程。

流程

1. 在 Business Central 中，前往 Menu → Design → Projects → Mortgage-Process。

2. 点 Add Asset → Business Process。

3. 输入以下值：

- 业务流程 ：MortgageApprovalProcess
- 软件包 ：选择 com.myspace.mortgage_app

软件包 指定创建资产的现有项目中的位置。在本例中，会在 com/myspace/mortgage_app 中创建。

4. 点 确定。图表编辑器将打开。

5. 在右上角点击 Properties  图标。

6. 向下滚动并展开 Process Data，然后点击 Process Variables 部分中的  。

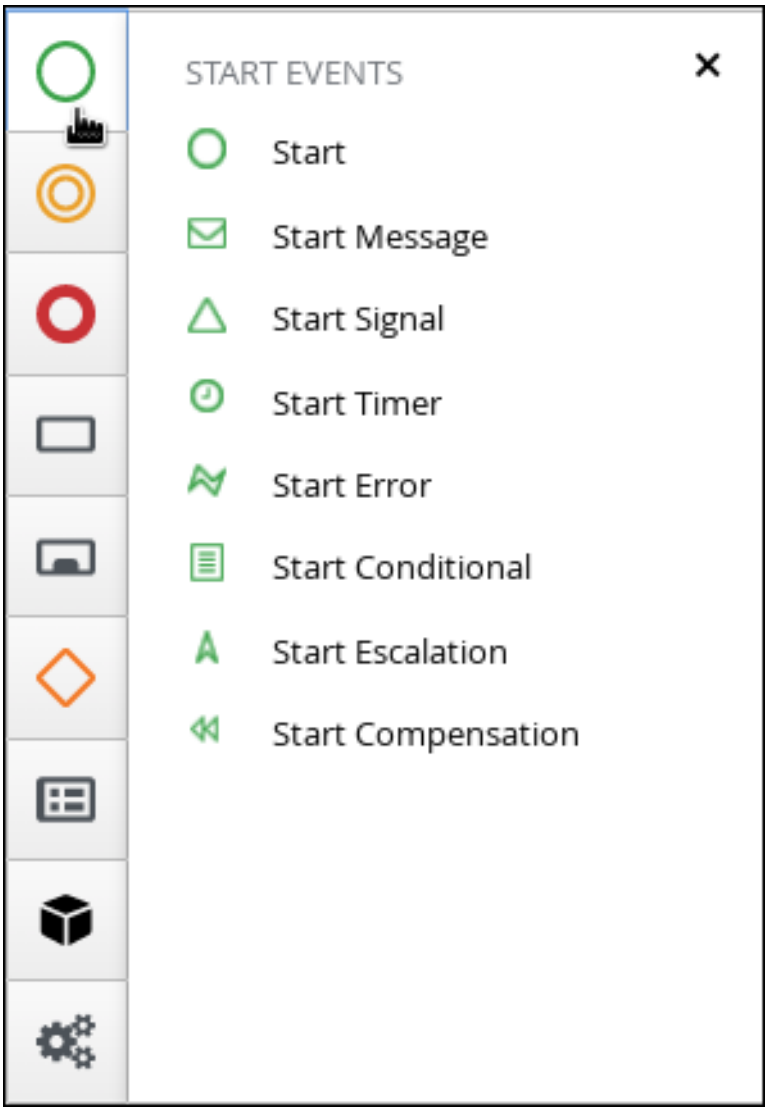
7. 输入以下值：

- 名称 ：应用程序
- 数据类型:Application [com.myspace.mortgage_app]

16.1.1. 创建出站连接和专用网关

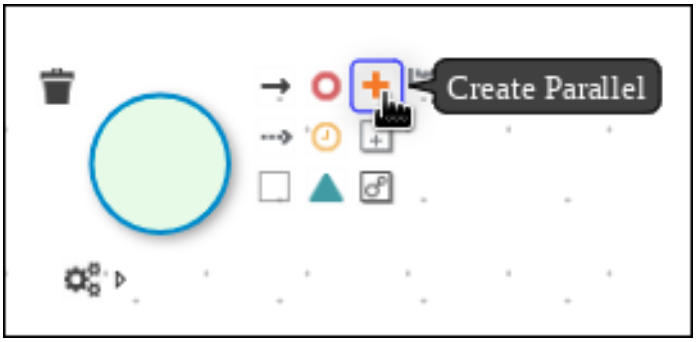
本节演示了如何创建传出连接、专用网关和业务规则任务。使用专用网关来做出决策，并根据可用数据对事件做出响应。

Red Hat Process Automation Manager 包含用于简化业务流程创建预定义节点类型。预定义的节点面板位于图表编辑器的左侧。



流程

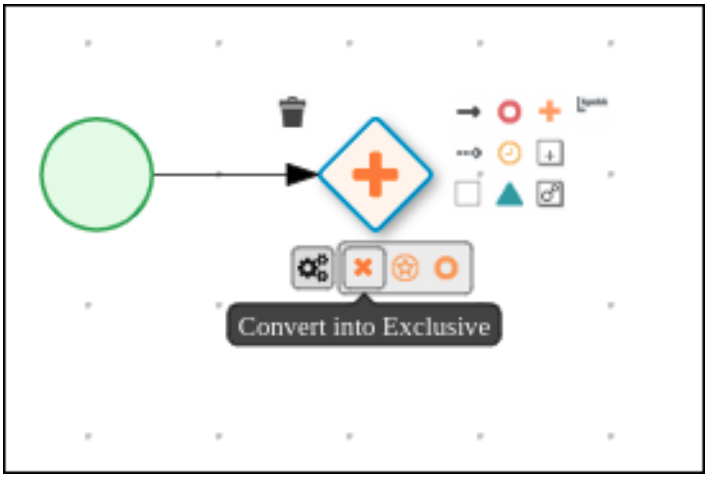
1. 将启动事件节点拖到 canvas 上。
2. 从启动事件创建传出连接到专用网关：
   a. 在 canvas 上，点击启动事件节点并点击 Create Parallel 图标。



b.

将鼠标悬停在并行的

图标上，然后点击 **Convert into Exclusive** 图标。



3.

从专用网关到商业规则任务创建传出连接：

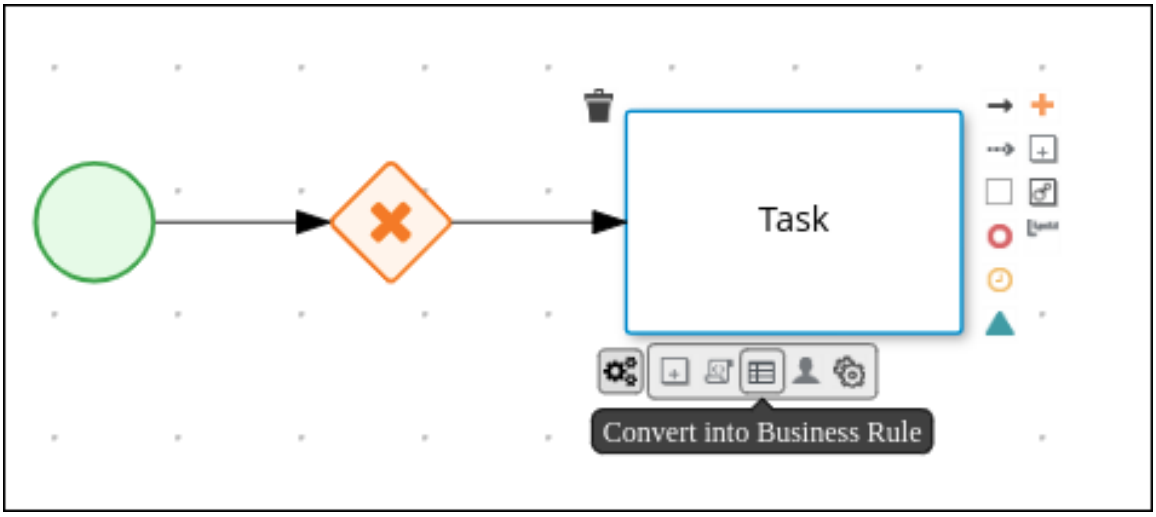
a.

在 canvas 上，点 exclusive gateway，再点击 **Create Task** 图标。

b.

将鼠标悬停在任务的

图标上，然后点击 **Convert into Business Rule** 图标。



4. **配置商业规则任务：**

a. **点击 业务规则任务。**

b. **如果没有打开 Properties 面板，点右上角的 Properties** 图标。

c. **在 Properties 面板中，在 Name 字段中输入 Validation。**

d. **展开 实现/执行，从 Rule Flow Group 菜单中选择 New，并输入 验证。**

e. **在 On Exit Action 字段中，输入以下 Java 表达式：**

```
System.out.println(application.getProperty());
```

f. **展开 数据分配，然后点击分配** 旁边的。

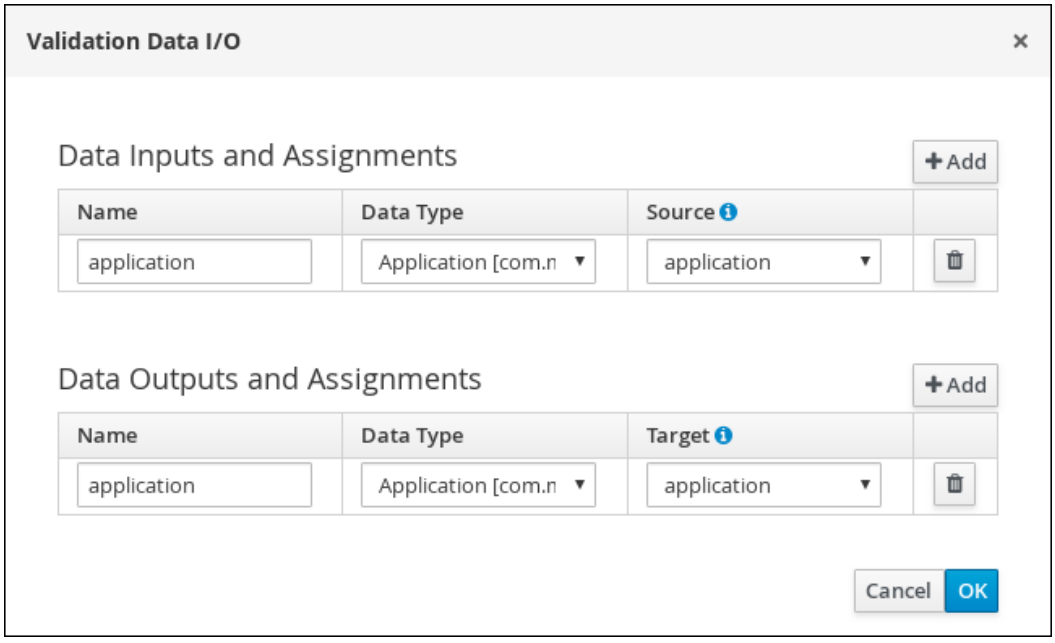
g. **在 Validation Data I/O 窗口中，点 Add 并创建以下分配：**

- **数据输入和分配**

- Name: application
- Data Type: Application [com.myspace.mortgage_app]
- 源：应用程序

- 数据类型和分配
  - Name: application
  - Data Type: Application [com.myspace.mortgage_app]
  - 目标：应用程序

图 16.1. 验证数据 I/O 分配



5. 在 Validation Data I/O 窗口中，单击 OK。

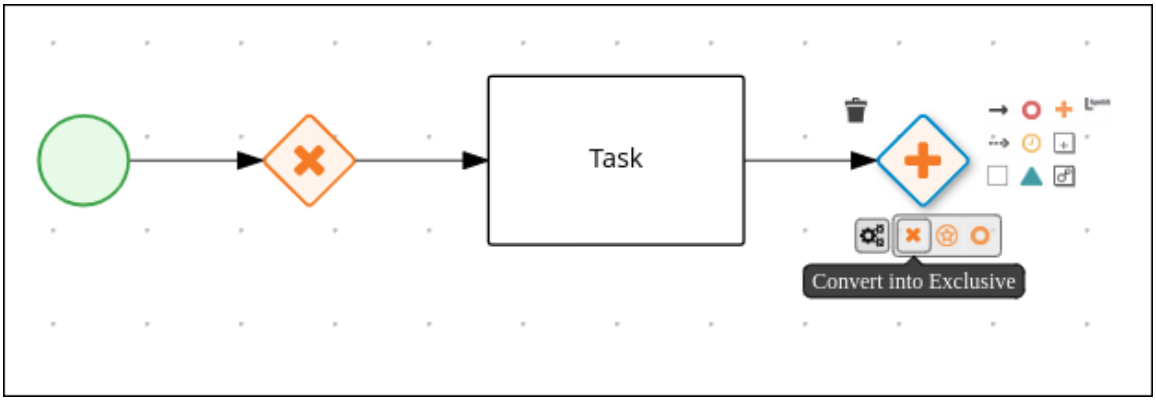
6. 在 canvas 的上方点击 Save 来确认您的更改。

16.1.2. 定义验证数据

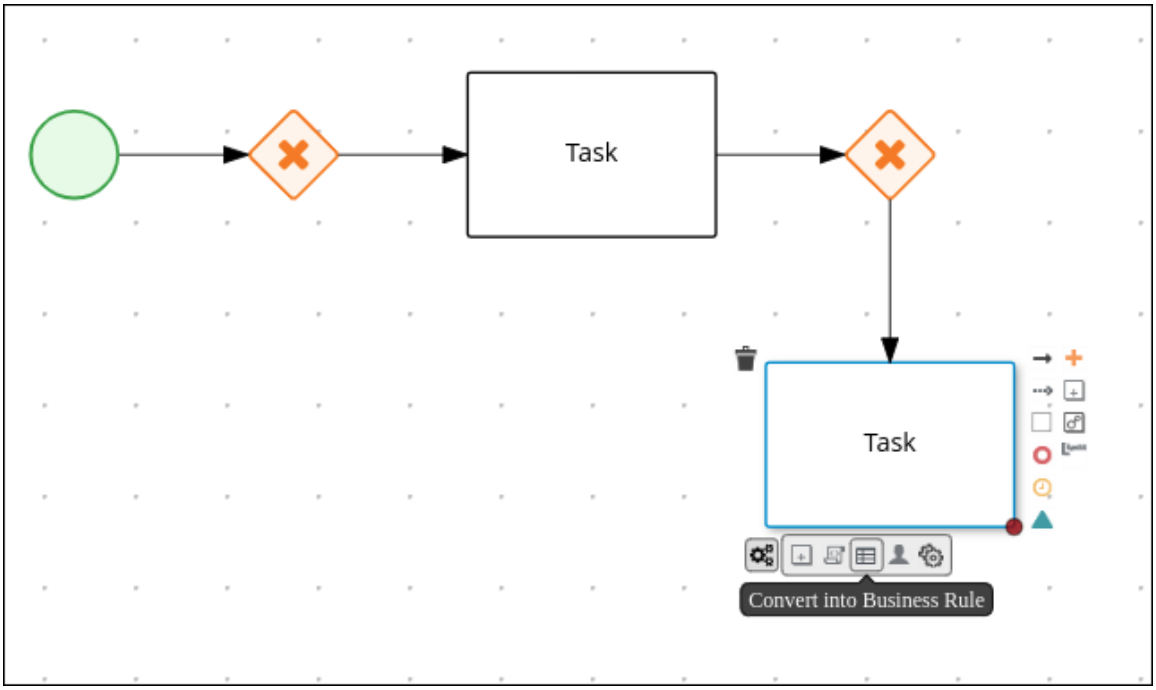
本节演示了如何定义验证数据，这些数据决定了应用程序数据是否正确、包含错误或缺少的信息。

流程

1. 从 Validation 任务创建传出连接到专用网关：
   a. 单击 Validation 任务，再单击 Create Parallel 图标。
   b. 将鼠标悬停在并行的  图标上，然后点击 Convert into Exclusive 图标。



2. 从专用网关创建传出连接到新商业规则任务：
   a. 点 exclusive Gateway，点 Create Task 图标。
   b. 在下列镜像所示，将新任务拖到专用网关下方。
   c. 将鼠标悬停在任务的  图标上，然后点击 Convert into Business Rule 图标。



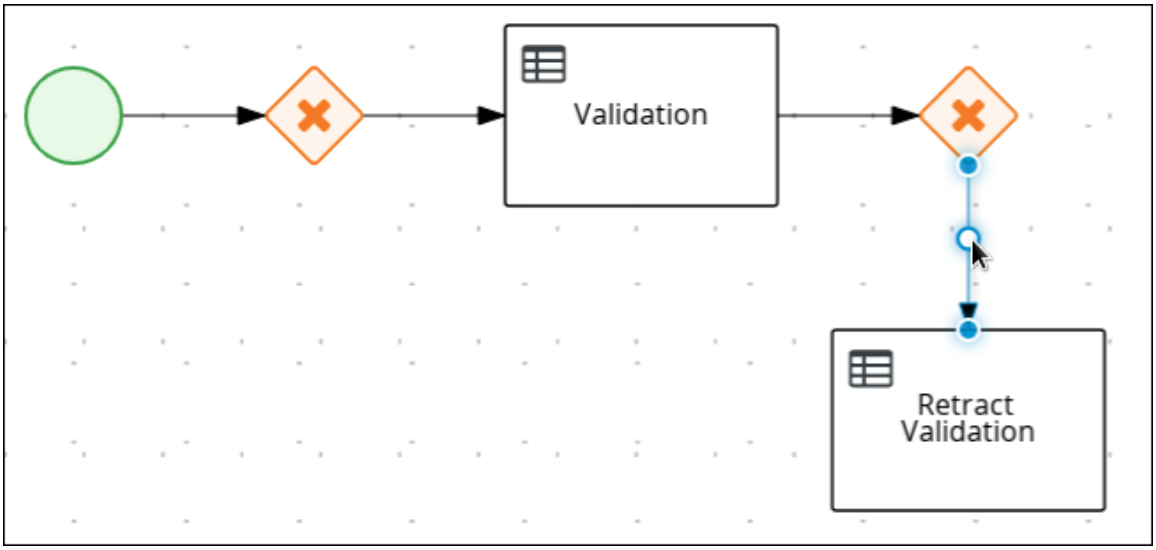
d. 如果没有打开 **Properties** 面板，点右上角的 **Properties** 图标。

e. 在 **Properties** 面板中，在 **Name** 字段中输入 **Retract Validation**。

f. 展开 **Implementation/Execution**，从 **Rule Flow Group** 菜单中选择 **New**，并 输入错误。

3. 配置专用网关和业务规则任务之间的连接：

a. 点连接。



b.

如果没有打开 Properties 面板，点右上角的 Properties 图标。

c.

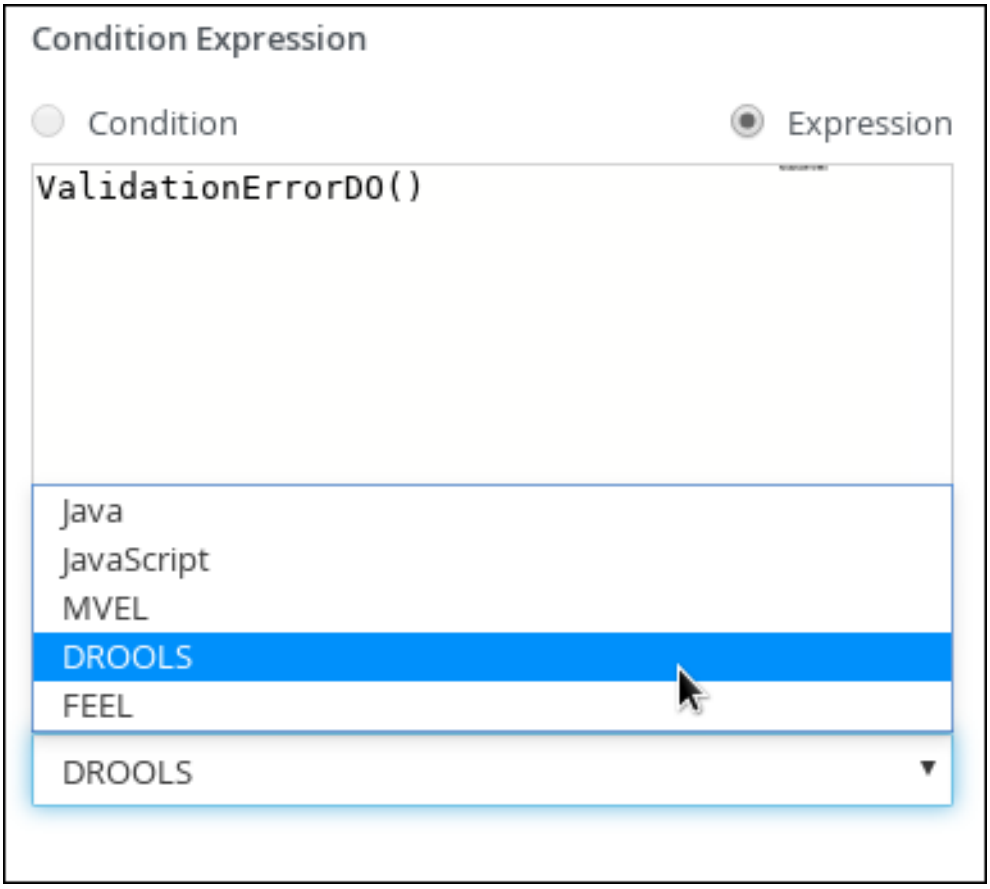
在 Properties 面板中，在 Name 字段中输入 Invalid。

d.

展开 实现/执行 并在条件表达式 部分中选择 Expression 。

e.

从列表中，选择 drools，然后在 Condition Expression 字段中输入 ValidationErrorDO（）。

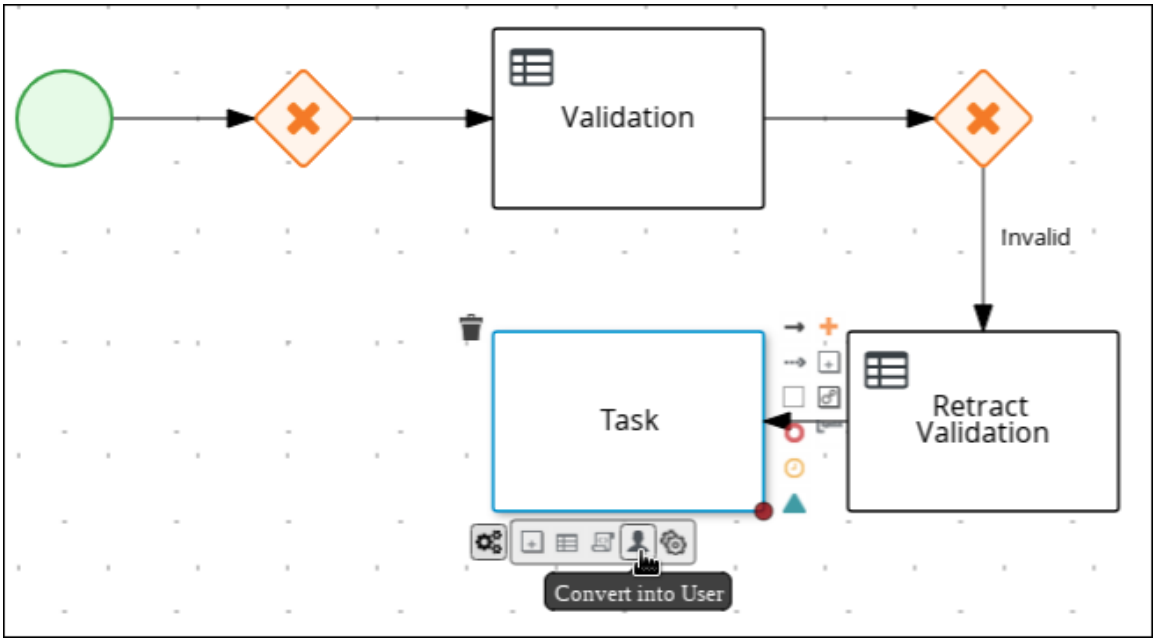


4. **从 Retract Validation 任务创建传出连接到新用户任务：**

   a. **点 Retract Validation 任务，点 Create Task 图标。**

   b. **在 Validation 任务下拖动新任务，如下所示。**

   c. **将鼠标悬停在任务的**  **图标上，然后点击 Convert into User 图标。**



d. 单击新用户任务并在 **Properties** 面板中的 **Name** 字段中输入 **Correct Data**。

e. 展开 实现/执行，并在 **Task Name** 字段中输入 **CorrectData**。

f. 从 **Groups** 菜单中选择 **New**，并输入 代理。

g. 点 **Assignments** 旁边的

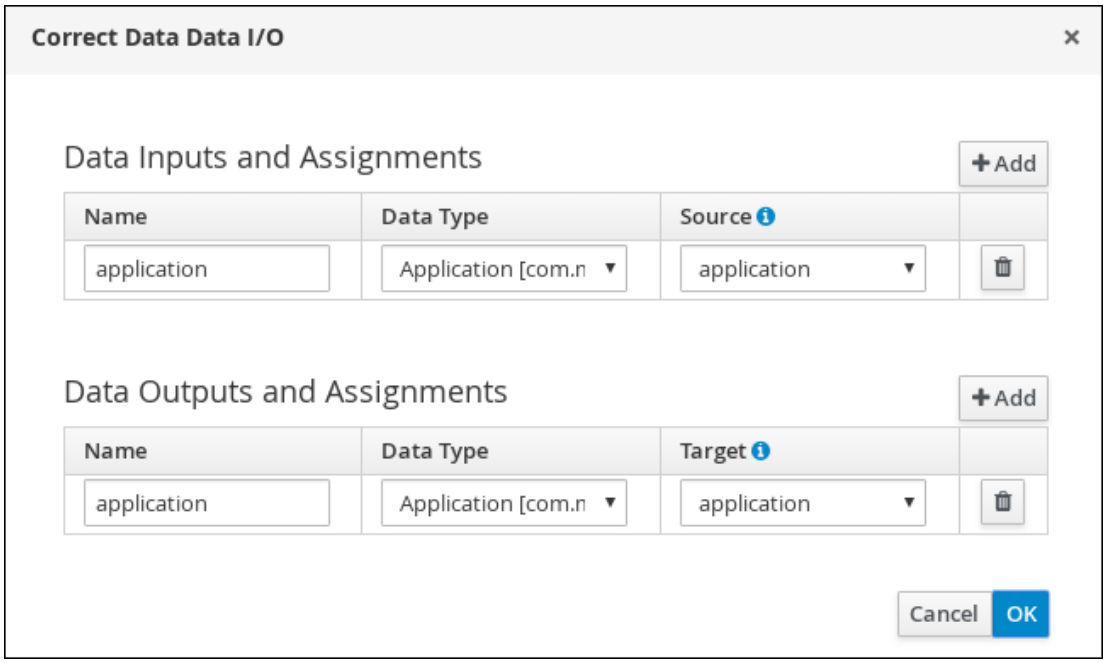
。

h. 在 **Correct Data I/O** 窗口中，点 **Add** 并创建以下分配：

- **Name: application**
- **Data Type: Application [com.myspace.mortgage_app]**
- 源：应用程序
  - 数据类型和分配

- **Name: application**
- **Data Type: Application [com.myspace.mortgage_app]**
- **目标：应用程序**

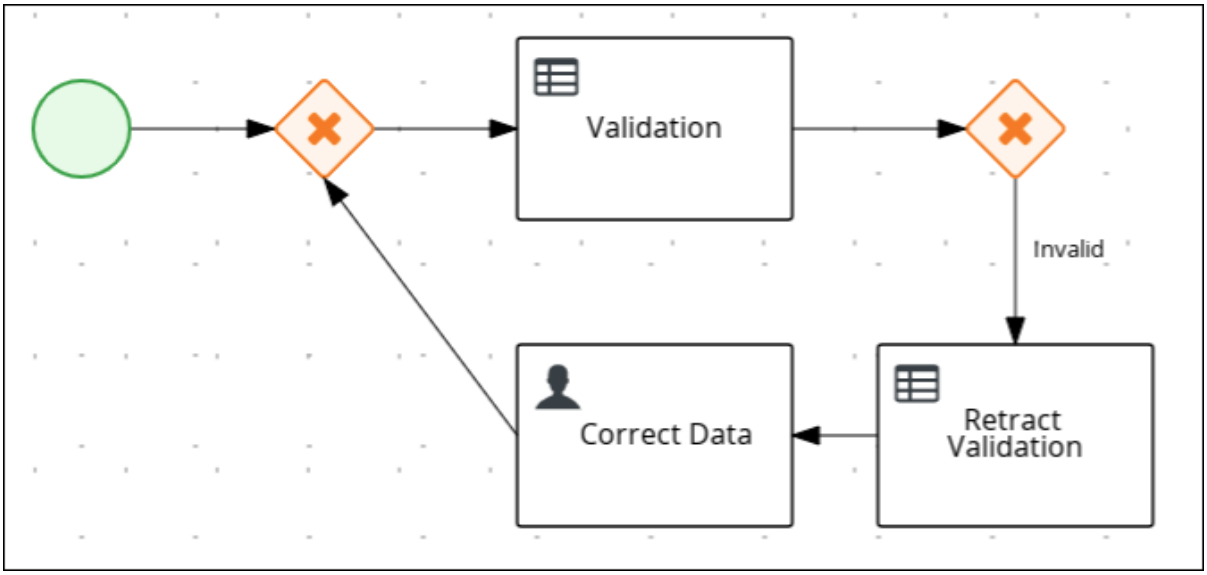
图 16.2. 正确的数据 I/O 分配



i. **在 Correct Data I/O 窗口中点击 OK。**

j. **在 canvas 的上方点击 Save。**

5. **单击 Correct Data user 任务，然后单击 Create sequence Flow 图标，并将它拖到第一个专用网关。您的工作流应类似以下示例：**

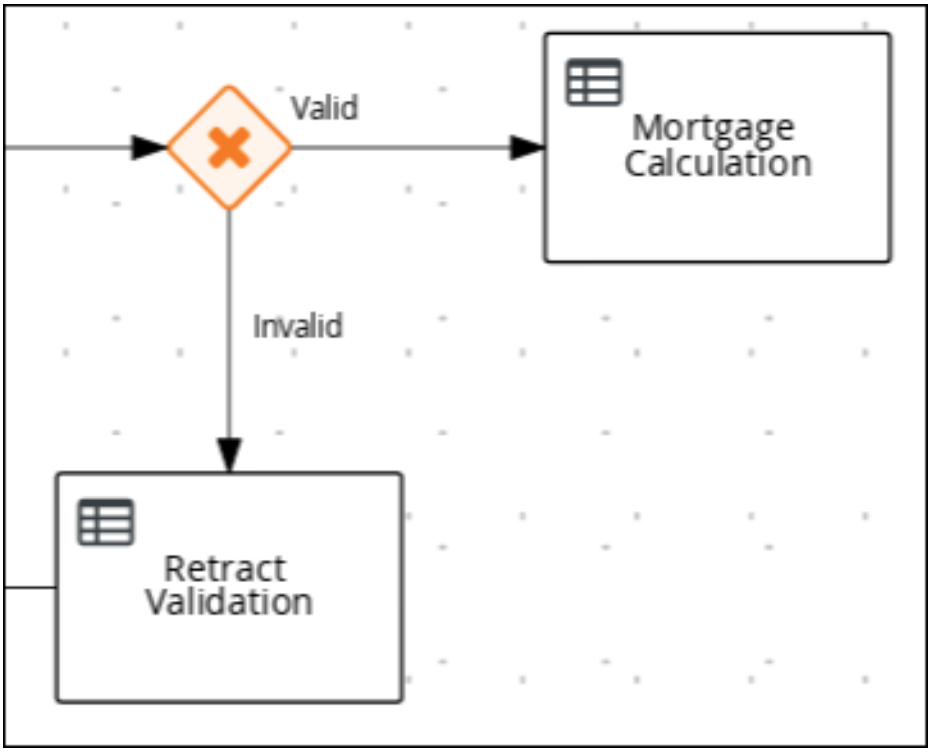


16.1.3. 计算mortgage

mortgage 计算业务流程决定了申请者的抵押制约限制。

流程

1. **返回到第二个专用网关，再创建与新警报规则任务的传出连接。**
2. **点创建的连接并在 Properties 面板中的 Name 字段中输入 Valid。**
   a. **展开 实现/执行 并在条件表达式 部分中选择 Expression 。**
   b. **从列表中，选择 drools，并在 Condition Expression 字段中输入 not ValidationErrorDO（）。**
3. **在 Properties 面板中点新的业务规则任务和 Properties 面板，在 Name 字段中输入 Mortgage Calculation。**



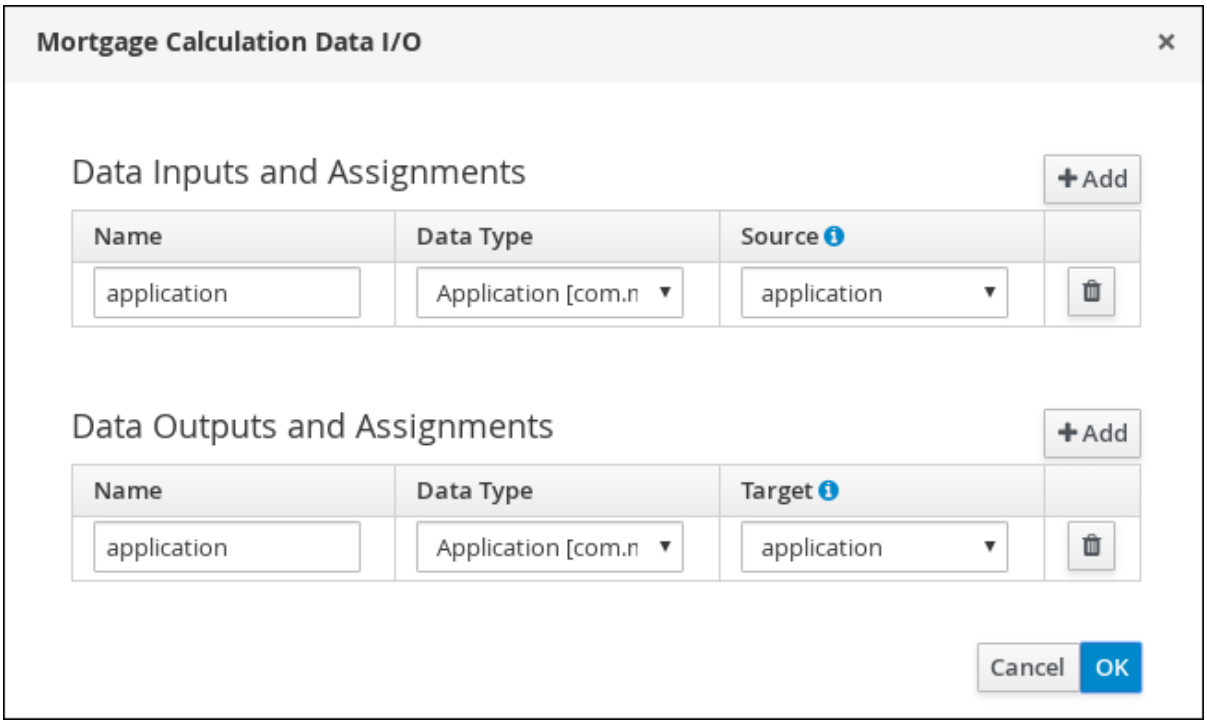
a. **展开 实施/执行，从 规则流组 菜单中选择 New，并输入m ortgagecalculation。**

4. **展开 数据分配，然后点击分配**

旁边的。

5. **在 Mortgage Calculation Data I/O 窗口中，点 Add 创建以下分配并点 Save。**

图 16.3. mortgage Calculation Data I/O 分配



6. 在 Mortgage Calculation Data I/O 窗口中单击 OK。

7. 点 canvas 上的空空间，向下滚动、展开 Process Data，然后点击 Process Variables 旁边的  。输入以下值：

- 名称 ：inlimit
- 数据类型 ：布尔值

8. 从 Mortgage Calculation 任务创建传出连接到新用户任务。

9. 单击用户任务，在 Name 字段中输入 Qualify。

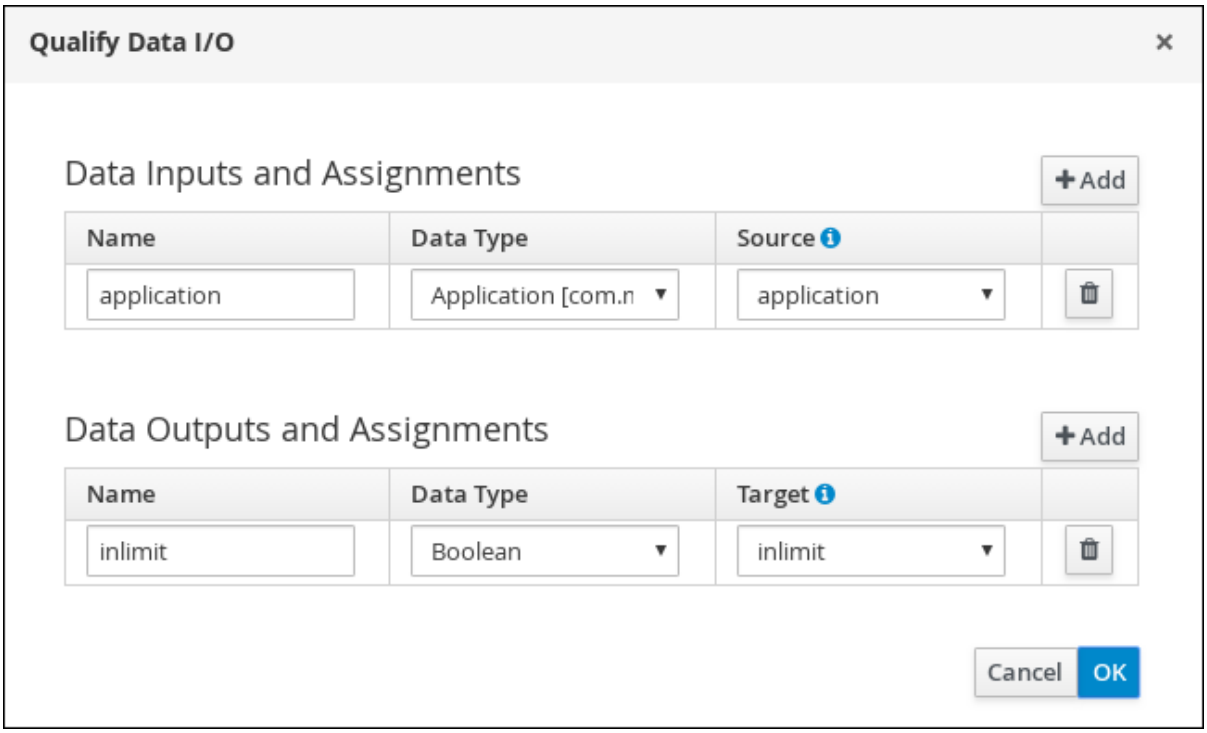
10. 展开 实现/执行，并在 Task Name 字段中输入 Qualify。

11. **从 Groups 菜单中选择 New，并输入 批准人。**

12. **点 Assignments 旁边的**

。在 Qualify Data I/O 窗口中，点 Add 创建以下分配：

图 16.4. 合格数据 I/O 分配



13. **在 Qualify Data I/O 窗口中，单击 OK。**

14. **在 canvas 的上方点击 Save 来确认您的更改。**

15. **单击 Qualify 用户任务，单击 Create parallel 菜单图标，并将它转换为专用网关。**

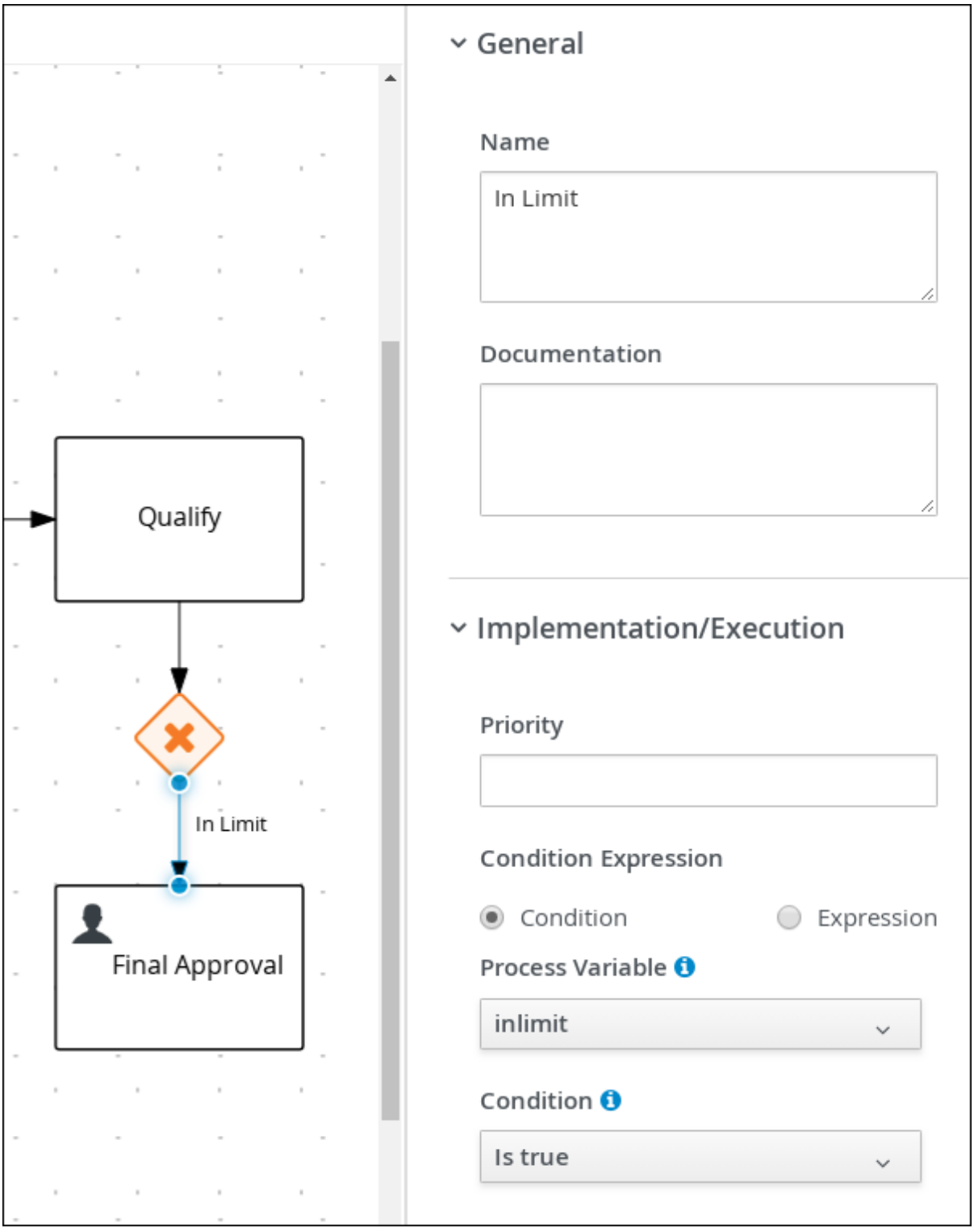
16. **在 Qualify 用户任务下拖动新的专用网关。**

17. **从专用网关创建传出连接，并将它连接到新用户任务。**

18. **在 Properties 面板的 Name 字段中，点连接并在 Limit 中输入。**

19. **展开 实现/执行 并在 Condition Expression 部分中选择 Condition。**

20. **从 Process Variable 下拉菜单中选择 inlimit，然后从 Condition 下拉菜单中选择 Is true。**



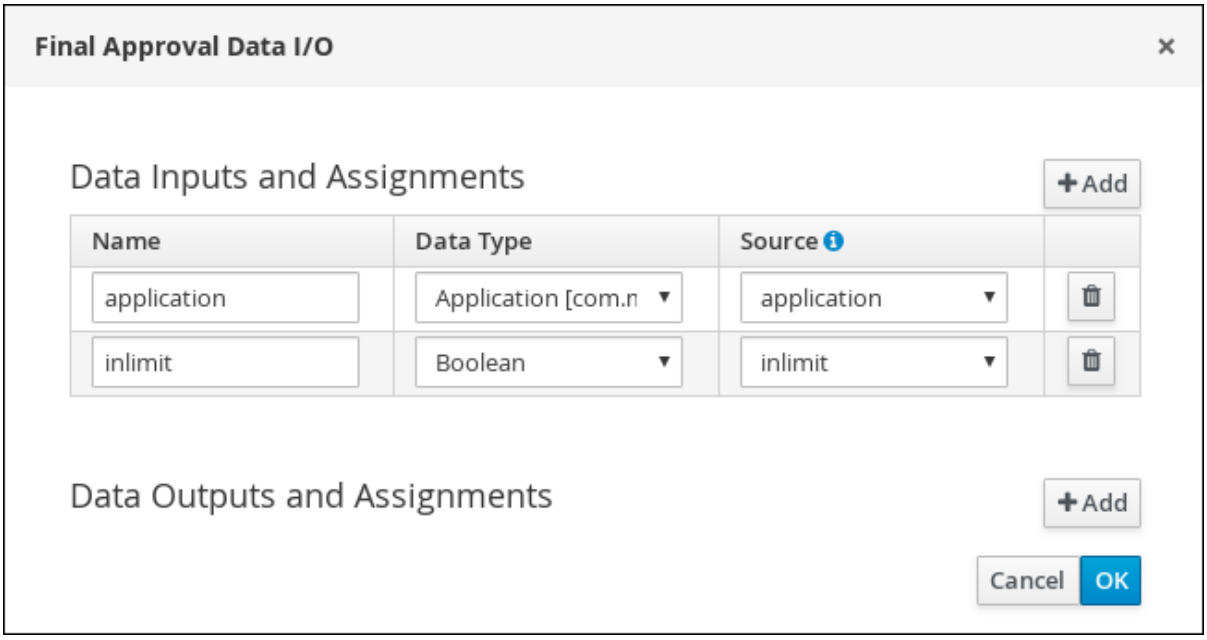
21. **点用户任务，在 Name 字段中输入 final Approval。**

22. 展开 Implementation/Execution，在 Task Name 字段中输入 FinalApproval。

23. 从 Groups 菜单中选择 New，并输入 manager。

24. 点 Assignments 旁边的  。在 最后的 Approval Data I/O 窗口中，点 Add 创建以下分配：

图 16.5. 最终批准数据 I/O 分配



25. 在 最后批准数据 I/O 窗口中点击 OK。

26. 在 canvas 的上方点击 Save 来确认您的更改。

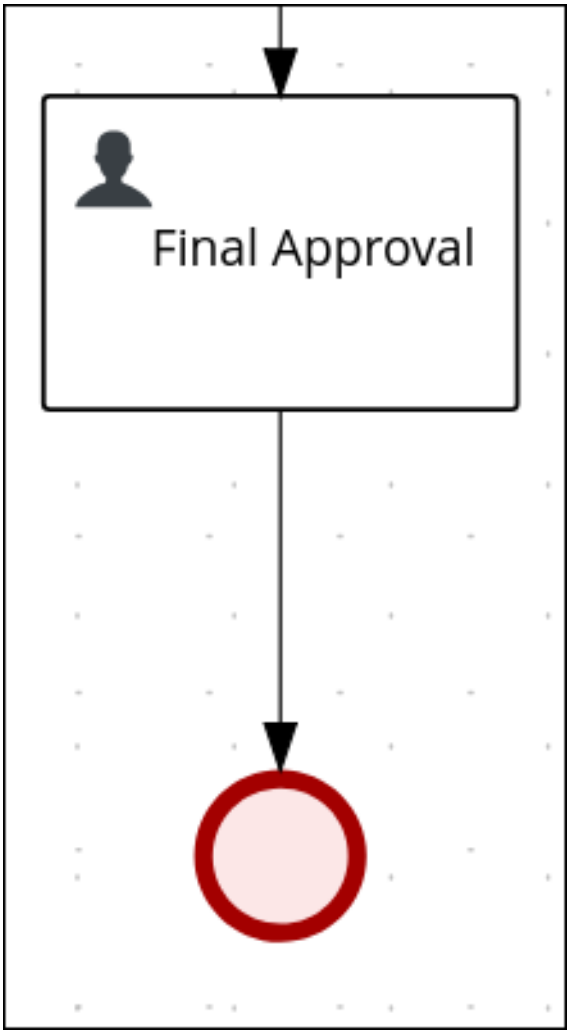
16.1.4. 增加停机时间

增加故障支付业务流程，通过增加其支付方式，检查其是否代表金级评分。最终结果是最终的 loan 批准，或基于申请者无法增加支付的金金拒绝。

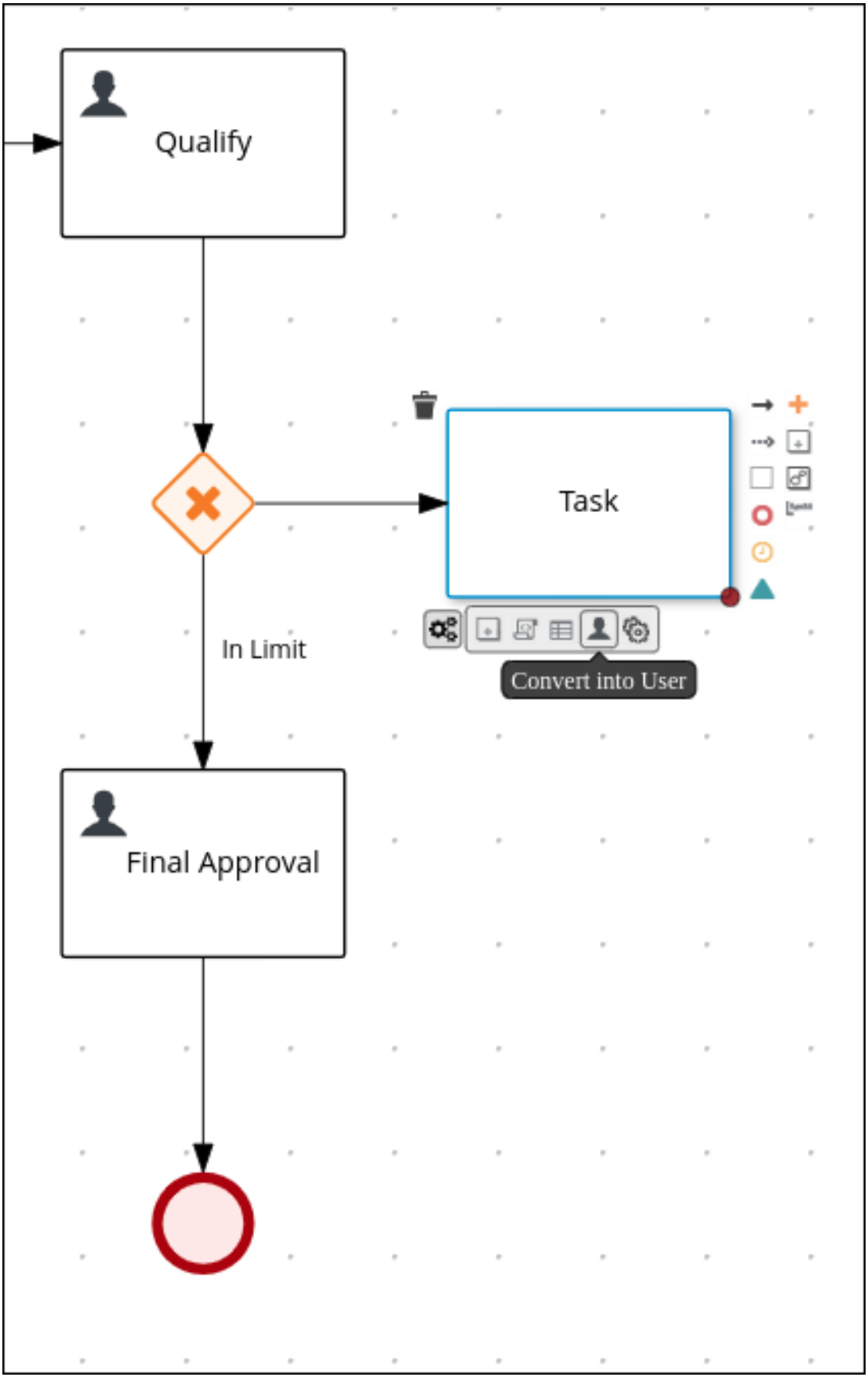
流程

1. 点击 最终批准 用户任务，然后从用户任务快速菜单中选择 Create End。

2. 将最终 批准用户任务下面的最终 事件移动。



3. 返回到与 最终批准 用户任务连接的专用网关。创建第二个传出连接，并将它连接到新用户任务。



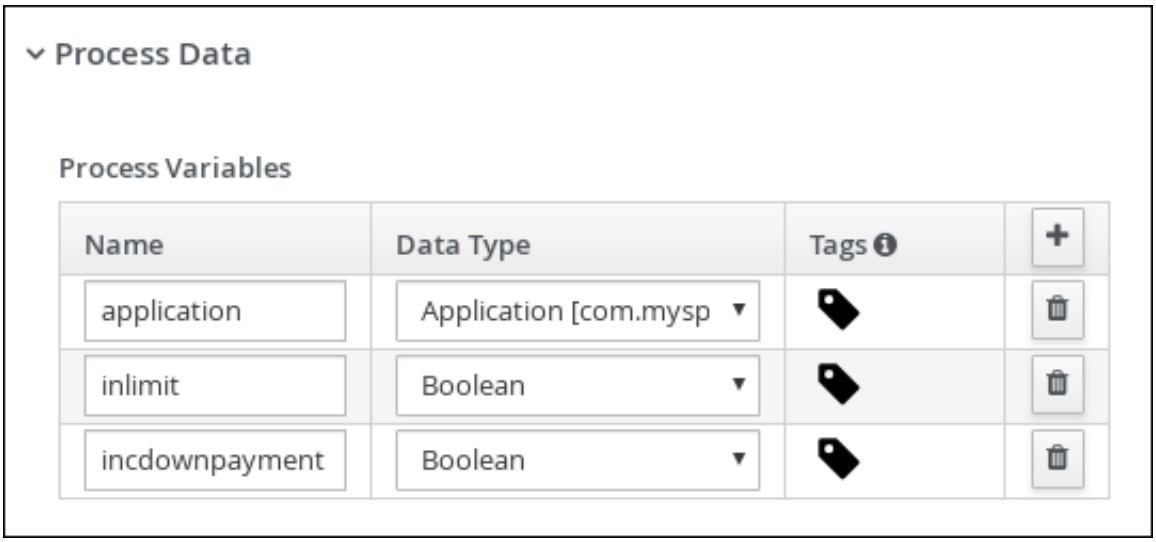
4. **在 Properties 面板的 Name 字段中，点连接和输入 Not in Limit。**

5. **展开 实现/执行 并在 Condition Expression 部分中选择 Condition。**

6. **从 Process Variable 下拉菜单中选择 inlimit，然后从 Condition 下拉菜单中选择 Is false。**

7. **点 canvas 上的空空间，向下滚动、展开 Process Data，然后点击 Process Variables 旁边的**  **。输入以下值：**

   - **名称 ：incdownpayment**
   - **数据类型 ：布尔值**



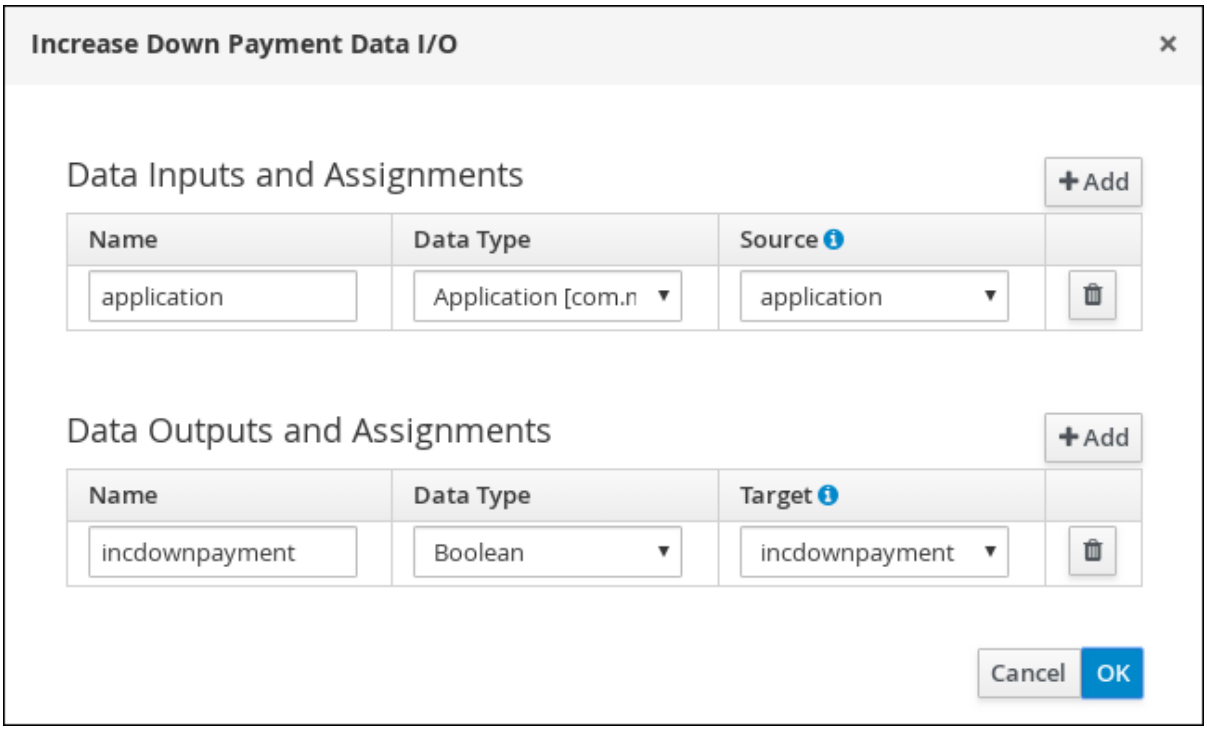
8. **在 Properties 面板中点新用户任务，在 Name 字段中输入 increase Down pay。**

9. **展开 实现/执行，并在 Task Name 字段中输入 increases DownPayment。**

10. **从 Groups 菜单中选择 New，并输入 代理。**

11. **点 Assignments 旁边的**  **。在 Increase Down pay Data I/O 窗口中，点 Add 创建以下分配：**

图 16.6. 增加关闭支付数据 I/O 分配



12. **在 increased Down pay Data I/O 窗口中点 OK。**

13. **在 canvas 的上方点击 Save 来确认您的更改。**

14. **点 increase Down pay user 任务，点 Create parallel 菜单图标，然后将其转换为专用网关。**

15. **在 Increase Down pay user 任务下拖动新的专用网关。**

16. **创建从专用网关到最终事件的传出连接。**

17. **点 Properties 面板的 Name 字段中的连接和输入 关闭支付增加。**

18. **展开 实现/执行 并在 Condition Expression 部分中选择 Expresssion。**

19. **输入 返回 !incdownpayment ； 然后从下拉菜单中选择 java。**

20. **从专用网关创建传出连接，并将它连接到第一个专用网关。**

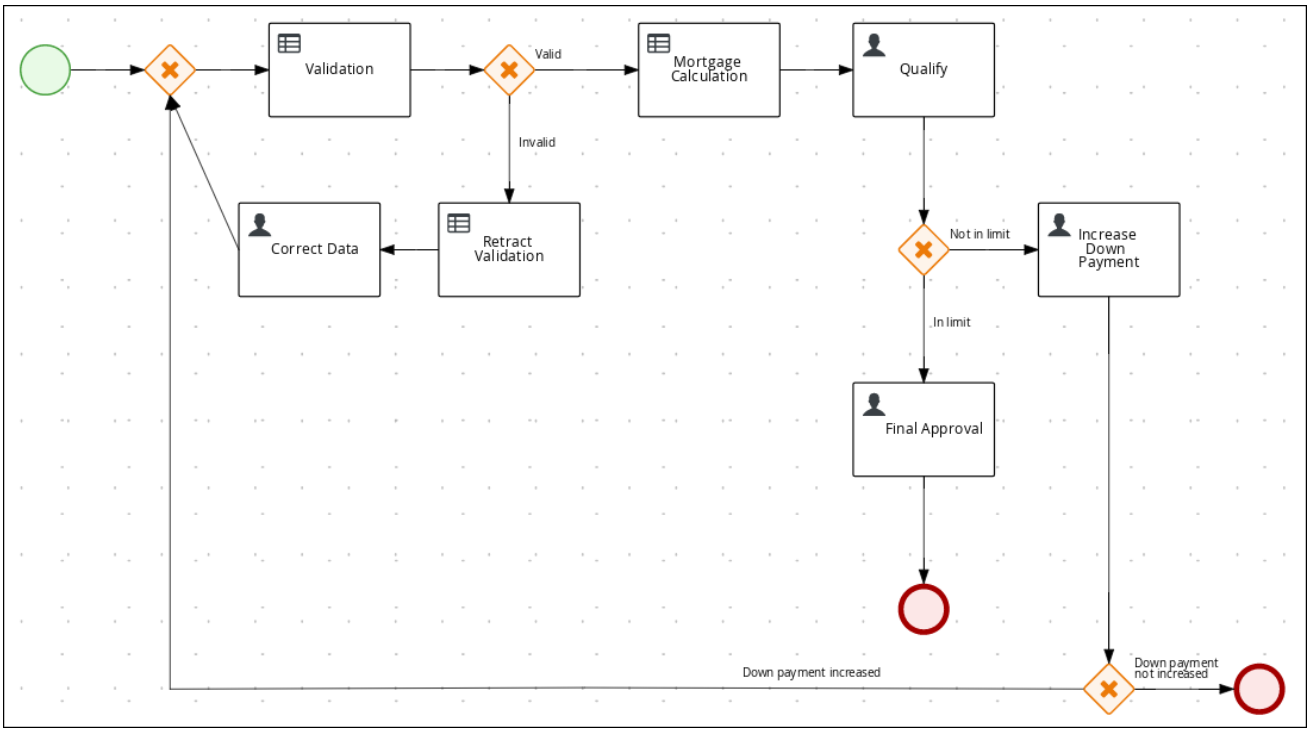
21. **点 Properties 面板的 Name 字段中的连接和输入 关闭支付增加。**

22. **展开 实现/执行 并在 Condition Expression 部分中选择 Expresssion。**

23. **输入 返回提示；然后从下拉菜单中选择 java。**

24. **在 canvas 上方，点 Save 确认更改并保存整个业务流程。**

图 **16.7.** 业务流程的最终版本



第 17 章 指导规则

指导规则是您通过规则创建在业务中心基于 UI 的指导规则设计者中创建的业务规则。指导规则设计器根据所定义规则的数据对象，提供可接受输入的字段和选项。您定义的指南规则与所有其他规则资产一样编译为 Drools Rule Language(DRL)规则。

与指导规则相关的所有数据对象都必须位于与指导规则相同的项目软件包中。默认导入同一软件包中的资产。创建必要的数据对象和指导规则后，您可以使用指南规则设计器的 Data Objects 选项卡来验证所有所需数据对象是否已列出或导入其他现有数据对象，方法是添加新项目。

17.1. 查看 MORTGAGE_PROCESS 业务规则

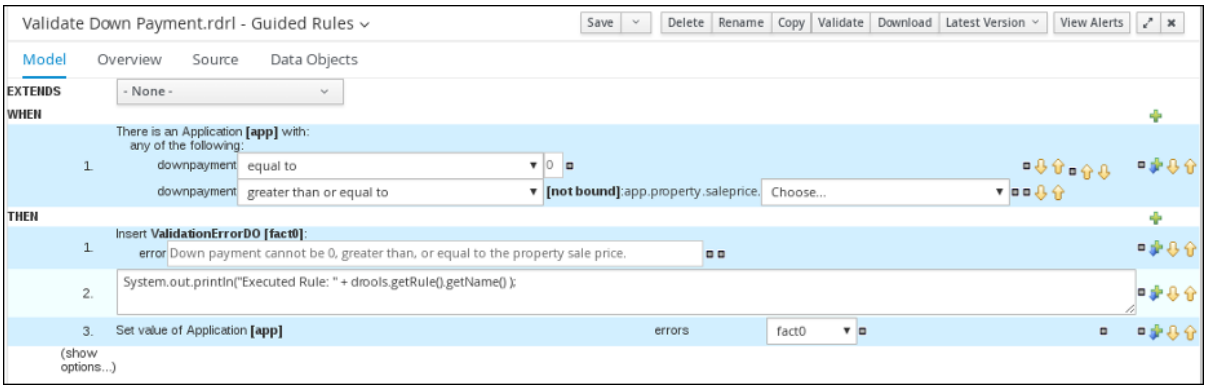
本章的目标是向您介绍 Mortgage_Process 项目的预定义业务规则。在本教程中，您不创建并定义业务规则。相反，请查看在 Mortgage_Process 示例项目中已定义的 WHEN 和 THEN 规则。有关创建指南的业务规则的详情，请参考使用 *指导规则设计决策服务*。

17.1.1. 查看 Validate Down pay Guided 规则

查看 WHEN 和 THEN 规则，以便您了解以后运行进程时如何设置和使用条件。

流程

1. 点 Menu → Design → Projects，然后点 Mortgage_Process。
2. 在 asset 列表中，点击右箭头查看资产列表的第二个页面，并点击 Validate Down pay guided 规则。
3. 查看 Validate Down pay Guided 规则的 WHEN 和 THEN 条件和值。



17.1.2. 查看周期性指南 规则

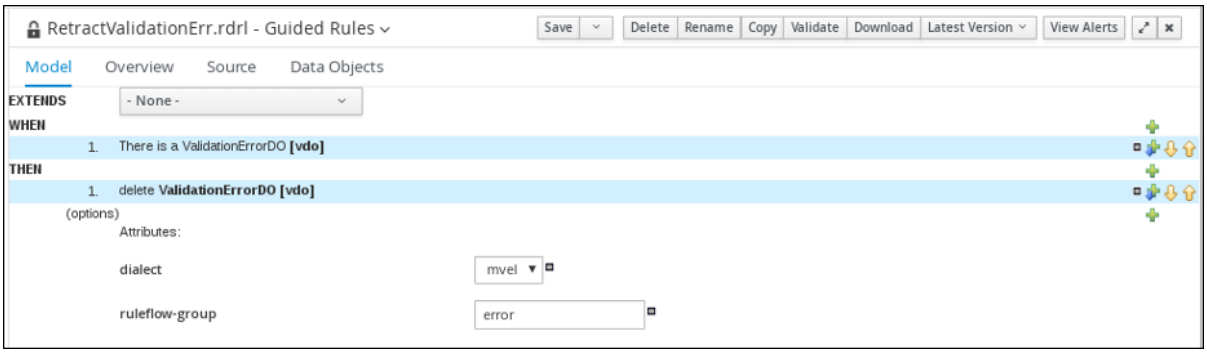
查看 WHEN 和 THEN 规则，以便您了解以后运行进程时如何设置和使用条件。

流程

1. 点 Menu → Design → Projects，然后点 Mortgage_Process。

2. 从资产列表中，点击右箭头查看资产列表的第二个页面，然后点击 RetractValidationErr Guided 规则。

3. 查看 RetractValidationErr Guided 规则的 WHEN 和 THEN 条件和值。



第 18 章 指导的决定表

指导的决策表是表格决策表的向导替代选择，以表格格式定义业务规则。借助指导的决策表，您由 Business Central 中的基于 UI 的向导领导，可帮助您根据项目中的指定数据对象定义规则属性、元数据、条件和操作。创建引导式练习后，您定义的规则会和所有其他规则资产一起编译为 Drools 规则语言 (DRL)规则。

与指导决策表相同的项目软件包中，所有相关的数据对象都必须位于与指导的决策表相同的数据对象。默认导入同一软件包中的资产。创建必要的数据对象和指导决策表后，您可以使用《指导决策表设计器中的数据 对象 "选项卡，验证所有所需的数据对象还是通过添加新项目来导入其他现有数据对象。

18.1. 查看 MORTGAGE 决策表

本章的目标是为您介绍 MortgageDecisionTable 决策表。在本教程中，您不会创建和设置决策表条件。相反，请查看 Mortgage_Process 示例应用程序项目的 MortgageDecisionTable Guided Decision Tables 资产中定义的值和条件。有关创建决策表的详情，请参考使用 *指导的决策表设计决策服务*。

先决条件

- 业务规则已定义。更多信息请参阅 第 17.1 节 “查看 Mortgage_Process 业务规则”。

流程

1. 在 Business Central 中，前往 Menu → Design → Projects → Mortgage_Process。
2. 向下滚动并单击 MortgageDecisionTable Guided Decision Tables asset。

MortgageDecisionTable

#	Description	ruleflow-group	Applicant Annual Income		Property			application
			$greater	$less_or_equal	$saleprice_less	$age_less	$location	Mortgage Amount
1		mortgagecalculation	100000	200000	300000	5	Urban	200000
2		mortgagecalculation	50000	99999	100000	10	Rural	100000

第 19 章 BUSINESS CENTRAL 中的表单

表单是页面的布局定义，定义为 HTML，在进程和任务实例化过程中作为用户的对话框窗口显示。任务表单从用户获取进程和任务实例执行的数据，而进程表单则取进程变量的输入和输出。

然后，使用数据输入分配将输入映射到任务，您可以在任务内部使用。任务完成后，数据映射为数据输出分配，以便为父进程实例提供数据。

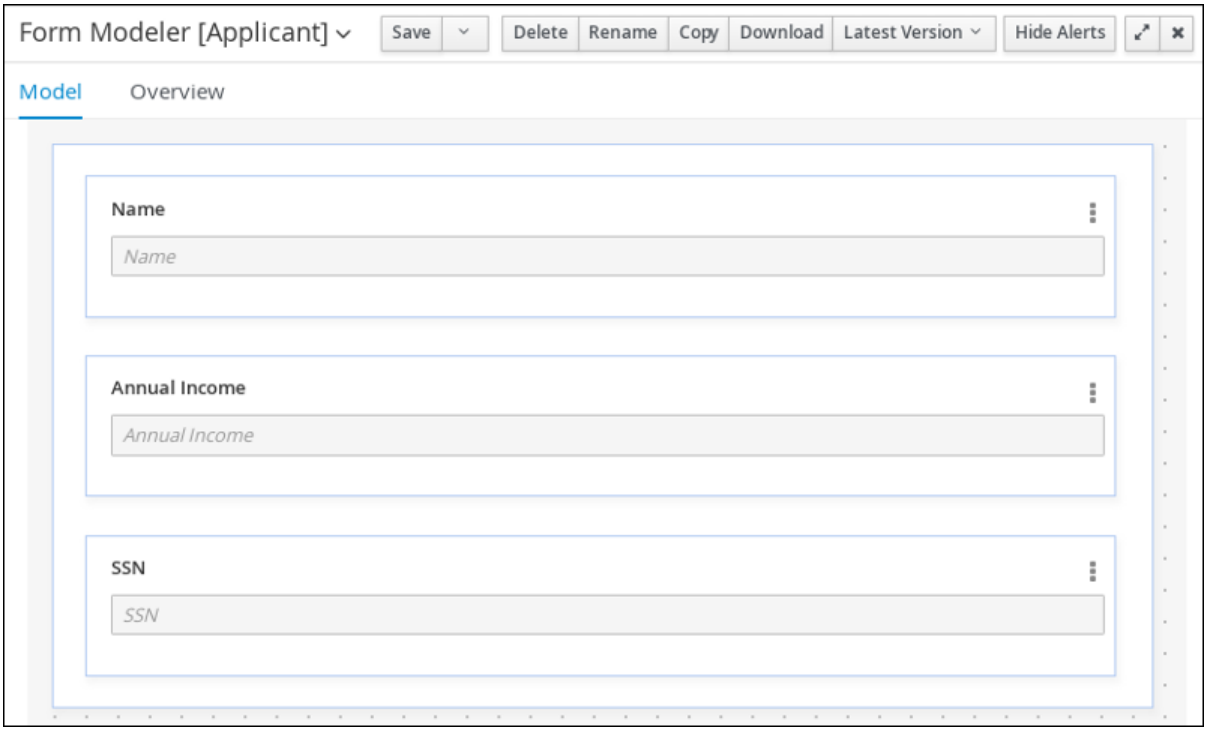
19.1. 查看 MORTGAGE_PROCESS 表单

本章的目标是介绍 Mortgage_Process 示例项目的预定义表单，用于为抵rtgage 应用程序进程收集用户数据。在本教程中，您不会创建并修改 Mortgage_Process 表单。相反，请查看预定义的示例表单。有关创建表单的详情，*请参阅使用 BPMN 模型设计业务流程*。

流程

1. 在 Business Central 中，前往 Menu → Design → Projects → Mortgage_Process。
2. 在 asset 列表中，点击右箭头查看资产列表的第二个页面，然后选择 Applicant 表单。

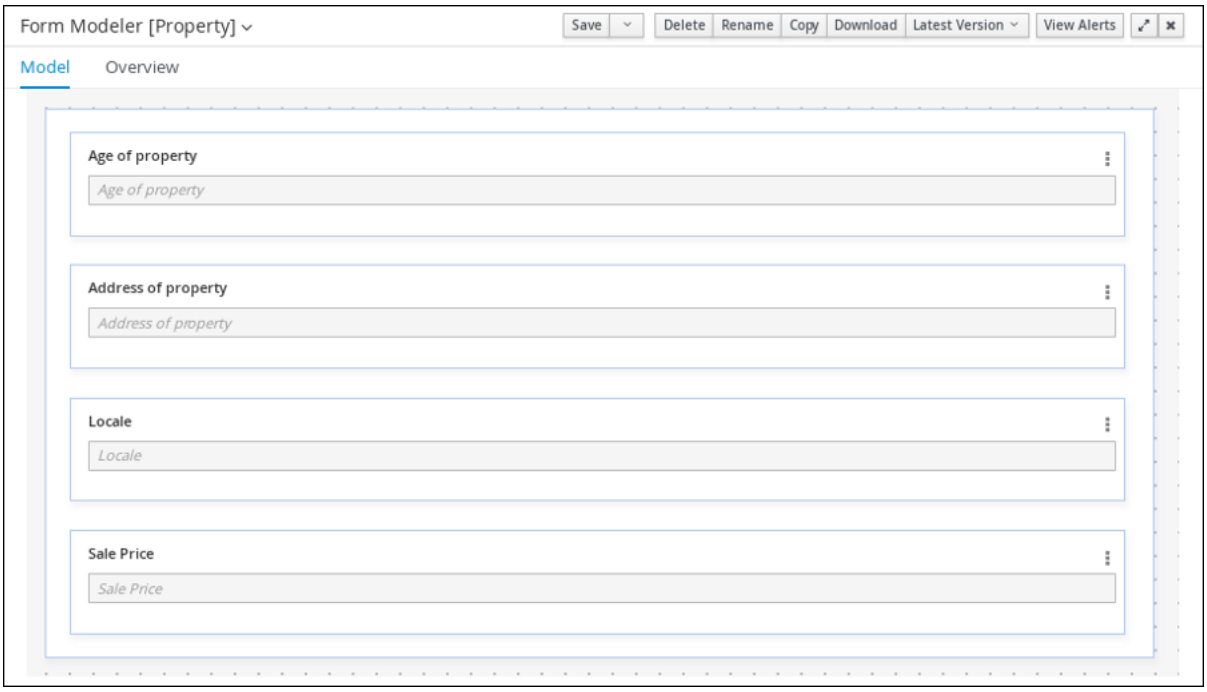
图 19.1. applicant 示例表单



3. 点 Menu → Design → Projects → Mortgage_Process。

4. 在 asset 列表中，选择 Property 表单。Property 表单在以下截屏中显示：

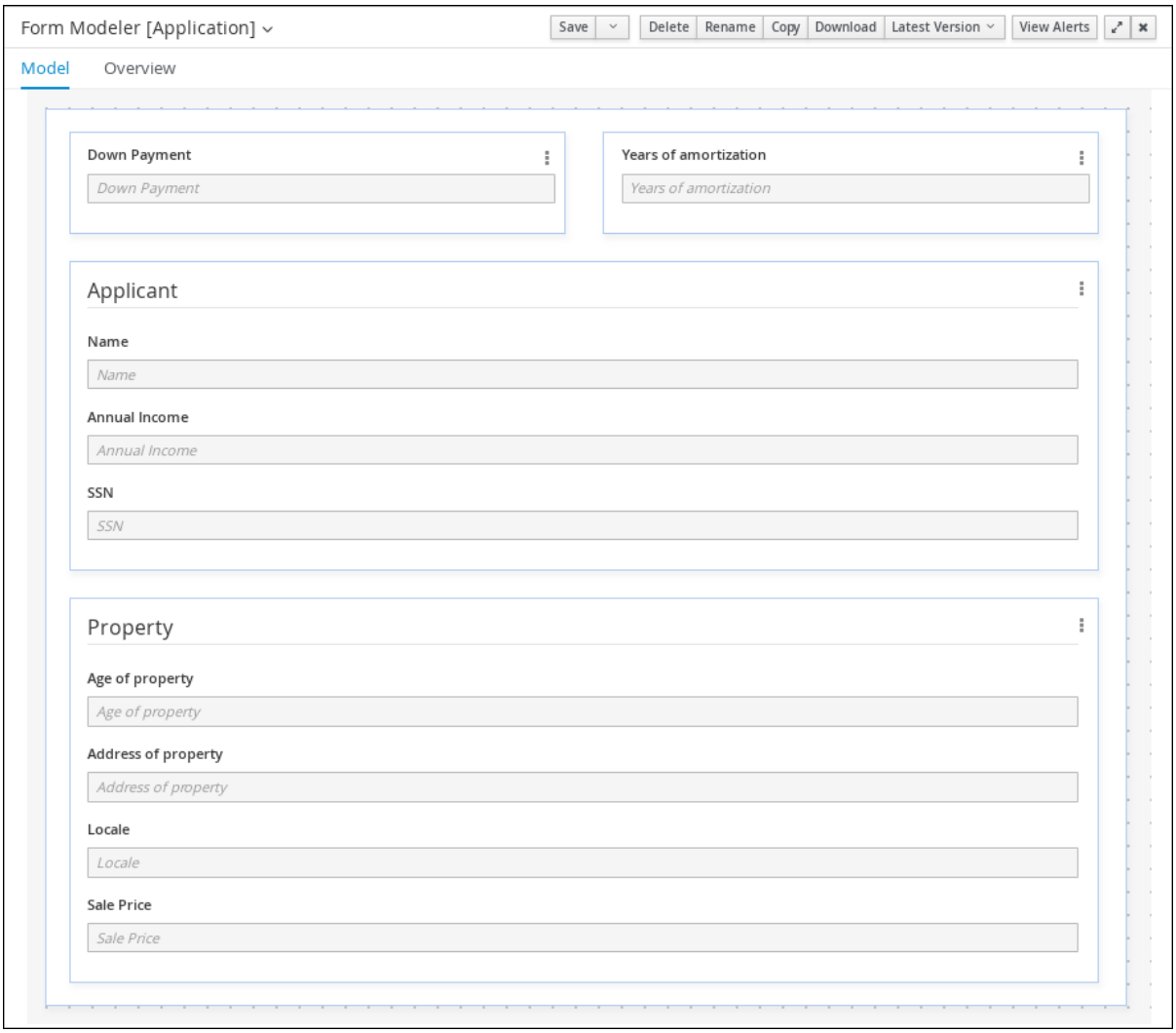
图 19.2. 属性示例表单



5. 点 Menu → Design → Projects → Mortgage_Process。

6. 在资产列表中选择 应用程序 表单。应用程序 表单在以下截屏中显示：

图 19.3. 应用程序示例表单



7. **单击右上角的 X 图标以关闭编辑器。**

第 20 章 部署 MORTGAGEAPPROVALPROCESS 进程应用程序

下面的章节将指导您如何在 Red Hat Process Automation Manager 中构建和部署 Mortgage_Process 应用的新实例。

先决条件

- KIE 服务器已部署并连接到 Business Central。

流程

1. 在 Business Central 中，前往 Menu → Design → Projects → Mortgage_Process。
2. 单击 Deploy。

- 如果项目名称中没有包括 KIE 容器（部署单元），则会自动创建具有默认值的容器。
- 如果已经部署了旧版本的项目，请转到项目设置并更改项目版本。完成后，保存更改并点击 Deploy。这会部署一个新版本的同一项目，以及最新的变化，以及旧版本。

注意

您还可以选择 Build & Install 选项来构建项目，并将 KJAR 文件发布到配置的 Maven 存储库，而无需部署到 KIE 服务器。在开发环境中，您可以点击 Deploy 将构建的 KJAR 文件部署到 KIE 服务器，而无需停止任何正在运行的实例（如果适用），或者点击 Redeploy 来部署构建的 KJAR 文件并替换所有实例。下次部署或重新部署构建的 KJAR 时，以前的部署单元（KIE 容器）会在同一目标 KIE 服务器中自动更新。在生产环境中，Redeploy 选项被禁用，且只能点击 Deploy 将构建的 KJAR 文件部署到 KIE 服务器上的新部署单元（KIE 容器）。

要配置 KIE 服务器环境模式，请将 org.kie.server.mode 系统属性设置为 org.kie.server.mode=development 或 org.kie.server.mode=production。要在 Business Central 中为对应项目配置部署行为，请转至 Project Settings → General Settings → Version，再切换 Development Mode 选项。默认情况下，Business Central 中的 KIE 服务器和所有新项目均为开发模式。您不能部署打开开发模式的项目，或使用手动将 SNAPSHOT 版本后缀添加到生产模式中的 KIE 服务器。

3. 要查看项目部署详情，可在屏幕顶部的部署横幅或 Deploy 下拉菜单中点击 View deployment details。这个选项将您定向到 Menu → Deploy → Execution Servers 页面。

第 21 章 执行 MORTGAGEAPPROVALPROCESS 进程应用程序

现在您已经部署了项目，您可以执行项目的定义功能。对于本教程，您的输入数据为抵rtgage 应用程序表单中，作为抵rtgage 代理。MortgageApprovalProcess 业务流程运行并确定申请者是否根据您之前定义的决策规则提供可接受的支付。业务流程的结束了规则测试或申请以增加购买所需的费用。如果应用程序通过了业务规则测试，银行的批准者将审核申请并批准或拒绝贷款。

先决条件

- KIE 服务器已部署并连接到 Business Central。
- 部署了 Mortgage_Process 应用程序。
- 任务处理的用户是以下组和角色的成员：
  - approver 组： 用于 Qualify 任务
  - broker 组： 用于检测数据 并增加故障支付 任务
  - Manager 角色：对于 最后批准 任务

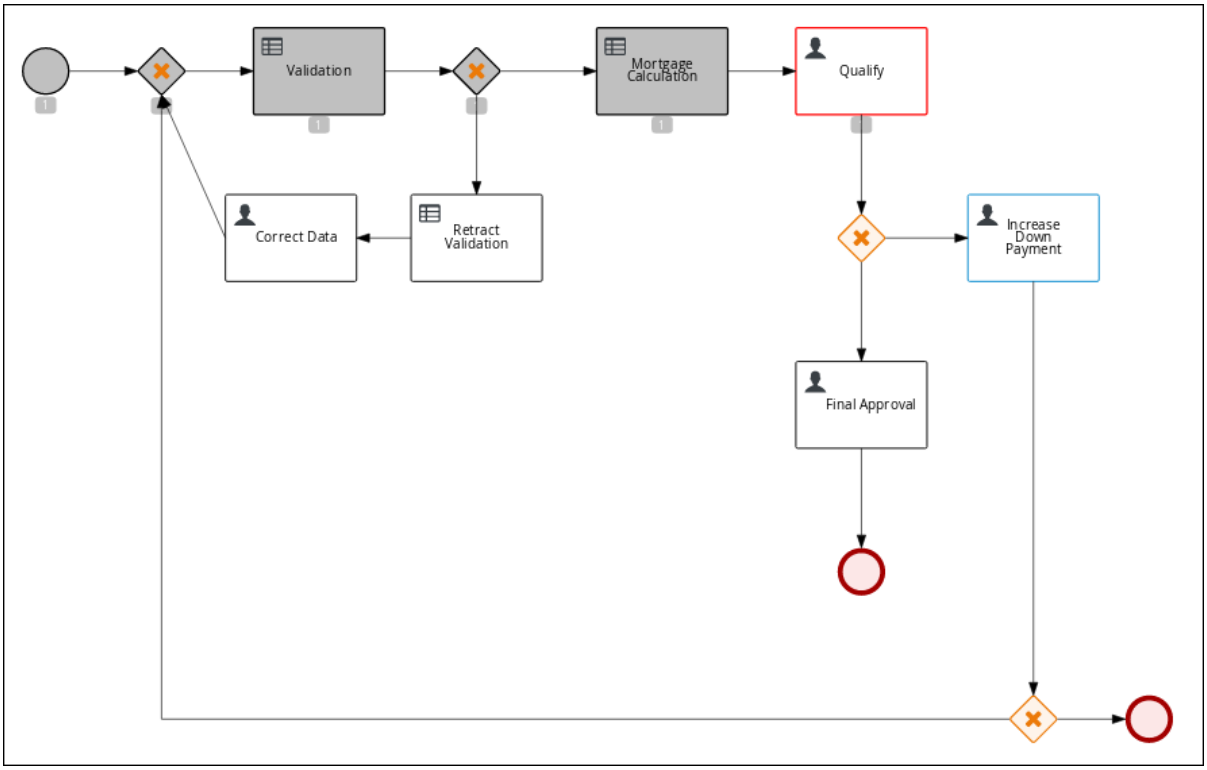
流程

1. 以 Bill （代理）登录 Red Hat Process Automation Manager，再点击 Menu → Manage → Process Definitions。
2. 点击 Actions 列中的三个垂直点，然后选择 Start 以开始打开 应用程序 表单，并在表单中输入以下值：
   - 停机支付 ：30000
   - 投影特年 ：10

- 名称 ：Ivo
- 年度 Income:60000
- SSN:123456789
- 属性的年龄 ：8
- 属性的地址 ：Brno
- locale:Rural
- 班加法案 ：50000

3. 单击 Submit 以启动新进程实例。启动进程实例后，Instance Details 视图将打开。

4. 单击 图表 选项卡，以查看进程图表中的进程流。在各个任务移动过程中，会突出显示进程的状态。



5. **从 Business Central 注销，然后以 Katy 重新登录。**

6. **点 Menu → Track → Task inbox。这将把您带到 Qualify 表格。**

7. **单击 Actions 列中的三个垂直点，然后选择并单击 Claim。Qualify 任务 Status 现在 保留。**

8. **单击 Qualify 任务行以打开并检查任务信息。点 Claim，然后在表单的底部开始。**

   应用程序表单现在可以激活批准或拒绝。

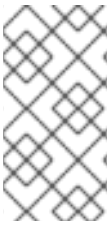
9. **要批准应用程序，可选择 是否限制应用程序？ 点击 Complete。**

10. **在 Task Inbox 中，点击 Final Approval row 的任意位置打开 最终批准 任务。**

11. **在 Final Approval 行中，点击 Actions 列中的三个垂直点，然后点击 Claim。**

12. **点击 最终批准 行中的任意位置打开 最终批准 任务。点表单底部的 Start。**

13. 请注意，选择了 Inlimit 复选框来反映该应用程序已准备好进行最终批准。点 Complete。



注意

Save 和 Release 按钮仅用于暂停批准过程，并在等待某个字段值时保存实例，或者释放另一个用户修改的任务。

第 22 章 监控 MORTGAGEAPPROVALPROCESS 进程应用程序

下面的章节显示，不同的银行员工（如系统管理员或知识工作）如何使用一些监控功能来跟踪制约批准过程的实例。

先决条件

- KIE 服务器已部署并连接到 Business Central。

流程

1. 登录 Red Hat Process Automation Manager，再点击 Menu → Manage → Process Instances。
2. 在 Manage Process Instances 窗口中，您可以设置过滤器，如 State、Errors、Id 等等。
3. 选择 State 过滤器中的 Completed，以查看所有完成的 MortgageApprovalProcess 实例。
4. 单击完成的 进程实例。
5. 点击以下每个标签页，获得用于监控特定进程实例可以使用哪些类型的信息：
   - 实例详情
   - 进程变量
   - 文档
   - 日志
   - 图

6. 点 Menu → Track → Process Reports。此视图包含各种图表，可帮助高级进程管理器获得基于 Type、开始日期、运行时间 等等的所有进程的概述，以帮助进行任务报告。

22.1. 使用 DEFAULT 或 ADVANCED 过滤器过滤进程实例

Business Central 现在为您提供默认和高级过滤器，以帮助您通过运行的进程实例过滤和搜索。您还可以使用 Advanced Filters 选项创建自定义过滤器。

22.1.1. 使用默认过滤器过滤进程实例

根据 状态、错误、过滤器、名称、开始日期 和 Last update 等属性过滤 进程实例。

流程

1. 在 Business Central 中，前往 Menu → Manage → Process Instances。
2. 在 Manage Process Instances 页面上，单击页面左侧的过滤器图标，以展开 Filters 窗格。

   此窗格列出了可用于过滤进程实例的以下进程属性：

   - State ：根据其状态过滤进程实例（活动、Aborted、Completed、Pending 和 Suspended）。
   - 错误 ：根据错误过滤进程实例。
   - 过滤符 ：根据 Id、Initiator、关联密钥或 Description 属性过滤进程实例。
     i. 选择所需属性。
     ii. 在下面的文本字段中输入搜索查询。

iii. 点应用。

- 名称 ：按定义名称过滤进程实例。
- 定义 Id ：按进程定义 ID 过滤进程实例。
- 部署 Id ：按进程部署 ID 过滤进程实例。
- 父进程实例 Id ：按父进程实例 ID 过滤进程实例。
- SLA 合规性 ：按 SLA 合规状态过滤进程实例。
- 开始日期 ：按创建日期过滤进程实例。
- 最后更新 ：按上次修改的日期过滤进程实例。

22.1.2. 使用高级过滤器过滤进程实例

使用 Advanced Filters 选项创建自定义进程实例过滤器。新创建的自定义过滤器添加到 Saved Filters 窗格中，可通过单击 Manage Process Instances 页面左侧的星号图标来访问。

流程

1. 在 Business Central 中，前往 Menu → Manage → Process Instances。
2. 在 Manage Process Instances 页面上，单击页面左侧的 Advanced Filters 图标。
3. 在 Advanced Filters 窗格中，输入过滤器的名称和描述，然后单击 Add New。
4. 从 Select 列 下拉列表中选择一个属性，如 processName。processName != value1 下拉列表更改的内容。

5. 再次点击下拉菜单并选择所需的逻辑查询。对于 **processName** 属性，选择 **等于**。

6. 将文本字段的值更改为要过滤的进程的名称。

注意

名称必须与项目业务流程中定义的值匹配。

7. 点 **Save**，并根据过滤器定义过滤进程。

8. 单击星号图标，以打开 **Saved Filters** 窗格。

在 **Saved Filters** 窗格中，您可以查看所有保存的高级过滤器。

第 23 章 其他资源

- *使用 BPMN 模型设计业务流程*

部分 III. RED HAT PROCESS AUTOMATION MANAGER 中的问题单管理入门

作为商业规则和流程开发人员，您可以在 Business Central 中使用案例管理资产来创建无法预测和临时的临时案例流程。问题单工作者或进程管理员也可以将 Business Central 用于案例管理和执行。Red Hat Process Automation Manager 为示例项目提供了 Business Central 中的示例业务资产作为参考。本文档论述了如何根据 Business Central 中包含的 IT_Orders 示例项目创建和测试示例 IT 订购项目。

先决条件

- Red Hat JBoss Enterprise Application Platform 7.4 已安装。有关安装信息，请参阅 *Red Hat JBoss Enterprise Application Platform 7.4 安装指南*。
- Red Hat Process Automation Manager 由 KIE 服务器安装和配置。如需更多信息，请参阅在 *Red Hat JBoss EAP 7.4 上安装和配置 Red Hat Process Automation Manager*。
- Red Hat Process Automation Manager 正在运行，您可以使用 kie-server、用户、和 admin 角色登录到 Business Central。
- 您已查看了问题单管理 *指定和构建案例中的信息*。

第 24 章 回顾 IT_ORDERS 示例项目

在创建自己的案例管理项目之前，请查看 Business Central 中的现有 IT_Orders 示例案例管理项目。此示例项目包含预定义的案例管理资产，作为您自己的案例项目的参考。

重要

业务流程应用程序示例包括仅技术预览功能。红帽产品服务等级协议(SLA)不支持技术预览功能，且可能并不完善，不建议在生产环境中使用。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。有关红帽 技术预览支持的更多信息，请参阅技术预览功能支持范围。

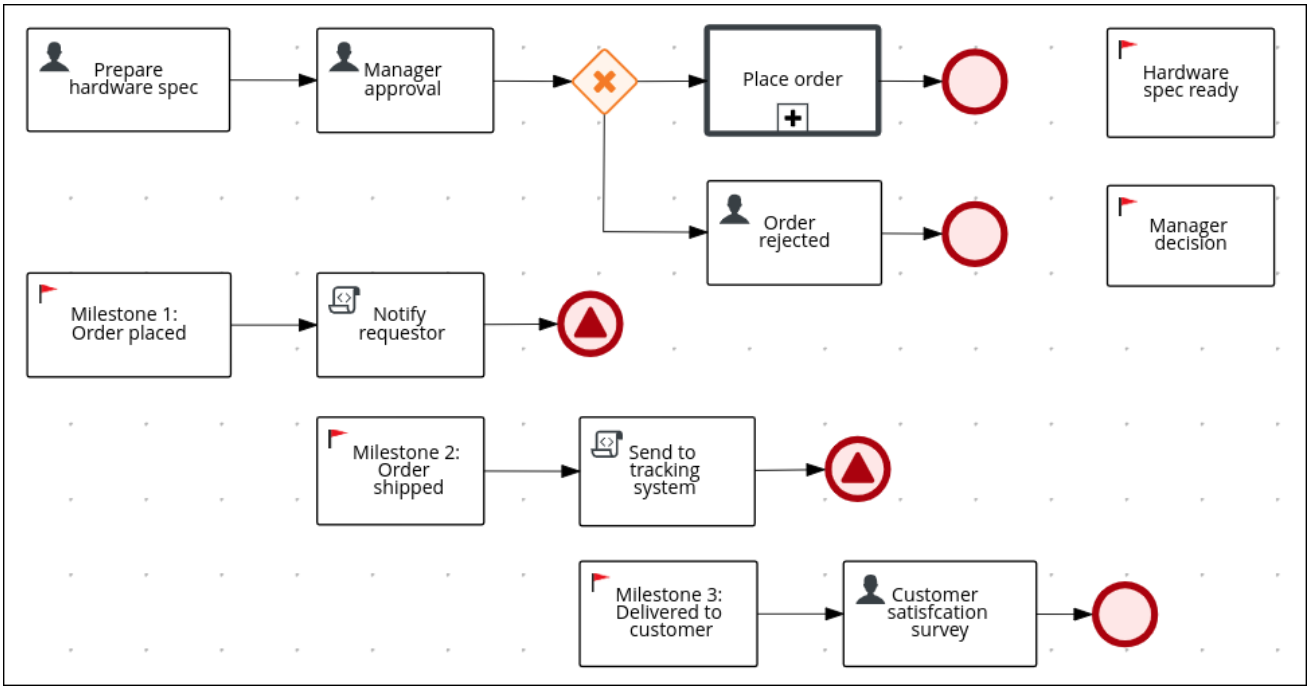
流程

1. 在 Business Central 中，转至 Menu → Design → Projects。如果已有项目，您可以点击 MySpace default 空间并从 Add Project 下拉菜单中选择 Try Samples 来访问示例。如果没有现有项目，请点击 Try samples。

2. 选择 IT_Orders 并点 确定。

项目 的资源视图将打开。选择每个示例资产来探索项目如何设计实现指定目标或工作流。

查看 orderhardware 业务流程，以帮助您了解业务流程流。



第 25 章 创建新的 IT_ORDERS 案例项目

在 Business Central 中创建一个新的 IT_Orders 项目，了解所有需要的资产及其在项目中使用的方式。

流程

1. 登录到 Business Central，再转到 Menu → Design → Project。

   Business Central 提供名为 MySpace 的默认空间。您可以使用默认空间来创建和测试示例项目。

2. 点击 Add Project 下拉菜单并选择 Case 项目 选项：

3. 在 Add Project 窗口中，在 Name 字段中输入 IT_Orders_New，然后输入 项目描述。

4. 单击 Add 以添加项目。

   项目 的资源视图将打开。

第 26 章 数据对象

数据对象是您创建的规则资产的构建块。数据对象是在项目指定软件包中作为 Java 对象实施的自定义数据类型。例如，您可以创建一个带有数据字段、address 和 DateOfBirth 的 Person 对象，以指定 loan Application 规则的个人详情。 这些自定义数据类型决定了您的资产和您的决策服务所基于的数据。

26.1. 创建 ITORDERSERVICE 数据对象

ITOrderService 数据对象指定用来定义 IT 顺序变量的数据类型。

先决条件

- 创建 IT_Orders_New 项目。

流程

1. 点 Add Asset → Data Object。
2. 在 Create new Data Object 向导中输入以下值：
   - 数据对象： ITOrderService
   - Package: com.myspace.it_orders_new
3. 点 确定。
4. 点击 Package 下拉菜单旁的
   
   为数据对象指定新软件包。
5. 输入 org.jbpm.demo.it_orders.services，然后点 Add。

6. 单击 **Save**，然后单击 **Yes, Move** 以确认您的更改。

26.2. 创建 SURVEY 数据对象

Survey data 对象包含数据字段，如 **deliveredOnTime** 和 **missingEquipment**。在设计问题单时，您将使用数据和值。

先决条件

- 创建 **IT_Orders_New** 项目。

流程

1. 在 **Business Central** 中，转至 **Menu** → **Design** → **Projects** 并点击 **IT_Orders_New**。
2. 点 **Add Asset** → **Data Object**。
3. 在 **Create new Data Object** 向导中输入以下值：
   - 数据对象 ：问卷调查
   - 软件包:com.myspace.it_orders_new
4. 点 确定。
5. 添加 **Survey** 数据对象约束。
   a. 点 **add** 字段。
   b. 输入以下值：

- **ID:注释**
- **标签: Leave empty**
- **键入:String**

c. **点 Create 并继续，然后输入以下值：**

- **ID:deliveredOnTime**
- **标签: Leave empty**
- **类型 ：布尔值**

d. **点 Create 并继续，然后输入以下值：**

- **id:missingEquipment**
- **标签: Leave empty**
- **键入:String**

e. **点 Create 并继续，然后输入以下值：**

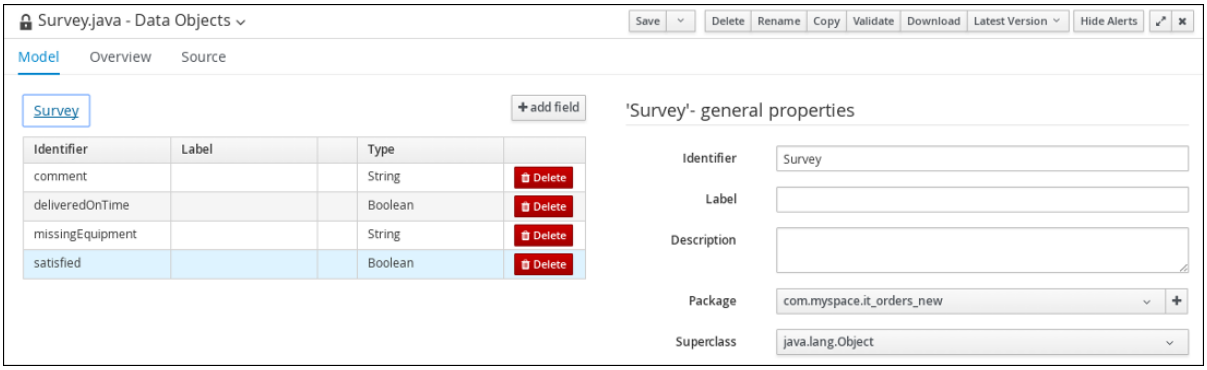
- **ID ：满意**
- **标签: Leave empty**

- 类型 ：布尔值

f. 点 Create。

6. 单击 Save 以确认更改。

图 26.1. 问卷调查数据对象详情

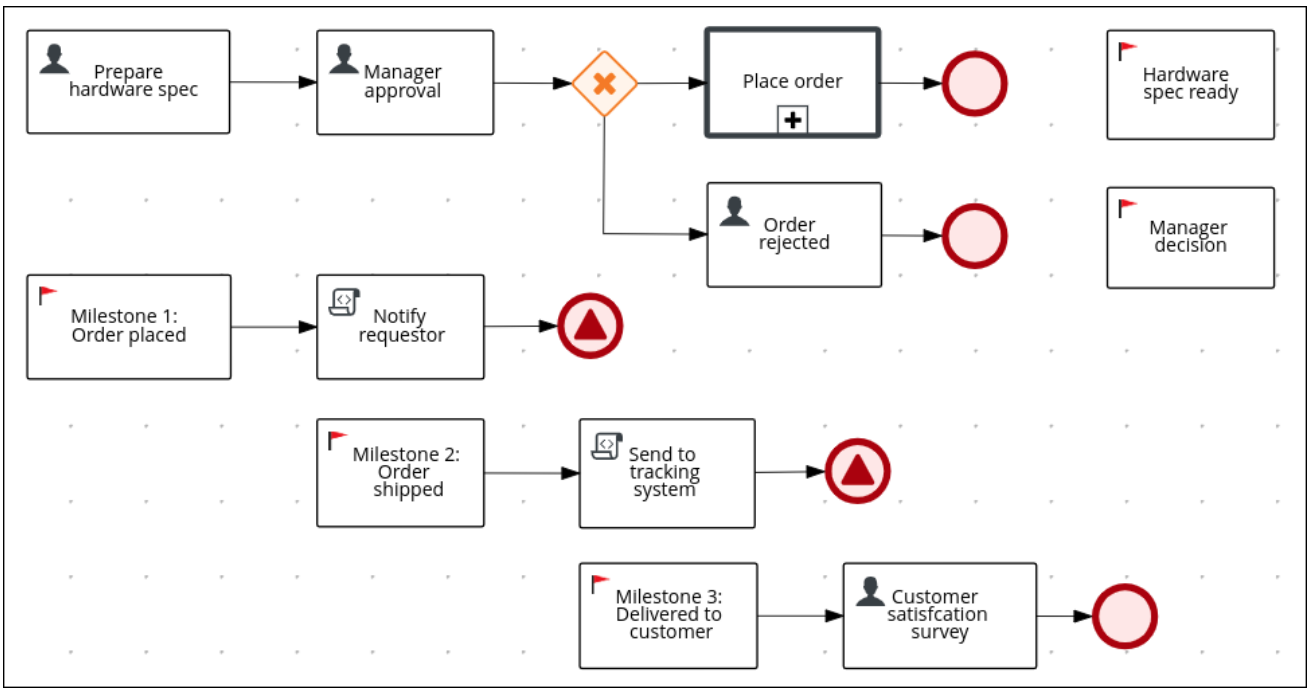


第 27 章 设计问题单定义

您在 Business Central 中使用流程设计器的设计案例。案例设计是案例管理的基础，可为每个案例设置具体的目标和任务。可以通过添加动态任务或进程，在运行时动态修改案例流程。在这一流程中，您将创建这种相同的问题单定义，以熟悉问题单定义设计过程。

Business Central 中的 IT_Orders 示例项目包括以下 orderhardware 业务流程案例定义。

图 27.1. 订单硬件业务流程案例定义



先决条件

- 您已在 Business Central 中创建了一个新问题单。更多信息请参阅 第 25 章 *创建新的 IT_Orders 案例项目*。
- 您已创建了数据对象。更多信息请参阅 第 26 章 *数据对象*。

流程

1. 在 Business Central 中，转至 Menu → Design → Projects 并点击 IT_Orders_New。
2. 点击 Add Asset → Case Definition。

3. 在 Create new Case 定义 窗口中，添加以下所需信息：

- 案例定义 ：输入 顺序硬件.这通常是正被管理的案例或项目的主题。
- 软件包 ：选择 com.myspace.it_orders_new 以指定创建案例文件的位置。

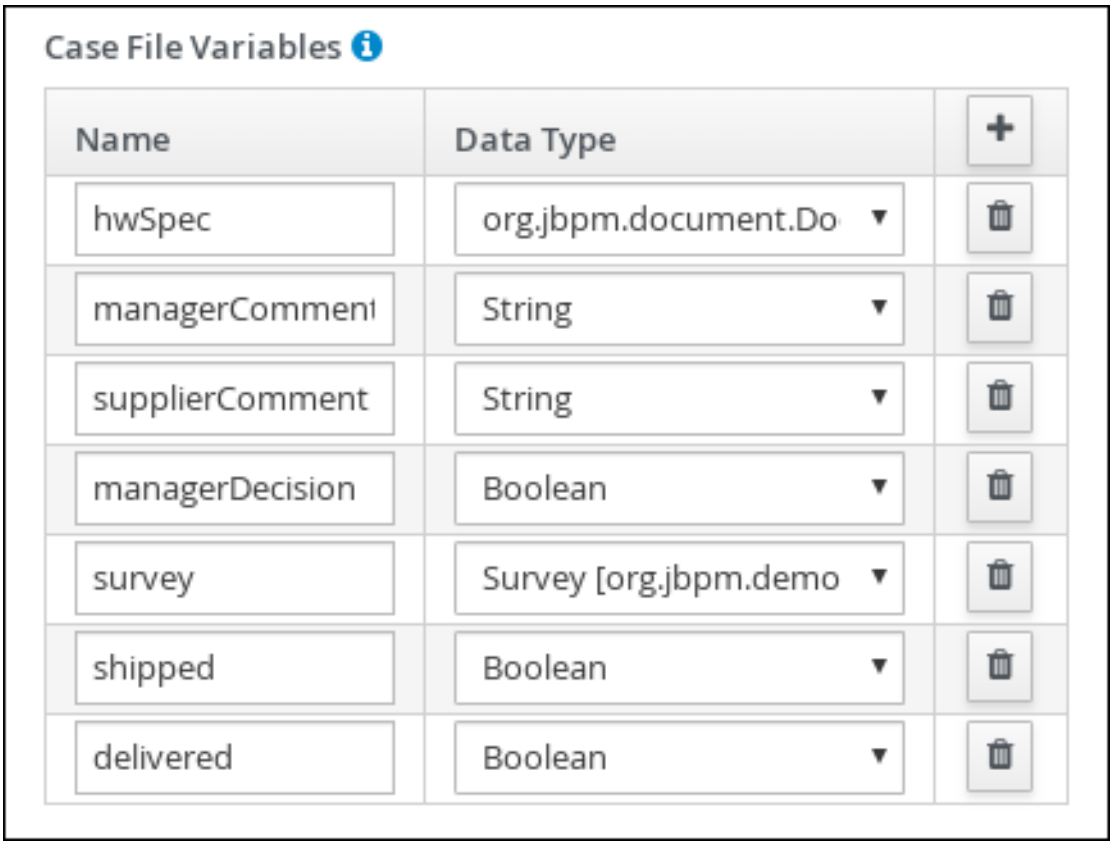
4. 单击 Ok 以打开进程设计程序。

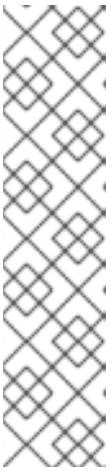
5. 定义案例文件变量的值，这些变量可供使用的子进程、子案例和业务规则访问。

a. 在右上角点击 Properties  图标。

b. 向下滚动并展开 Case Management，点 Case File Variables 部分中的  ，并输入以下内容：

图 27.2. orderhardware case file 变量





注意

以下示例文件变量是自定义数据类型：

- **hwSpec: org.jbpm.document.Document（此值中的类型）**
- 调查 ：调查 **[com.myspace.it_orders_new]**（选择这个值）

6. **点击 Save。**

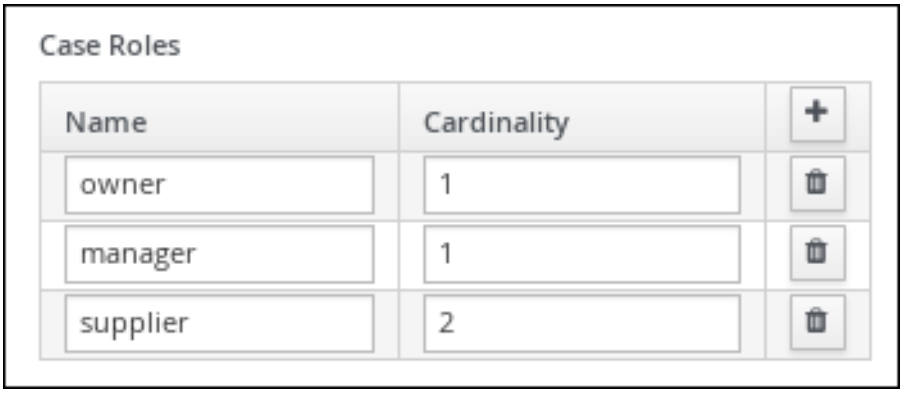
7. **定义案例中涉及的角色。**

   a. **在右上角点击 Properties**  **图标。**

   b. **向下滚动并展开 Case Management，点 Case Roles 部分中的**

，并输入以下内容：

图 27.3. orderhardware case roles



- 所有者 ：提出硬件订单请求的员工。role cardinality 设置为 1，这表示只能将一个人或组分配给此角色。
- 管理器 ：员工经理；将批准或拒绝请求硬件的人员。role cardinality 设置为 1，这表示只能将一个人或组分配给此角色。
- 供应商 ：系统中 IT 硬件可用的供应商。角色卡性设置为 2，这意味着可以为这个角色分配多个供应商。

8. 点击 Save。

27.1. 创建 PLACE ORDER 子进程

创建 Place order -process，它是一个单独的业务流程，由供应商执行。这是在执行问题单执行过程中的可重复使用的过程，如 第 27 章 *设计问题单定义* 所述。

先决条件

- 您已在 Business Central 中创建了一个新问题单。更多信息请参阅 第 25 章 *创建新的 IT_Orders 案例项目*。
- 您已创建了数据对象。更多信息请参阅 第 26 章 *数据对象*。

流程

1. **在 Business Central 中，前往 Menu → Design → Projects → IT_Orders_New。**

2. **在项目菜单中点击 Add Asset → Business Process。**

3. **在 Create new Business Process 向导中输入以下值：**

   - **业务流程 ：订购位置**
   - **软件包: 选择 com.myspace.it_orders_new**

4. **点 确定。图表编辑器将打开。**

5. **点击 canvas 中的空空间，在右上角点 Properties**  **图标。**

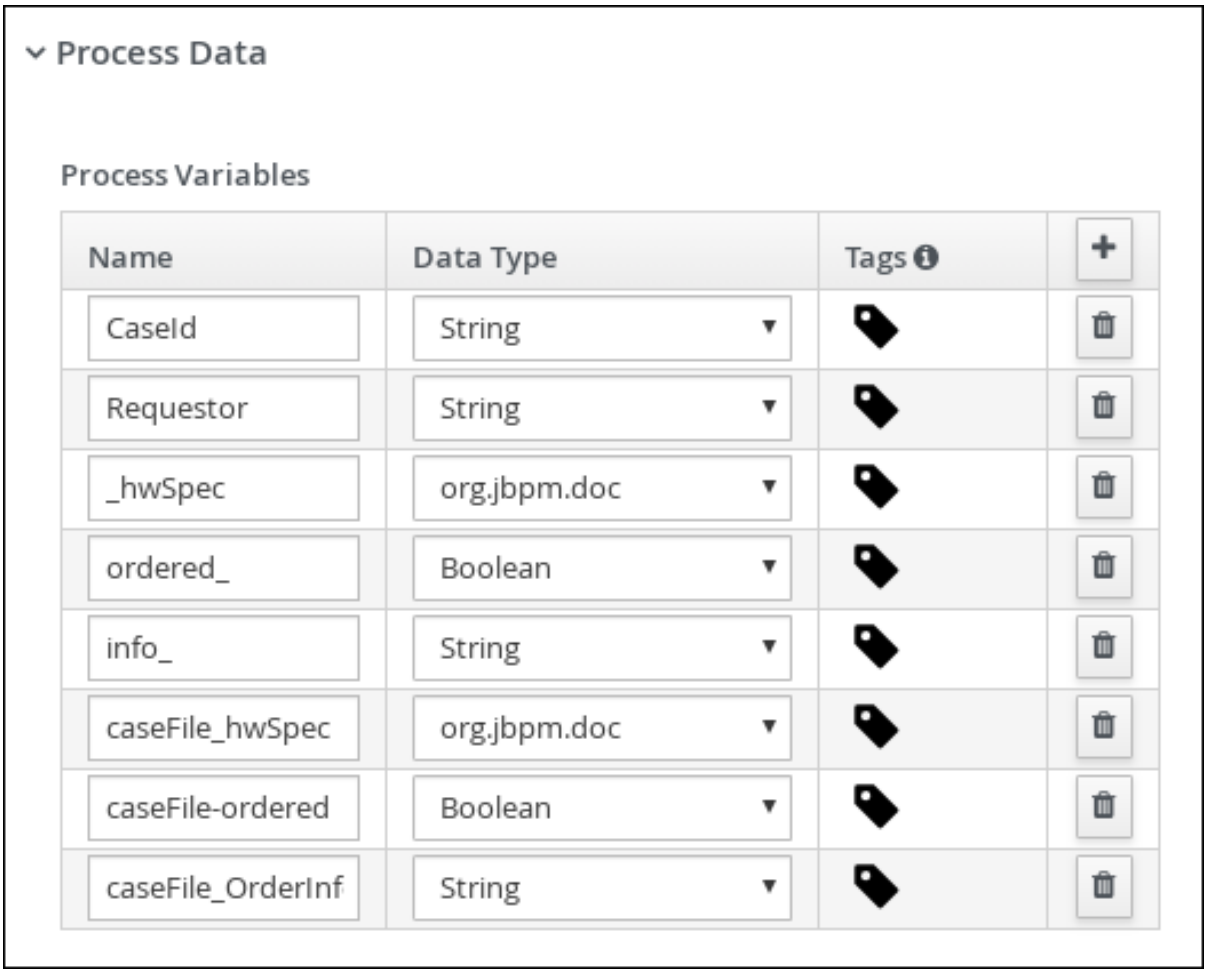
6. **向下滚动，展开 Process Data，点 Process Variables 部分中的**  **，然后在 Process Variables 下输入以下值：**

表 27.1. 进程变量

Name	数据类型
CaseID	字符串
requestor	字符串
_hwSpec	org.jbm.doc
ordered_	布尔值
info_	字符串
caseFile_hwSpec	org.jbm.doc

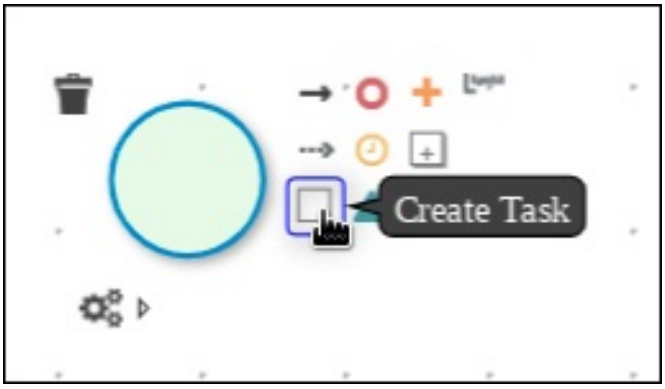
Name	数据类型
caseFile-ordered	布尔值
caseFile-orderinf	字符串

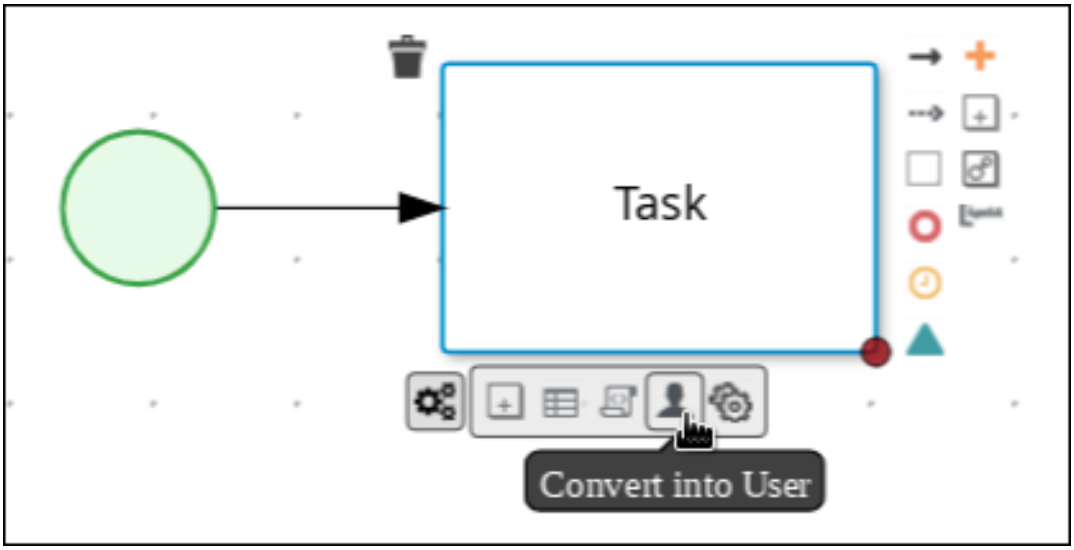
图 27.4. 完成的进程变量



7. **点击 Save。**

8. **将启动事件拖到 canvas 上，从启动事件创建传出连接到任务，并将新任务转换为用户任务。**





9. 点用户任务并在 **Properties** 面板中点 **Name** 字段输入 **Place order**。

10. 展开 **Implementation/Execution**，点 **Groups** 菜单下的 **Add**，点 **Select → New**，以及输入 供应商。

11. 在 **Assignments** 字段中点

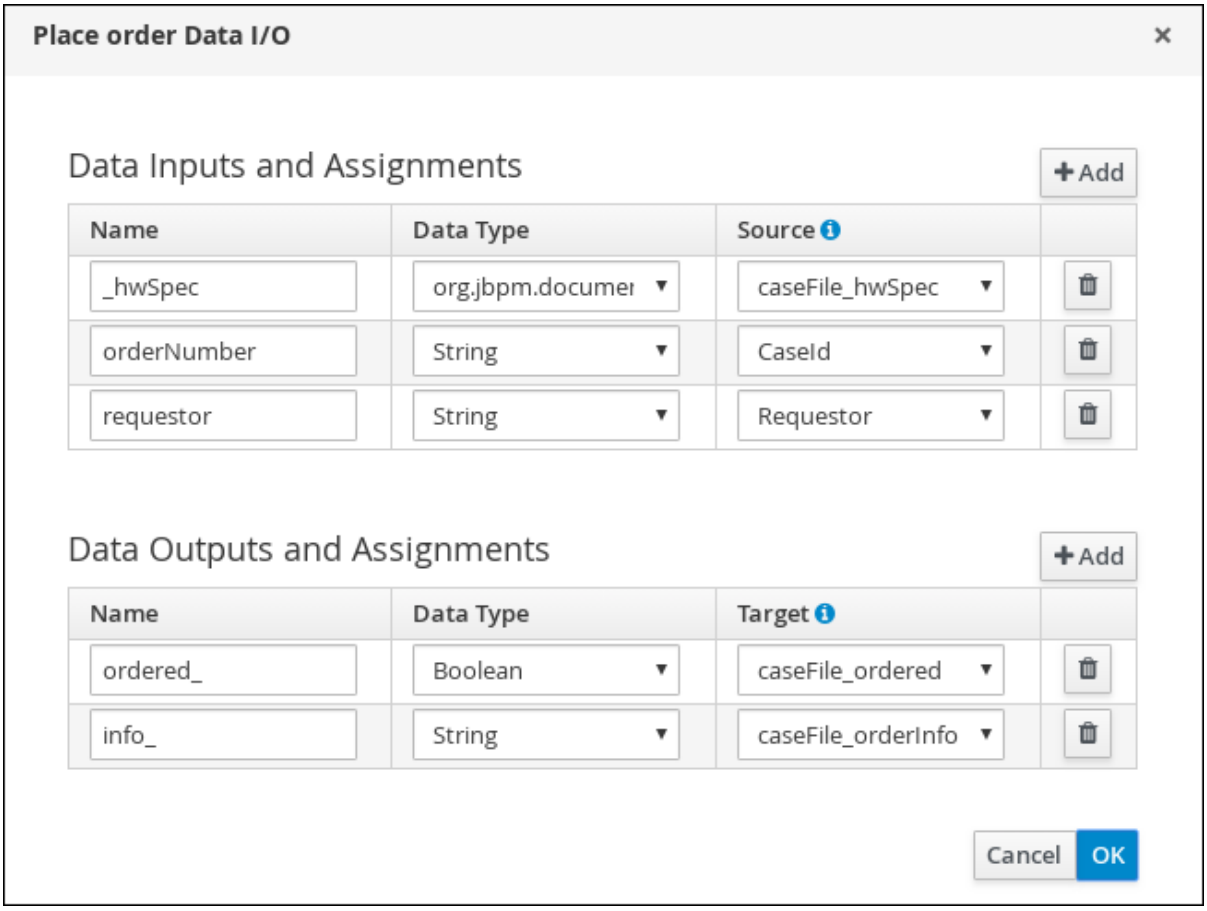
，并在 **Place order Data I/O** 对话框中添加以下数据输入和输出：

表 27.2. 数据输入和分配

Name	数据类型	源
_hwSpec	org.jbpm.document	caseFile_hwSpec
orderNumber	字符串	CaseId
requestor	字符串	requestor

表 27.3. 数据输出和分配

Name	数据类型	目标
ordered_	布尔值	caseFile_ordered
info_	字符串	CaseFile_orderInfo



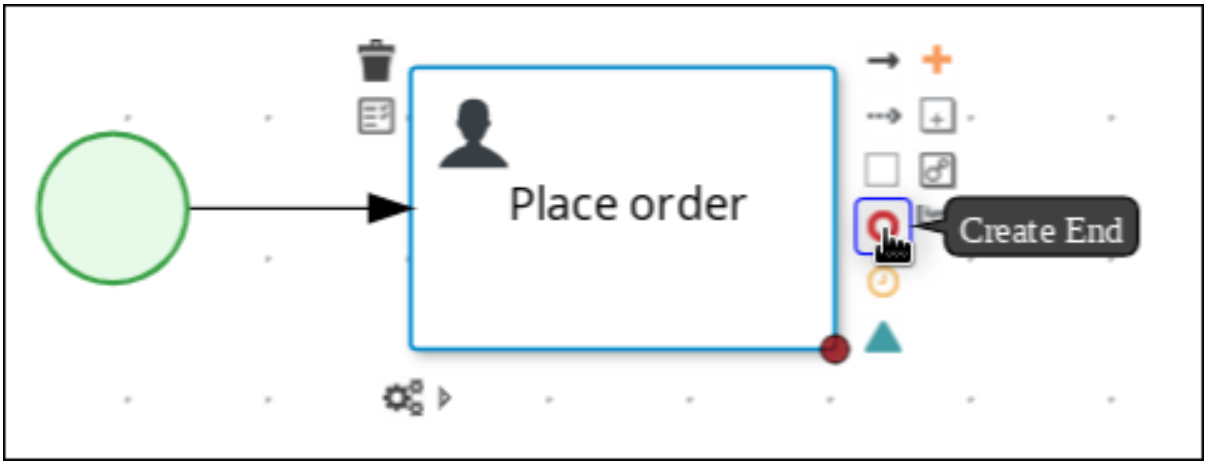
对于第一个输入分配，选择 Custom 作为 Data Type 和 input org.jbpm.document.Document。

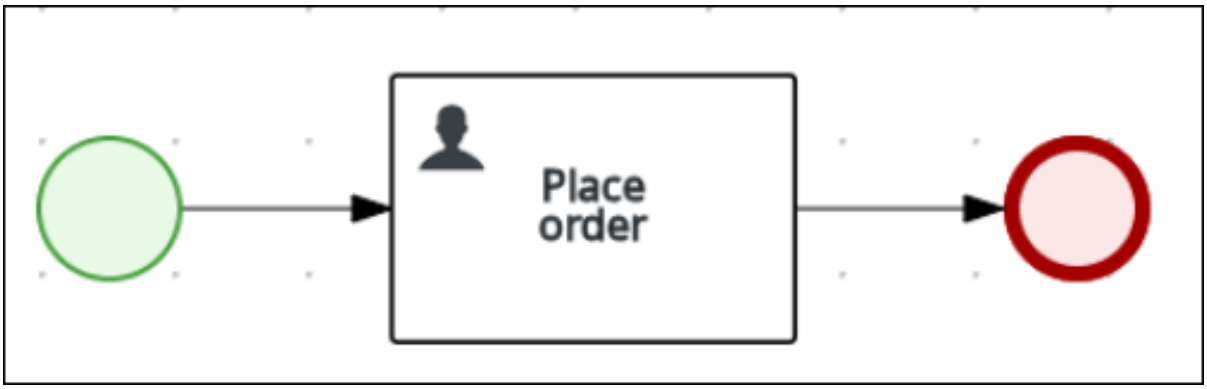
12. **点击 确定。**

13. **选中 Skippable 复选框，并在 Description 字段中输入以下文本：**

    批准顺序 #{CaseId} 要放置

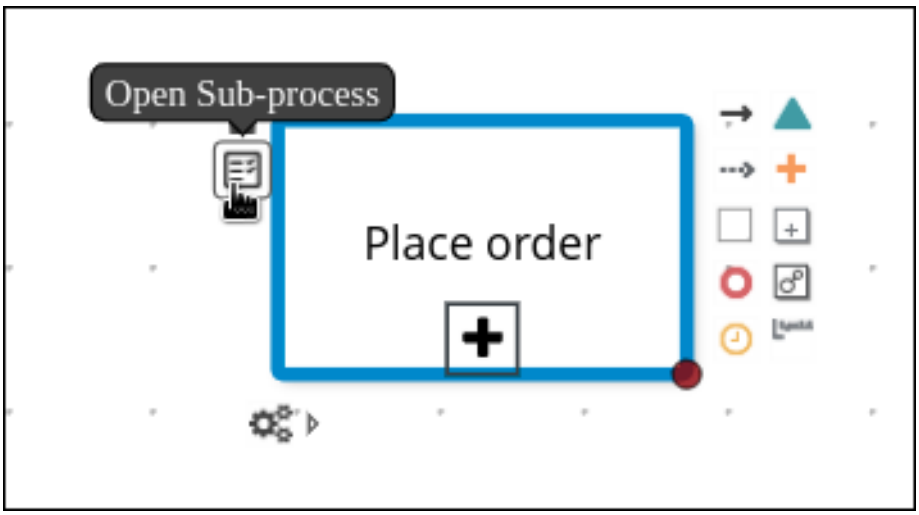
14. **从 Place order 用户任务创建传出连接，并将其连接到最终事件。**





15. 单击 **Save** 以确认更改。

您可以通过单击主进程中的 **Place order** 任务，然后单击 **Open Sub-process** 任务图标，在 **Business Central** 的新编辑器中打开子进程。



27.2. 创建管理器批准业务流程

管理器批准过程确定是否将放置或拒绝订购。

流程

1. 在 Business Central 中，转至 Menu → Design → Projects → IT_Orders_New → orderhardware Business Process。

2. 创建并配置 Prepare hardware spec 用户任务：

   a. 在 Object Library 中展开任务，并将用户任务拖到 canvas 上，并将新任务转换为用户任务。

   b. 点新用户任务并点击右上角的 Properties  图标。

   c. 在 Name 字段中输入 Prepare hardware spec。

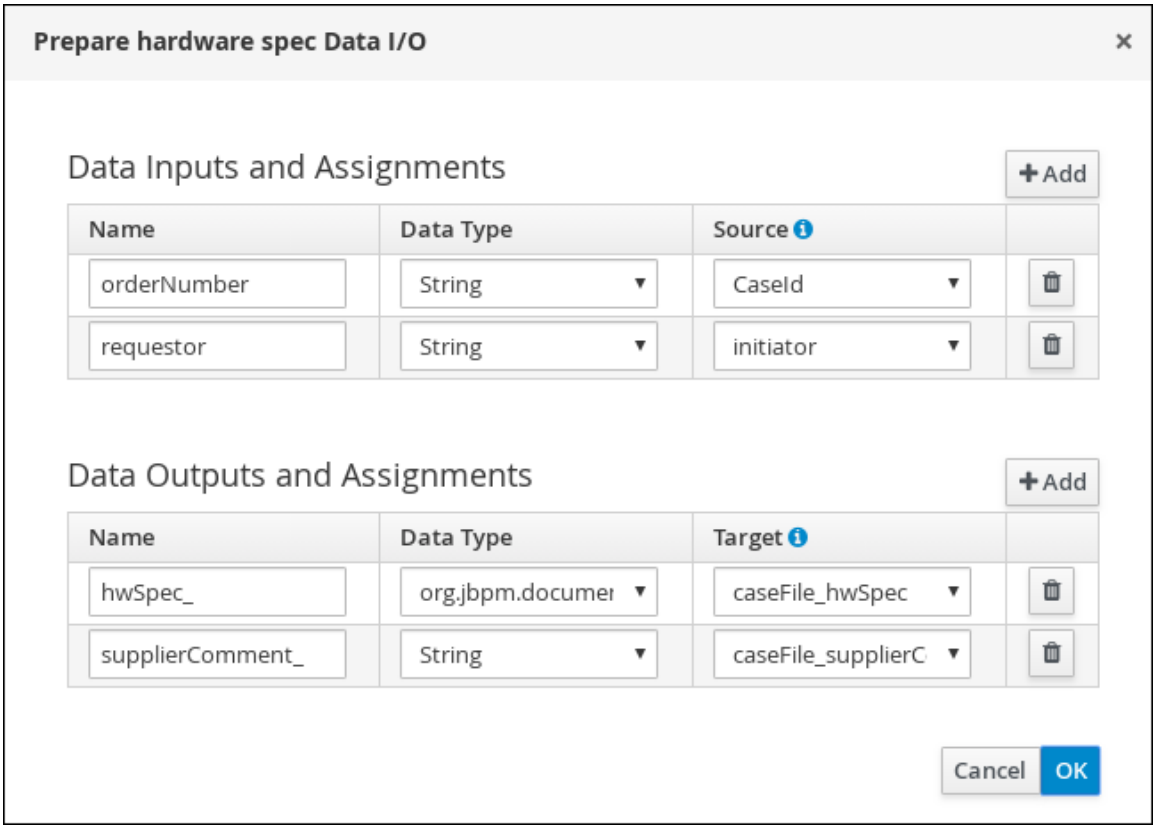
   d. 展开 Implementation/Execution，点 Groups 菜单下的 Add，点 Select → New，以及输入 供应商。

   e. 在 Task Name 字段中输入 PrepareHardwareSpec。

   f. 选中 Skippable 复选框，并在 Description 字段中输入以下文本：

      为 #{initiator}(order number #{CaseId})准备硬件规格.

   g. 在 Assignments 字段中点  并添加以下内容：



h. **点击 确定。**

3. **创建并配置 Manager 批准用户任务：**

a. **单击 Prepare hardware spec user 任务，再创建一个新用户任务。**

b. **点新用户任务并点击右上角的 Properties** 图标。

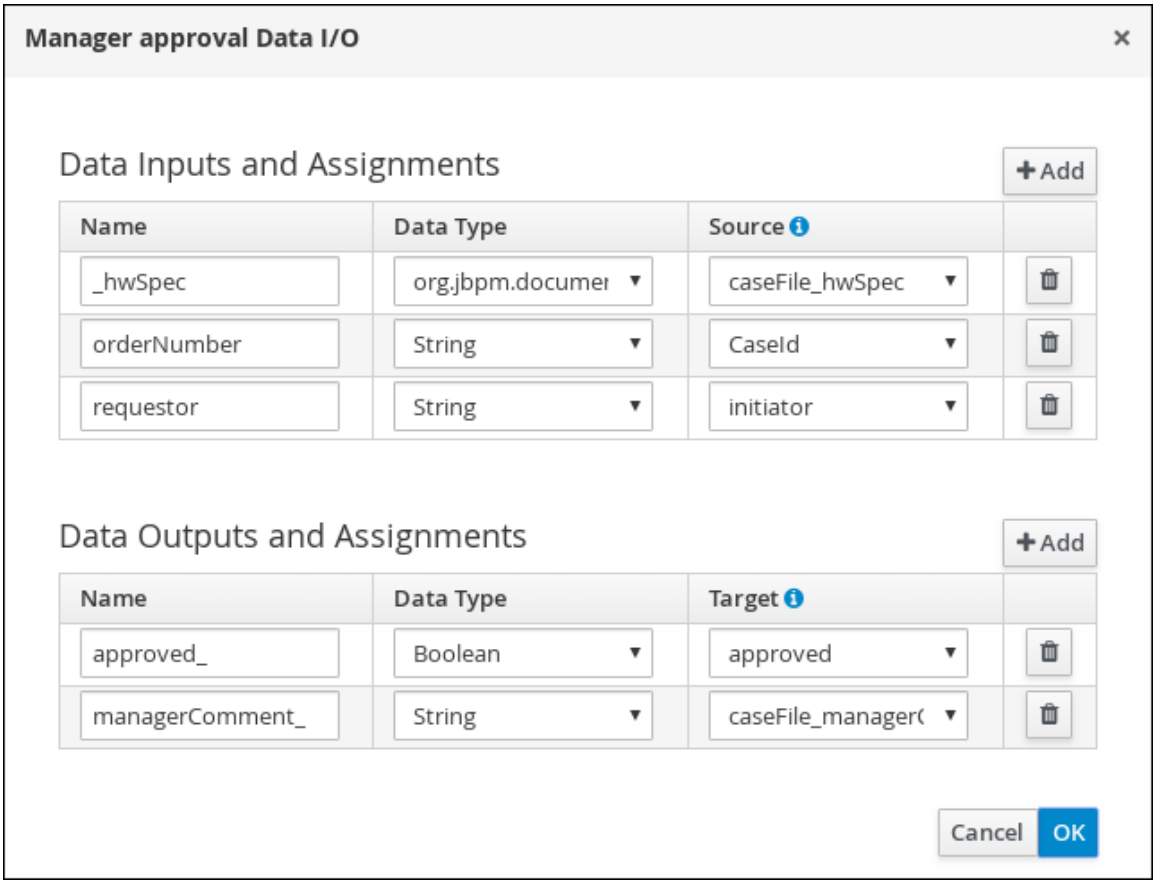
c. **在 Name 字段中，点用户任务并在 Properties 面板中的输入 Manager 批准。**

d. **展开 Implementation/Execution，点 Actors 菜单下的 Add，点 Select → New，以及输入 管理器。**

e. **Task Name 字段中的输入 ManagerApproval。**

f. **在 Assignments 字段中点**

并添加以下内容：



g. **点击 确定。**

h. **选中 Skippable 复选框，并在 Description 字段中输入以下文本：**

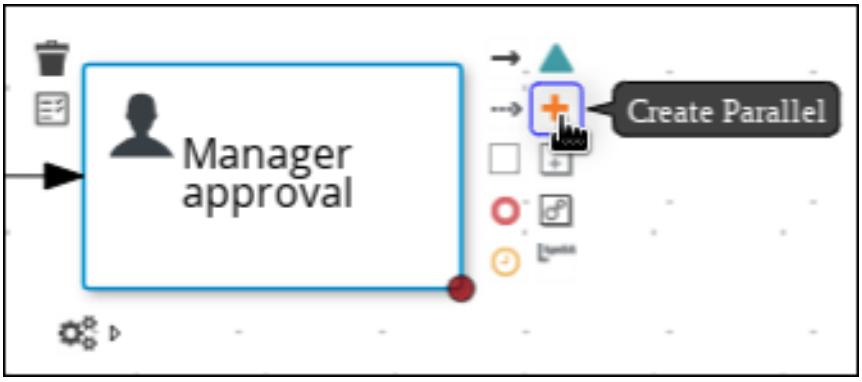
批准新硬件的 #{initiator}(order number #{CaseId})

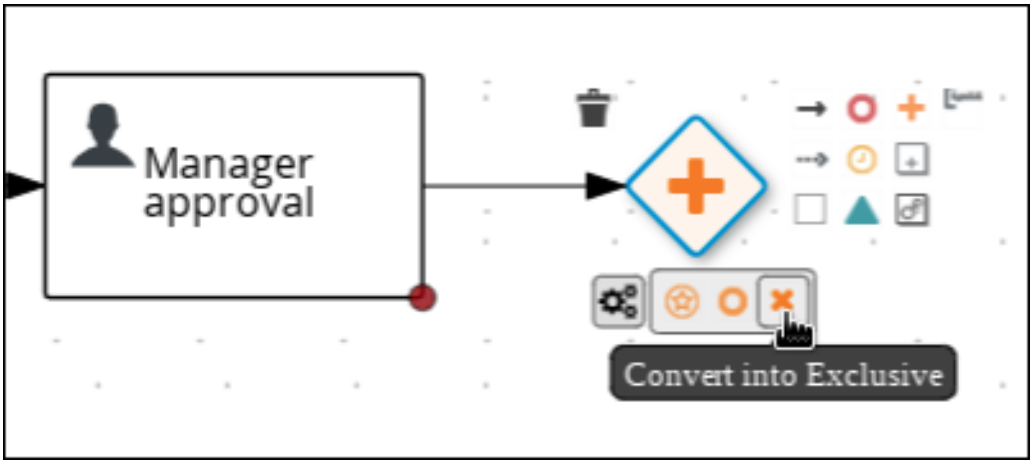
i. **在 On Exit Action 字段中输入以下 Java 表达式：**

```
kcontext.setVariable("caseFile_managerDecision", approved);
```

j. **点击 Save。**

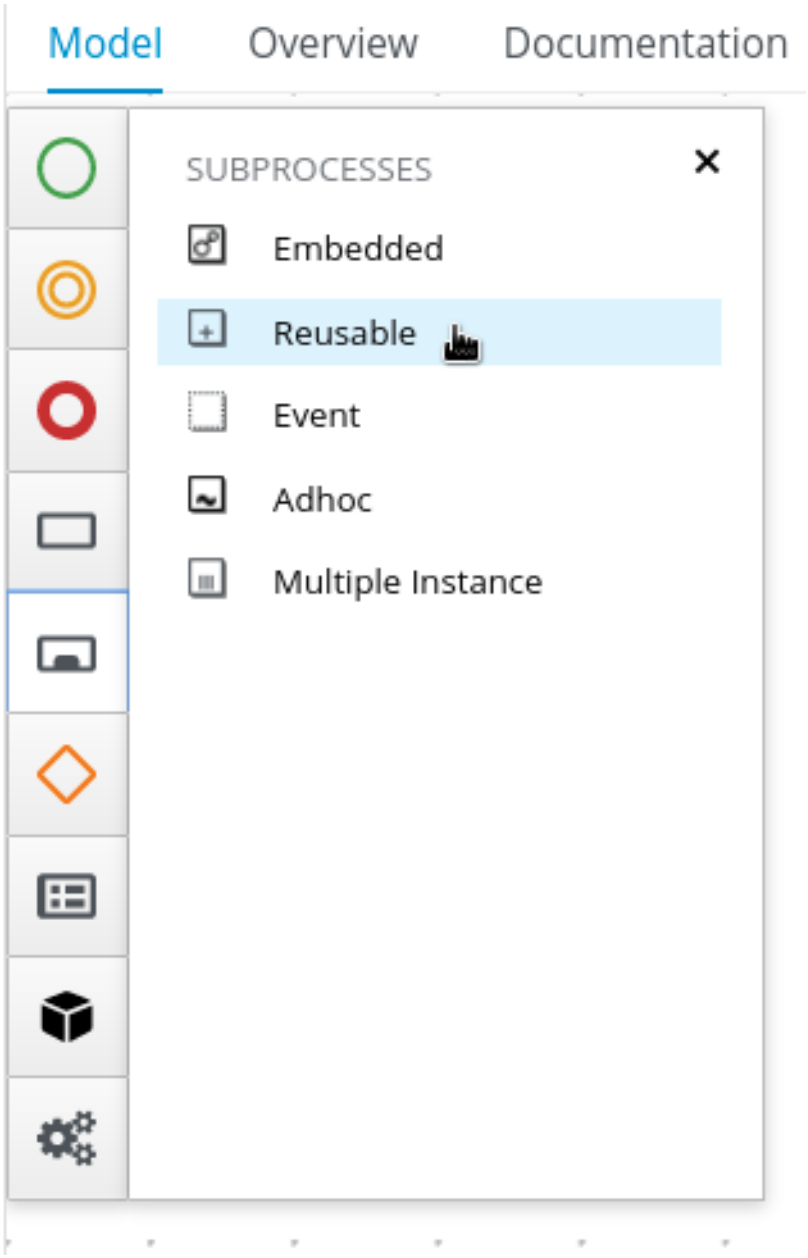
4. **点 Manager 批准 用户任务，创建基于 Data-clusive(XOR)网关。**





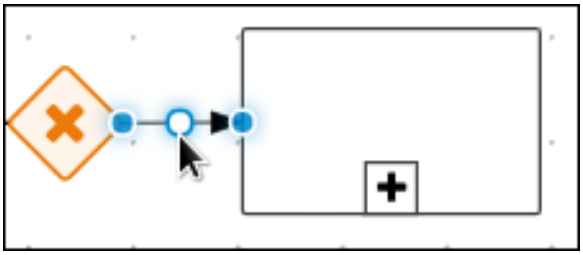
5. 创建并配置 **Place order reusable** 子进程：

   a. 从 对象库 展开 子进程，单击 **Reusable**，再将新元素拖到基于 **Data-clusive(XOR)网关的右侧。**



b.

将基于 Data-clusive(XOR)网关连接到子进程。



c.

点新的子任务，然后点击右上角的 Properties 图标。

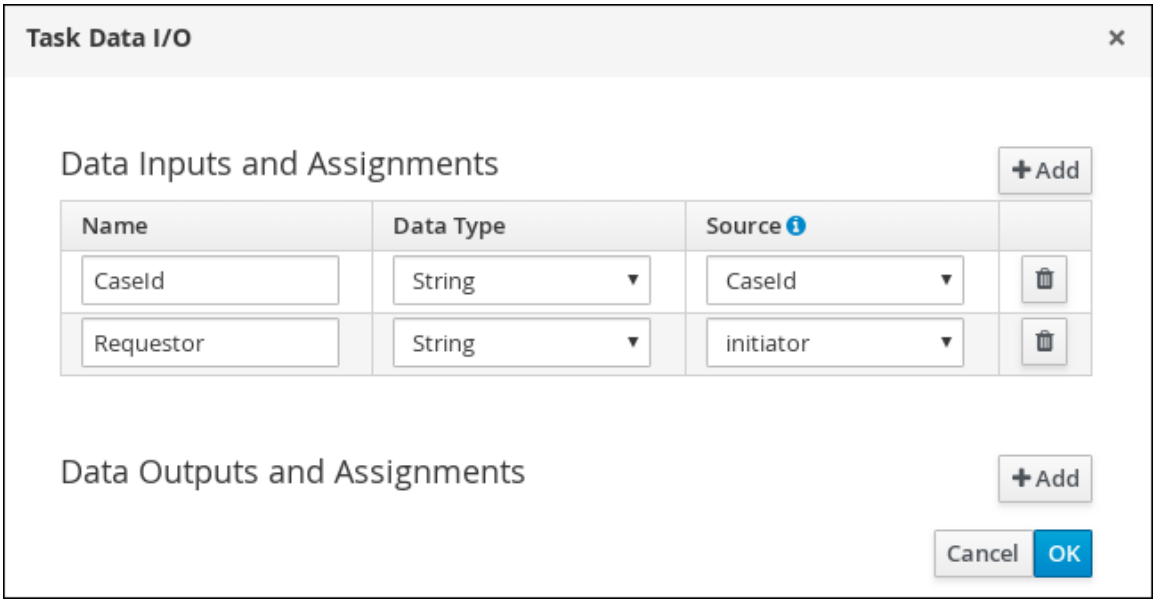
d.

Name 字段中的输入 Place order.

e.

展开 Data Assignments，点 Assignments 字段中的

并添加以下内容：



f.

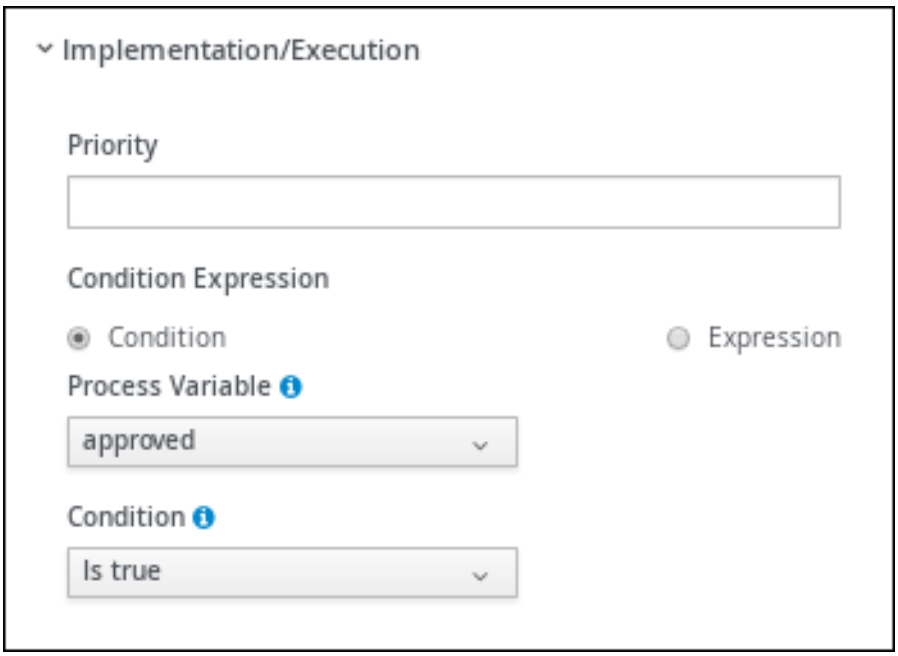
点击 确定。

g.

点基于 Data-based Exclusive (XOR)网关到子进程的连接，然后点击 Properties  **图标。**

h.

展开 Implementation/Execution，选择 Condition，并设置以下条件表达式。

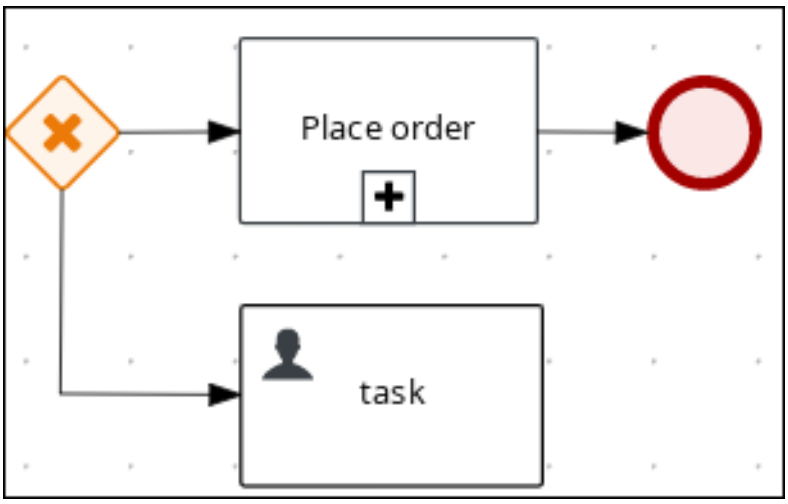


i. 单击 **Place order user** 任务，再创建最终事件。

6. 创建并配置拒绝用户任务的顺序：

a. 单击 **Data- Exclusive(XOR)**网关并创建新用户任务。

b. 拖动新任务，使其对齐在 **Place order** 任务之下。



c. 点新用户任务并点击右上角的 **Properties** 图标。

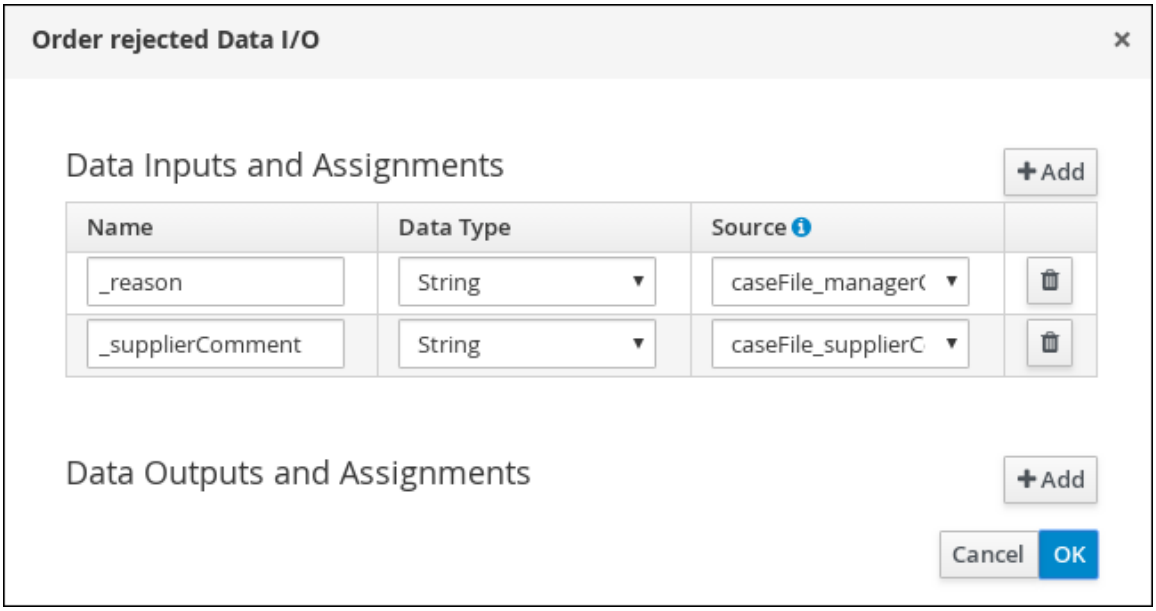
d. **Name** 字段中 拒绝的输入顺序。

e. 在 **Task Name** 字段中展开 实现/执行 和输入 顺序拒绝。

f. 点 **Actors** 菜单下的 **Add**，点 **Select** → **New**，以及输入 所有者。

g. 在 **Assignments** 字段中点

并添加以下内容：



h. **点击 确定。**

i. **选中 Skippable 复选框，并在 Description 字段中输入以下文本：**

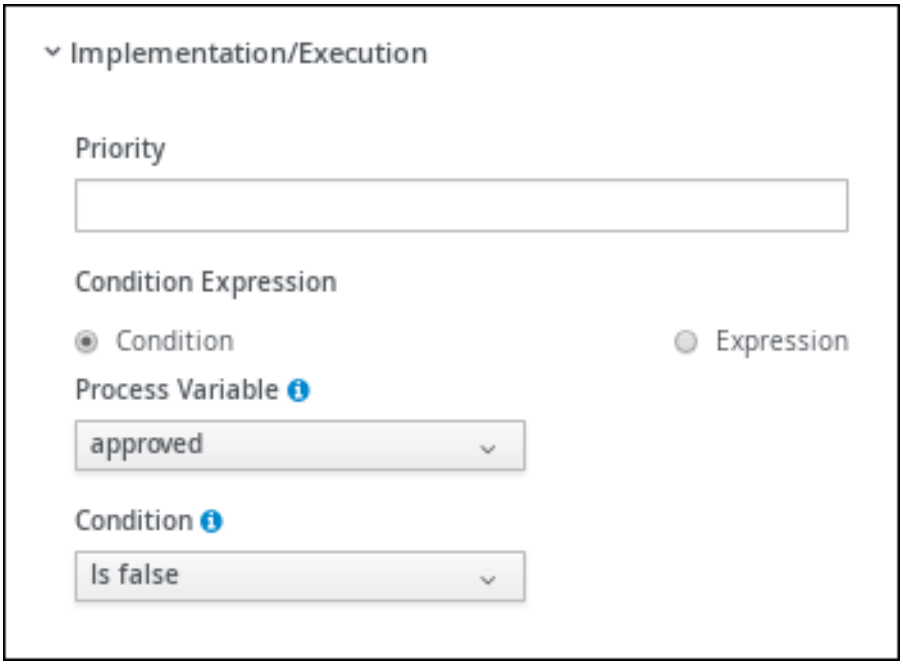
order #{CaseId} 已由 manager 拒绝

j. **单击 Order rejected user 任务，再创建一个最终事件。**

k. **点击 Save。**

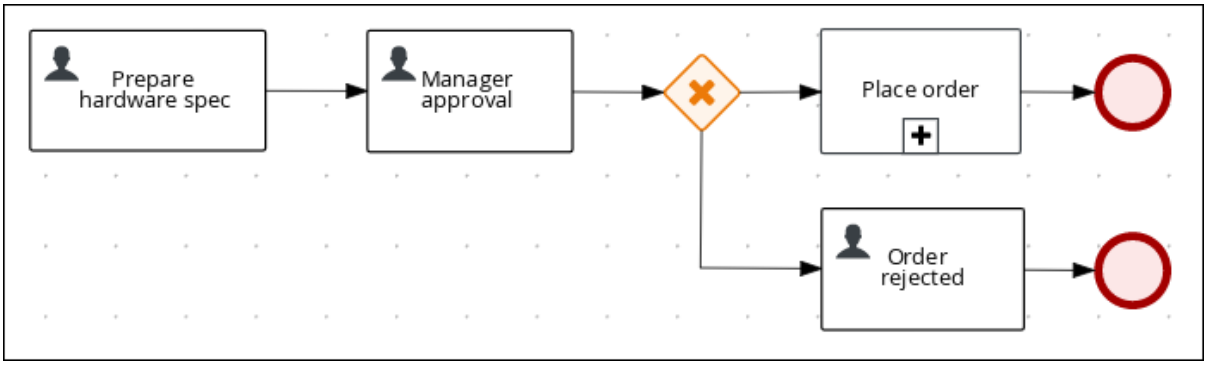
7. **点基于 Data 的 Exclusive (XOR)网关的连接到 Order rejected user 任务，然后点 Properties** 图标。

8. **展开 Implementation/Execution，选择 Condition，并设置以下条件表达式。**



9. 点击 **Save**。

图 27.5. Manager 批准业务流程



第 28 章 MILESTONES

milestones 是一个特殊的服务任务，可通过将 milestone 节点添加到进程设计器面板，在案例定义设计人员中配置。在创建新问题单定义时，默认包含在设计面板上作为 AdHoc Autostart 的 milestone。默认情况下，新创建的 milestones 默认设置为 AdHoc Autostart。

在阶段结束时，案例管理里程通常发生，但也可是实现其他里程干的结果。milestone 始终需要定义一个条件来跟踪进度。当数据添加到问题单时，milestones 对问题单文件数据做出反应。milestone 代表问题单实例内实现的一个点。它可用于标记某些事件，对于关键性能指示器(KPI)跟踪或识别仍然完成的任务很有用。

在执行时，milestones 可以处于以下任意状态：

- Active ：该条件已在里程one上定义了，但该条件尚未满足。
- 完成: 满足 milestone 条件，达到 milestone，且案例可以继续进行下一任务。
- terminated ：milestone 不再是问题单进程的一部分，因此不再需要。

虽然 milestone 可用或完成，但可以通过信号手动触发，或者在问题单实例启动时自动配置 AdHoc Autostart。可以根据需要多次触发 milestones，但在满足条件时直接实现。

28.1. 创建硬件规格就绪 MILESTONE

创建在完成所需硬件规格文档时访问的 HardwareSpecReady milestone。

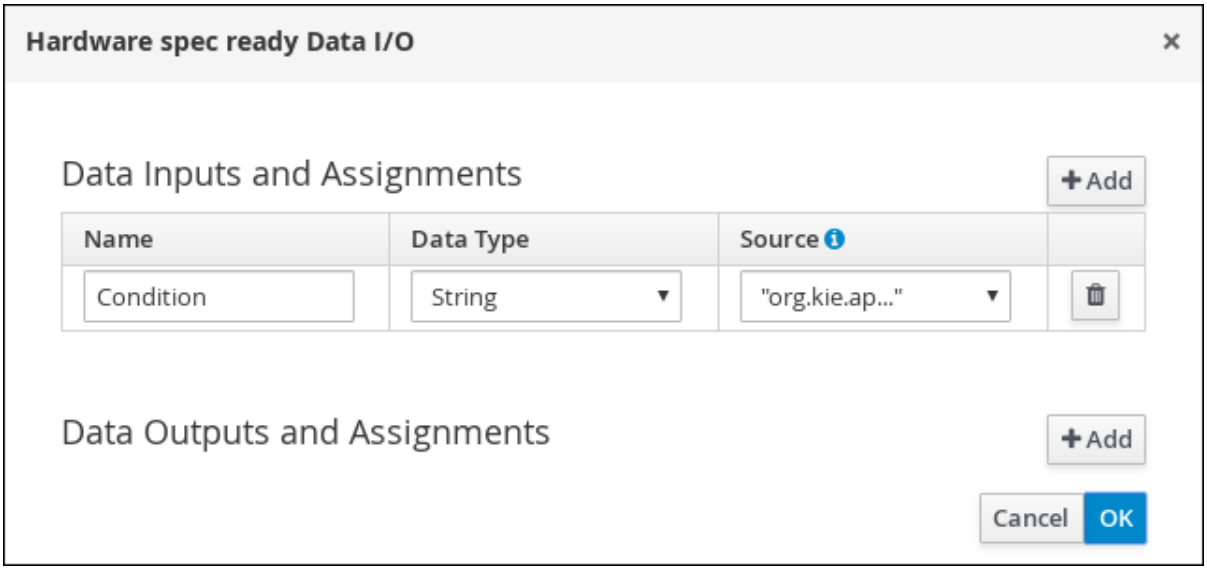
流程

1. 在流程设计器中，展开 Object Library 中的 Milestone，将新的里程one 拖放到 canvas 上，并将其放置在 Place order 结束事件的右侧。
2. 点新的 milestone，然后点击右上角的 Properties  图标。

3. 在 Name 字段中提供了输入硬件规格。

4. 展开 实施/执行 并选择 AdHoc Autostart。

5. 展开 Data Assignments，点 Assignments 字段中的  ，并添加以下内容：



点击 Source 列下拉列表，选择 Constant，并输入 org.kie.api.runtime.process.CaseData（data.get("hwSpec")!= null）。

6. 点击 确定。

28.2. 创建 MANAGER 决策里程ONE

当 managerDecision 变量被赋予响应时，达到此里程one。

流程

1. 在流程设计器中，展开 Object Library 中的 Milestone，并将新的 milestone 拖到 HardwareSpecReady milestone 的下方。

2. 点新的 milestone，然后点击右上角的 Properties 

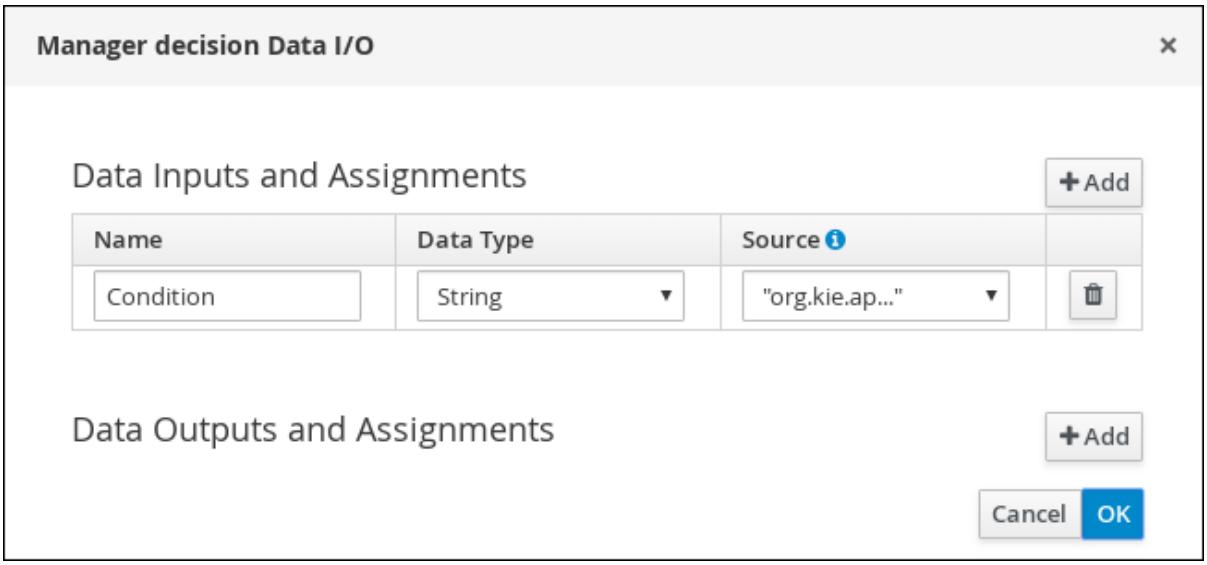
图标。

3. Name 字段中的输入 管理器决定。

4. 展开 实施/执行 并选择 AdHoc Autostart。

5. 展开 Data Assignments，点 Assignments 字段中的

并添加以下内容：



点击 Source 列下拉列表，选择 Constant，并输入 org.kie.api.runtime.process.CaseData（data.get("managerDecision")!= null）。

6. 点击 确定。

28.3. 创建 ORDER PLACED MILESTONE

当 排序 的变量（作为 Place order 子进程的一部分）被达到这个 milestone 时，已得到一个响应。

流程

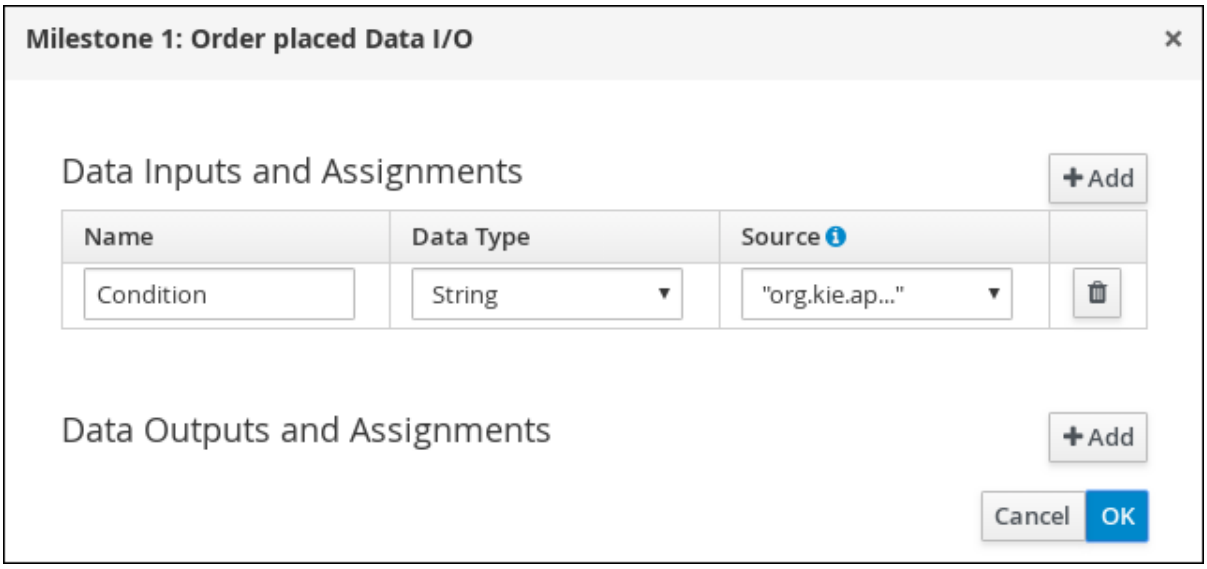
1. 在流程设计器中，展开 Object Library 中的 Milestone，并在 Prepare hardware spec 用户任务下将一个新的 milestone 拖到 canvas 下。

2. 点新的 milestone，然后点击右上角的 Properties 图标。

3. 输入 Milestone 1：顺序放置在 Name 字段中。

4. 展开 实施/执行 并选择 AdHoc Autostart。

5. 展开 Data Assignments，点 Assignments 字段中的 ，并添加以下内容：



点 Source 列下拉列表，选择 Constant，并输入 org.kie.api.runtime.process.CaseData（data.get("ordered")== true）。这意味着，存在一个名为 ordered 的 case 变量，值为 true。

6. 点击 确定。

7. 单击 Milestone 1: Order，并创建一个新脚本任务。

8. 点新脚本任务，然后点击右上角的 Properties 图标。

9. **Name 字段中的输入 Notify requestor。**

10. **展开 实施/执行 和输入 System.out.println("Notification::Order placed");.**

11. 单击 **Notify requestor script 任务，再创建一个信号结束事件。**

12. 单击**信号事件，然后在右上角单击 属性。**

图标。

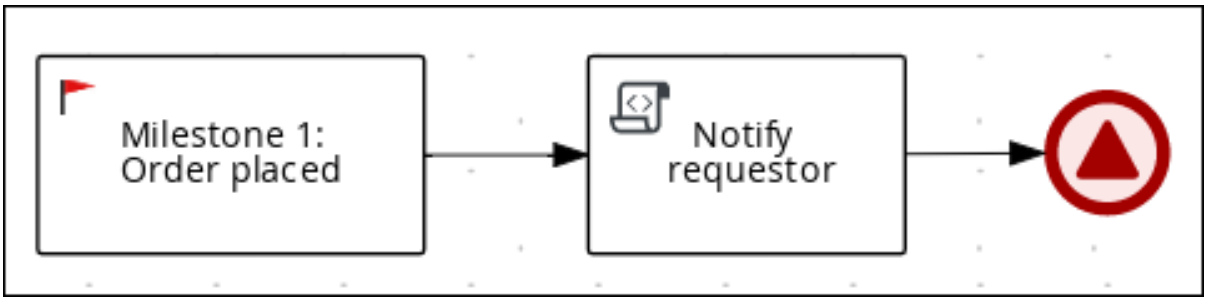
13. **展开 实施/执行，单击 Signal 字段中的下箭头，然后选择 New。**

14. **输入 Milestone 2：顺序提供。**

15. 单击 **Signal Scope 字段中的下箭头，选择 Process Instance。**

16. **点击 Save。**

图 **28.1. 排序出 milestone**



28.4. 创建 ORDER 所提供的 MILESTONE

这个 **milestone 的条件是提供一个名为 的案例文件 变量是 true。此里程one没有启用 adhoc Autostart。相反，当订单就绪时，它由信号事件触发。**

流程

1.

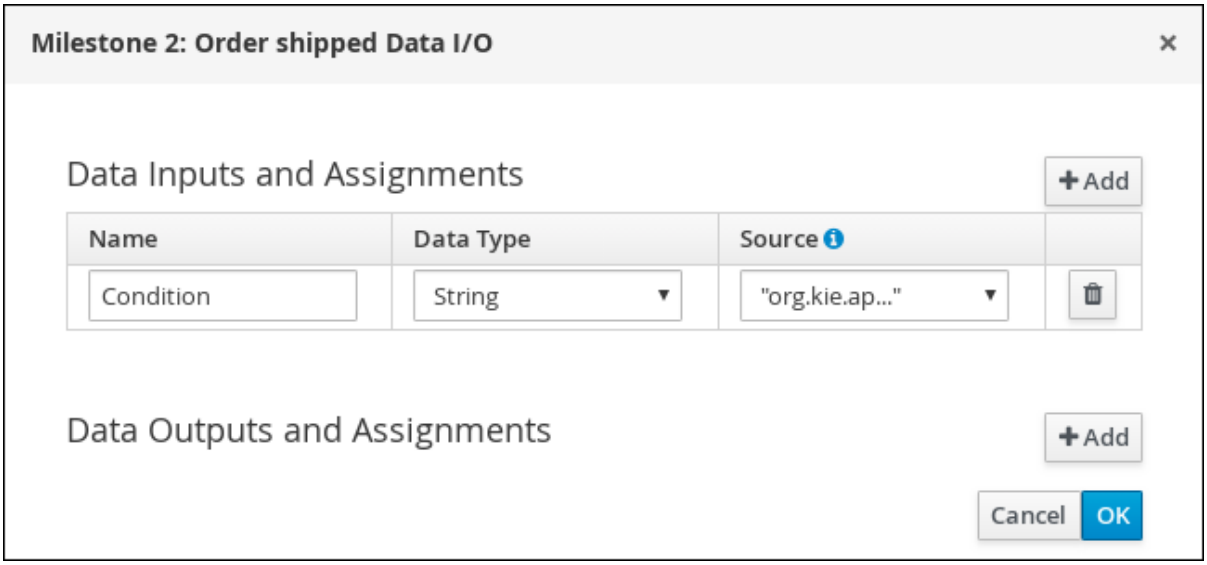
在流程设计器中，展开 Object Library 中的 Milestone，并在 Notify requestor 脚本任务下方将一个新的 milestone 拖到内容下方。

2. 点新的 milestone，然后点击右上角的 Properties 图标。

3. 输入 Milestone 2： 名称 字段中提供的顺序。

4. 扩展 实施/执行，并确保未选择 AdHoc Autostart。

5. 展开 Data Assignments，点 Assignments 字段中的 ，并添加以下内容：



点 Source 列下拉列表，选择 Constant，并输入 org.kie.api.runtime.process.CaseData（data.get("shipped")== true）。这意味着，提供了名为 的大小写变量，其值为 true。

6. 点击 确定。

7. 点 Milestone 2: Order，并创建一个新脚本任务。

8. 点新脚本任务，然后点击右上角的 Properties

图标。

9. **Name 字段中 要跟踪系统 的输入。**

10. **展开 实施/执行 和输入 System.out.println("Order added to tracking system");.**

11. 单击 **Send 以跟踪系统 脚本任务，再创建一个信号结束事件。**

12. 单击**信号事件，然后在右上角单击 属性。**

图标。

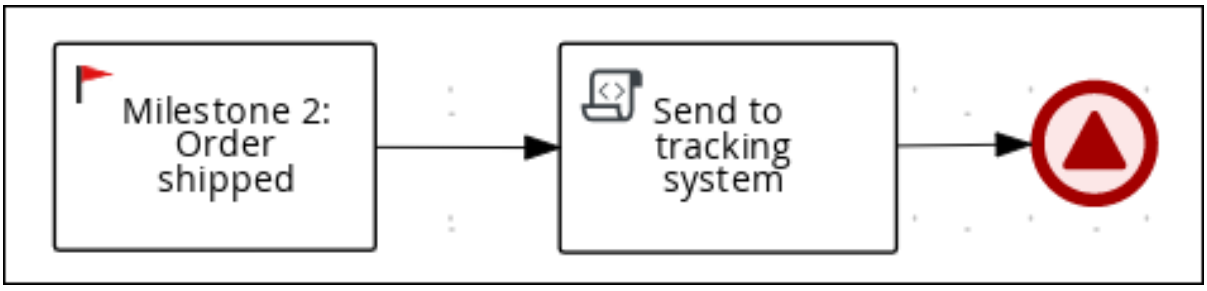
13. **展开 实施/执行，单击 Signal 字段中的下箭头，然后选择 New。**

14. **输入 Milestone 3：向客户交付给客户.**

15. 单击 **Signal Scope 字段中的下箭头，选择 Process Instance。**

16. **点击 Save。**

图 **28.2.** 订单**(milestone)**



28.5. 创建提供给客户里程ONE

这个 **milestone 的条件是名为 delivered 的案例文件变量是 true。此里程one没有启用 adhoc Autostart。相反，它在向客户成功提供订单后，由信号事件触发。**

流程

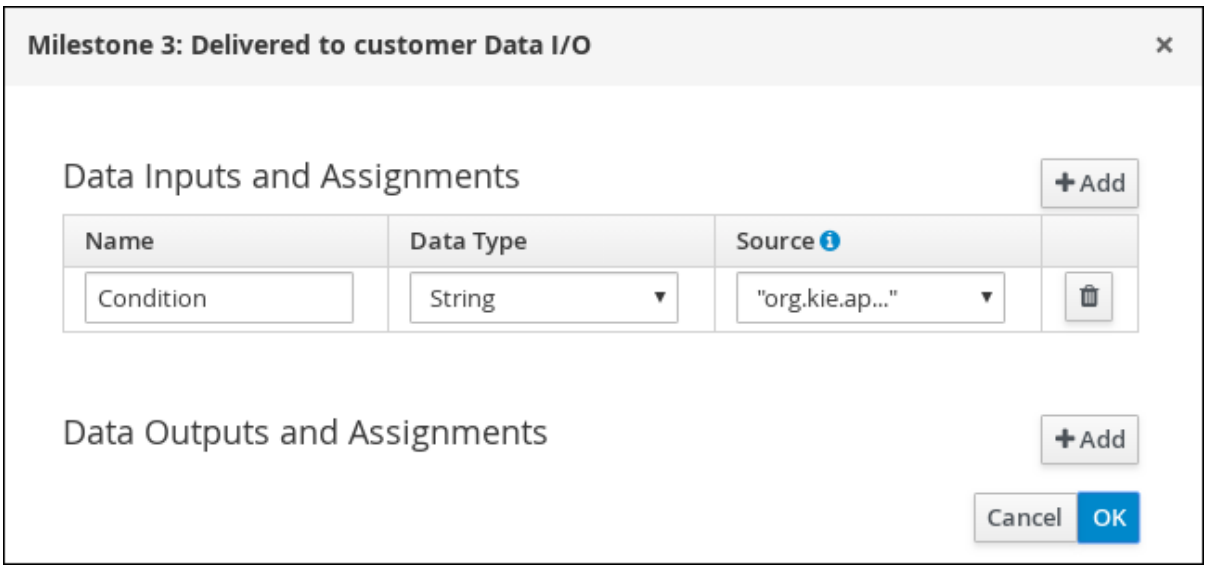
1. 在流程设计器中，展开 **Object Library** 中的 **Milestone**，并在 **Send to tracking system script** 任务下将一个新的 **milestone** 拖到来跟踪系统脚本任务。

2. 点新的 **milestone**，然后点击右上角的 **Properties** 图标。

3. 输入 **Milestone 3：** 在 **Name** 字段中提供给客户。

4. 扩展 实施/执行，并确保未选择 **AdHoc Autostart**。

5. 展开 **Data Assignments**，点 **Assignments** 字段中的 ，并添加以下内容：



点 **Source** 列下拉列表，选择 **Constant**，并输入 **org.kie.api.runtime.process.CaseData（data.get("delivered")== true）**。这意味着，存在一个名为 **delivered** 的 **case** 变量，值为 **true**。

6. 点击 **确定**。

7. 单击 **Milestone 3：向客户交付** 并创建新用户任务。

a.

点新用户任务并点击右上角的 **Properties** 图标。

b.

在 **Name** 字段中输入 客户满意度调查。

c.

展开 **Implementation/Execution**，点 **Actors** 菜单下的 **Add**，点 **Select → New**，以及输入 所有者。

d.

在 **Task Name** 字段中输入 **CustomerSurvey**。

e.

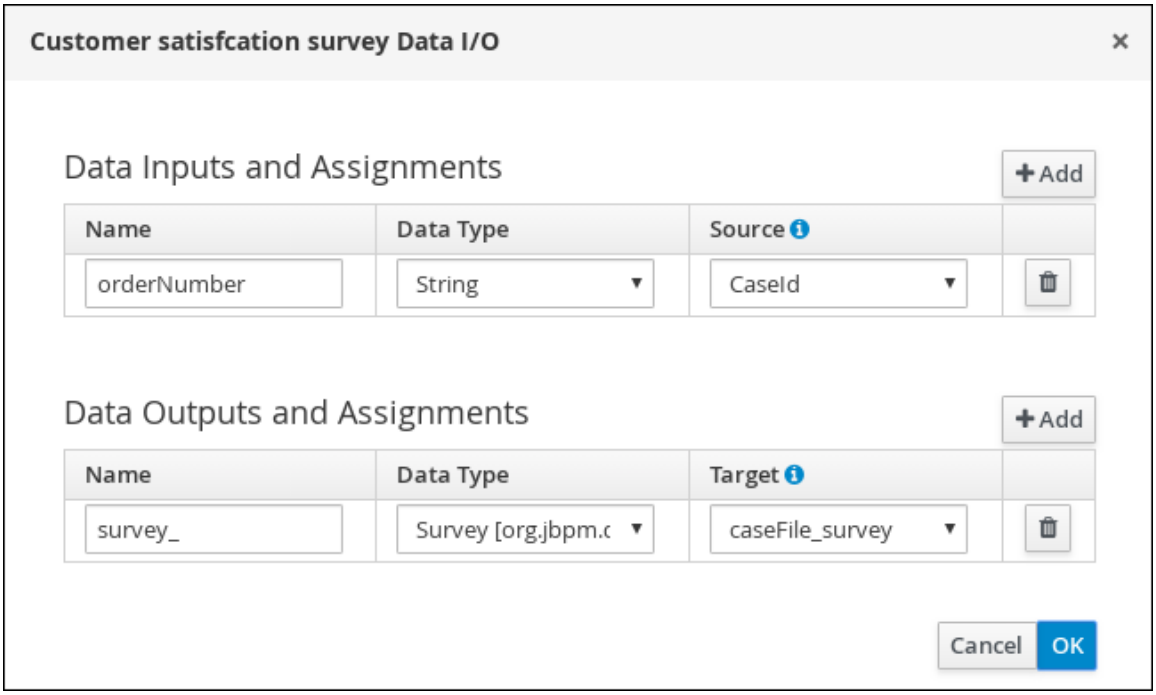
选择 **Skippable** 复选框，并在 **Description** 字段中输入以下文本：

订单 **#{CaseId}** 的满意度调查

f.

在 **Assignments** 字段中点

并添加以下内容：



g.

点击 确定。

8. 单击 **客户满意度调查用户任务，并创建最终活动。**

9. 单击 **Save 以确认更改。**

图 **28.3. 交付给客户里程one**

在所有 milestone 序列完成后，可以关闭 IT 顺序案例。然而，由于情况的特别性质，如果情况并非如此，则可能会重新打开此情况。例如，客户或项目不会收到订单或项目有故障。可以根据需要重新触发任务或添加到问题单定义中，即使在运行时也是如此。

第 29 章 部署和测试 IT 订单案例项目

创建并定义新 IT_Orders_New case 项目的所有组件后，部署和测试新项目。

先决条件

- 您已有一个连接到 Business Central 的 KIE 服务器实例。如需更多信息，请参阅在 *Red Hat JBoss EAP 7.4 上安装和配置 Red Hat Process Automation Manager*。
- 您已在 Business Central 中创建了一个新问题单。更多信息请参阅 第 25 章 *创建新的 IT_Orders 案例项目*。
- 您已创建了数据对象。更多信息请参阅 第 26 章 *数据对象*。
- 您已创建了 Place order 子进程。更多信息请参阅 第 27.1 节 “创建 Place order 子进程”。
- 您已设计了 orderhardware 案例定义。更多信息请参阅 第 27 章 *设计问题单定义*。

流程

1. 在 Business Central 中，转至 Menu → Design → Projects 并点击 IT_Orders_New。
2. 单击 Deploy。
3. 进入 Menu → Manage → Process Definitions → Manage Process Instances → New Process Instance。
4. 进入 Menu → Deploy 并点击 Execution Servers，验证是否已部署并启动新容器。
5. 用例管理显示应用程序启动一个新的问题单实例。有关使用 Showcase 应用程序的说明，*请参阅使用 Showcase 应用程序进行问题单管理*。

第 30 章 其他资源

- *为问题单管理设计和构建案例*
- *使用 Showcase 应用程序进行问题单管理*
- *进程服务的入门*

部分 IV. 开始使用红帽构建的 OPTAPLANNER

作为业务规则开发人员，您可以使用红帽构建的 OptaPlanner 来查找最佳解决方案，以根据一组有限的资源以及在特定限制约束下规划问题。

使用本文档开始使用 OptaPlanner 解决问题。

第 31 章 红帽构建的 OPTAPLANNER 简介

OptaPlanner 是一个轻量级、可嵌入的规划引擎，可优化计划问题。它帮助普通 Java 编程人员有效地解决规划问题，它可将优化性与风险计算相结合。

例如，OptaPlanner 帮助解决各种用例：

- *staff/Patient Ros* ters ：它还有助于为 nurses 创建定时表并跟踪病人管理。
- *教育* 时间表：它有助于安排课程、课程、考试和会议演示。
- *shop Schedules* ：它跟踪车装行、计算机队列计划和 workforce 任务规划。
- *危机* ：通过减少资源消耗（如文章和发证）来最小化浪费。

每个组织都面临规划问题；也就是说，它们提供有限的资源（员工、资产、时间和金钱）。

OptaPlanner 是 Apache 软件许可证 2.0 下的开源软件。它是 100% 的纯 Java，在大多数 Java 虚拟机(JVM)上运行。

31.1. 计划问题

根据有限的资源及特定限制，*计划问题* 是一种最佳目标。最佳目标可以是任何数量，例如：

- 最大化利润 - 实现最大利润最佳目标。
- 最小化原则占用 - 最佳目标具有最低程度的环境影响。
- 最大化员工或客户的满意度 - 最佳目标将优先考虑员工或客户的需求。

实现这些目标的能力取决于可用的资源数量。例如，以下资源可能会受限制：

- 人员数量
- 时间量
- budget
- 物理资产，如机器、车、车、计算机、构建

您还必须考虑与这些资源相关的特定限制，如个人工作小时数、使用某些机器或其它设备间的兼容性。

红帽构建的 OptaPlanner 帮助 Java 编程人员有效地解决约束满意度问题。它将优化 heuristics 和 metaheuristics 合并起来，且有有效的分数计算。

31.2. 规划问题中的 NP 完整性

提供的用例 *可能是* NP-complete 或 NP-hard，这意味着应用以下语句：

- 在合理时间，轻松地验证特定解决方案以解决问题。
- 在合理时间无法找到问题的最佳解决方案。

含义在于解决您的问题可能比您预期的问题更难，因为两个常见的技术并不知道：

- brute 强制算法（即使更高级的变体）过长。
- 例如，一个快速算法，例如在 组合问题 *中，首先放入最大项目*，这是目前从最佳效果最远的解决方案。

通过使用高级优化算法，OptaPlanner 在解决此类计划问题的适当解决方案。

31.3. 用于规划问题的解决方案

计划问题有很多解决方案。

多种解决方案类别如下：

可能的解决方案

可能的解决方法是任何解决方案，无论它是否会破坏任何数量的限制。规划问题通常存在大量可能的解决方案。其中很多解决方案都非常有用。

可行的解决方案

可行的解决方案是不会破坏任何（负）硬约束的解决方案。可行的解决方案数量相对于可能的解决方案。有时没有可行的解决方案。每个可行的解决方案都是一个可能的解决方案。

最佳解决方案

最佳解决方案是具有最高分数的解决方案。规划问题通常具有较少的最佳解决方案。即使没有可行的解决方案，它们始终只有一个最佳解决方案，且最佳解决方案并不可行。

找到最佳解决方案

最好的解决方法是解决方案，在指定时间内按实施获得的最高分值。找到的最佳解决方案可能可行，并有足够的时间，这是最佳的解决方案。

Counterintuively，可能的解决方案数量非常大（如果被正确计算），即使是很小的数据集。

在 planner-engine 分发文件夹中提供的示例中，大多数实例都有大量可能的解决方案。由于无法获得最佳解决方案，因此任何实施都强制评估所有可能的解决方案的子集。

OptaPlanner 支持一些优化算法，以便高效地通过这些算法来有效地处理大量可能的解决方案。

根据用例，某些优化算法会比其他性能更好，但无法预先了解。使用 OptaPlanner，您可以在 XML 或代码的几行中更改解决器配置来切换优化算法。

31.4. 有关规划问题的约束

通常，计划问题至少有两个级别限制：

- *（负）硬约束* 不能出现问题。

  例如，一个老师无法同时教授两个不同的课时。

- 如果可以避免，则应该损坏 *（负）软约束*。

  例如，Teacher A 不希望在星期五下午进行教学。

有些问题也有正的约束：

- 如果可能，应达到 *正的软约束（或好处）*。

  例如，Teacher B 喜欢在上星期一早上教学。

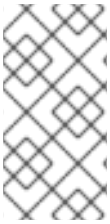
一些基本问题只具有硬限制。有些问题有三个或更多限制级别，如 hard、medium 和 soft 约束。

这些限制定义了计划问题 *的分数计算*（也称为适合性 *功能*）。规划问题的每个解决方案都会以分数为准。使用 OptaPlanner 时，分数限制以面向对象的语言编写，如 Java，或者在 Drools 规则中使用。

这种类型的代码非常灵活，可扩展。

31.5. 红帽构建的 OPTAPLANNER 示例

Red Hat Process Automation Manager 提供了几个 OptaPlanner 示例。您可以检查示例代码，并根据需要对其进行修改，以满足您的需要。



注意

红帽不提供对 Red Hat Process Automation Manager 发行版本中包含的示例代码的支持。

某些 OptaPlanner 示例可以解决问题，这些问题在学术 contests中呈现。下表中的 Contest 列列出了 contests。它还标识了示例，*作为 一项测试的目的或不切实际的。真实的竞争* 测试是满足以下标准的官方独立测试：

- 明确定义实际用例
- 实际限制
- 多个真实数据集
- 在特定硬件的特定时间限制内可重复生成的结果
- 来自学术和/或企业运营研究社区的严重参与。

真实的 Contests 提供了 OptaPlanner 与竞争软件和学术研究的目标比较。

表 31.1. 示例概述

示例	域	大小	Contest	目录名称
N queens	1 个实体类 （1 变量）	**256**个实体 值 **256** 搜索空间 **^616**	无得分 (cheatable)	**nqueens**
云平衡	1 个实体类 （1 变量）	entity abrt **2400** 值 IFL **800** 搜索空间 Warning **10^6967**	否（由我们定义）	**Cloudbalancing**

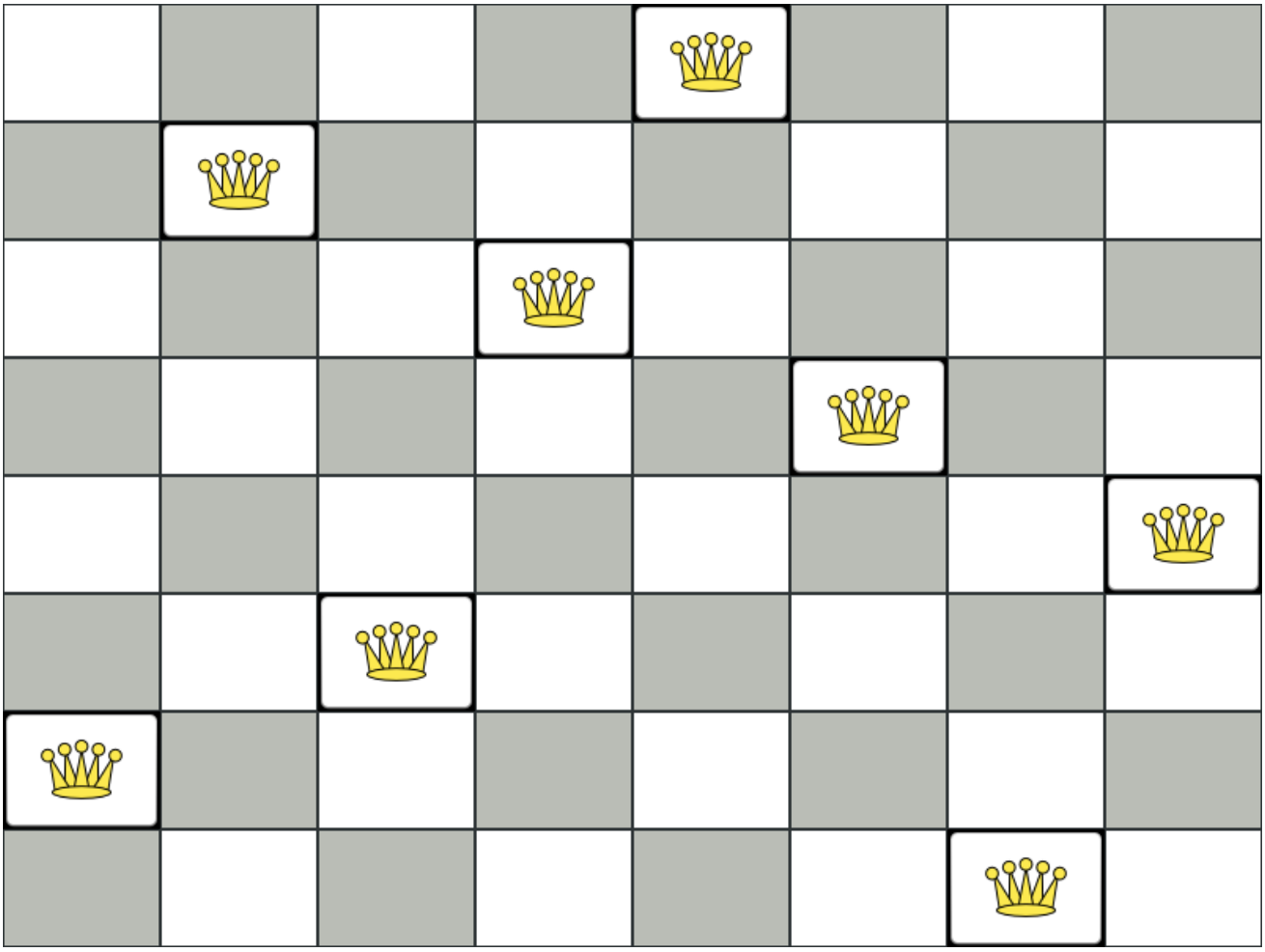
示例	域	大小	Contest	目录名称
traveling salesman	1 个实体类 （1链的变量）	entity dropped **980** 值 InventoryService **980** 搜索空间 768 **10^2504**	不切实际的 TSP web	**tsp**
Tennis club 调度	1 个实体类 （1 变量）	entity abrt **72** 值 InventoryService **7** 搜索空格 Warning **10^60**	否（由我们定义）	**十nis**
会议调度	1 个实体类 （2 变量）	entity abrt **10** 值 abrt **320** 和 时间 **5** 搜索空间 768 **10^320**	否（由我们定义）	**会议阶段**
课程时间表	1 个实体类 （2 变量）	entity Equal **434** 值 abrt **25** 和 时间 **20** 搜索空格 Warning **10^1171**	现实 ITC 2007 年跟踪 3	**curriculumCourse**
机器重新分配	1 个实体类 （1 变量）	entity Equal **50000** 值 **5000** 搜索空间 **^184948**	2012 年几乎真实的 ROADEF	**machineReassignment**
载体路由	1 个实体类 （1链的变量） 1 个影子实体类 （1 个自动影子变量）	entity abrt **2740** 值 InventoryService **2795** 搜索空间 768 **10^8380**	无切实际的 VRP 网络	**vehiclerouting**
使用时间窗 的载设备路由	所有 Vehicle 路由 （1 个影子变量）	entity abrt **2740** 值 InventoryService **2795** 搜索空间 768 **10^8380**	无切实际的 VRP 网络	**vehiclerouting**

示例	域	大小	Contest	目录名称
项目作业调度	1 个实体类 （2 变量） （1 个影子变量）	entity Equal **640** 值 fsanitize **?** 和 sHistoryLimit **?** 搜索空格 **?**	2013 年几乎真实的 MISTA	**projectjobscheduling**
任务分配	1 个实体类 （1链的变量） （1 个影子变量） 1 个影子实体类 （1 个自动影子变量）	entity Equal **500** 值 InventoryService **520** 搜索空间 768 **10^1168**	没有被我们定义	**taskassigning**
考试时间表	2 个实体类（层次结构） （2 变量）	entity abrt **1096** 值 InventoryService **80** 和 时间 **49** 搜索空格 >_< **10^3374**	真实的 ITC 2007 年跟踪 1	**考试项目**
Nurse rostering	1 个实体类 （1 变量）	entity abrt **752** 值 IFL **50** 搜索空间 768 **10^1277**	现实 INRC 2010	**nurserostering**
traveling tournament	1 个实体类 （1 变量）	entity abrt **1560** 值 sHistoryLimit **78** 搜索空间 Warning **10^2301**	不切实际的 TTP	**travelingtournament**
更低的时间调度	1 个实体类 （2 变量）	entity Equal **500** 值 InventoryService **100** 和 时间 **288** 搜索空间 768 **10^20078**	几乎真实的 ICON Energy	**cheaptimescheduling**
投资	1 个实体类 （1 变量）	entity abrt **11** 值 = **1000** 搜索空间 768 **10^4**	没有被我们定义	**投资**

示例	域	大小	Contest	目录名称
会议调度	1 个实体类 （2 变量）	IFL **216** 值 abrt **18** 和 时间 **20** 搜索空间 768 **10^552**	没有被我们定义	**会议阶段**
sttour	1 个实体类 （1链的变量） （4 个影子变量） 1 个影子实体类 （1 个自动影子变量）	entity dropped **47** 值 abrt **48** 搜索空格 768 **10^59**	没有被我们定义	**sttour**
flight crew 调度	1 个实体类 （1 变量） 1 个影子实体类 （1 个自动影子变量）	entity dropped **4375** 值 abrt **750** 搜索空间 2.4 **10^12578**	没有被我们定义	**flightcrewschedulin g**

31.6. N QUEENS

在 *n* 个大小小板上放置 *n* queens，以便无法互相攻击。最常见的 *n* queens puzzle 是 8 个 queuzzle，有 *n = 8* ：



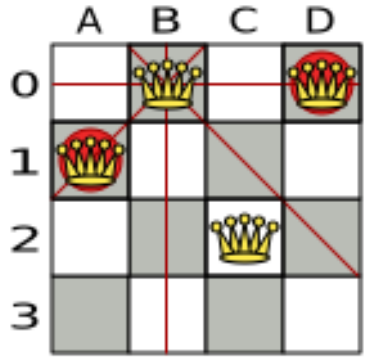
约束：

- 使用主板的 *n* 列和 *n* 行。
- 在主板上放置 *n* queens。
- 无法相互攻击两个频率。queen 可攻击同一横向、垂直或部门其他任何频率的其他频率。

本文档主要使用四个不同点。

建议的解决方案可能是：

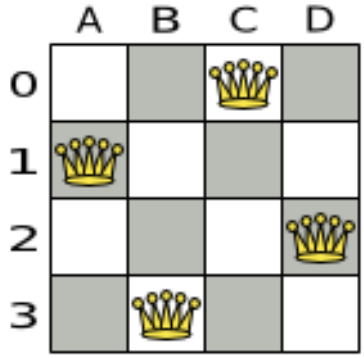
图 **31.1. 四点欺诈解决方案**



以上解决方案是错误的，因为 **queens A1 和 B0 可以相互攻击（因此可以说 B0 和 D0）。删除 queen B0 会尊重"不两种 queens"约束，但会破坏 "place *n* queens" 约束。**

以下是一个正确的解决方案：

图 **31.2. Four quele queuzzle 的正确解决方案**



所有约束都已满足，因此解决方案正确。

请注意，大多数 *n* 个词汇都有多个正确的解决方案。我们将重点介绍查找特定 *n* 的单一正确的解决方案，而不是为特定 *n* 查找可能的正确解决方案。

问题大小

```
4queens   has   4 queens with a search space of    256.
8queens   has   8 queens with a search space of   10^7.
16queens  has  16 queens with a search space of  10^19.
32queens  has  32 queens with a search space of  10^48.
64queens  has  64 queens with a search space of 10^115.
256queens has 256 queens with a search space of 10^616.
```

N queens 示例的实现还没有优化，因为它作为新手示例的功能。然而，它可以轻松地处理 64 queens。出现了一些变化，它已被显示，可轻松处理 5000 queens 等等。

31.6.1. N queens 的域模型

这个示例使用域模型解决四条问题。

- 创建域模型

  好的域模型可以方便理解和解决您的规划问题。

  这是 n queens 示例的域模型：

```
public class Column {

    private int index;

    // ... getters and setters
}
```

```
public class Row {

    private int index;

    // ... getters and setters
}
```

```
public class Queen {

    private Column column;
    private Row row;

    public int getAscendingDiagonalIndex() {...}
    public int getDescendingDiagonalIndex() {...}

    // ... getters and setters
}
```

- 计算搜索空间.

Queen 实例有一个 Column （例如：0 为列 A， 1 为列 B, ...）和一个 Row （its 行， 例如 0 代表行 0， 1 是行 1, ...）。

可以根据列和行计算升序行和降序行。

列和行索引从主板的左上角开始。

```
public class NQueens {

    private int n;
    private List<Column> columnList;
    private List<Row> rowList;

    private List<Queen> queenList;

    private SimpleScore score;

    // ... getters and setters
}
```

- 查找解决方案

单个 NQueens 实例包含所有 Queen 实例的列表。它是 Solution 实施， 它将提供给、被解决并从 Solver 中检索。

请注意， 在四个 queens 示例中， NQueens getN（） 方法总是返回 4。

图 31.3. Four Queens 的解决方案

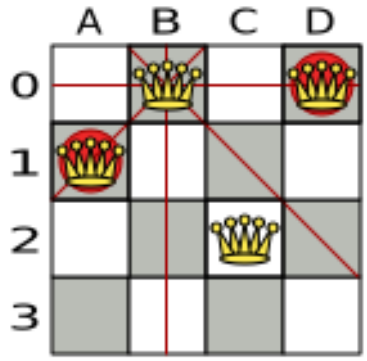


表 31.2. 域模型中的解决方案详情

	columnIndex	rowIndex	ascendingDiagonalIndex (columnIndex + rowIndex)	descendingDiagonalIndex (columnIndex - rowIndex)
A1	0	1	1 (**)	-1
B0	1	0 (*)	1 (**)	1
C2	2	2	4	0
D0	3	0 (*)	3	3

当两个 queens 共享同一列时，行或横线，如(*)和(**)，它们可以相互攻击。

31.7. 云平衡

有关本示例的详情，请参阅 Red Hat build of OptaPlanner quick start Guide。

31.8. TRAVELING SALESMAN（TSP - TRAVELING SALESMAN 问题）

给定城市列表，找到一个只访问每个城市的 salesman 最短导览。

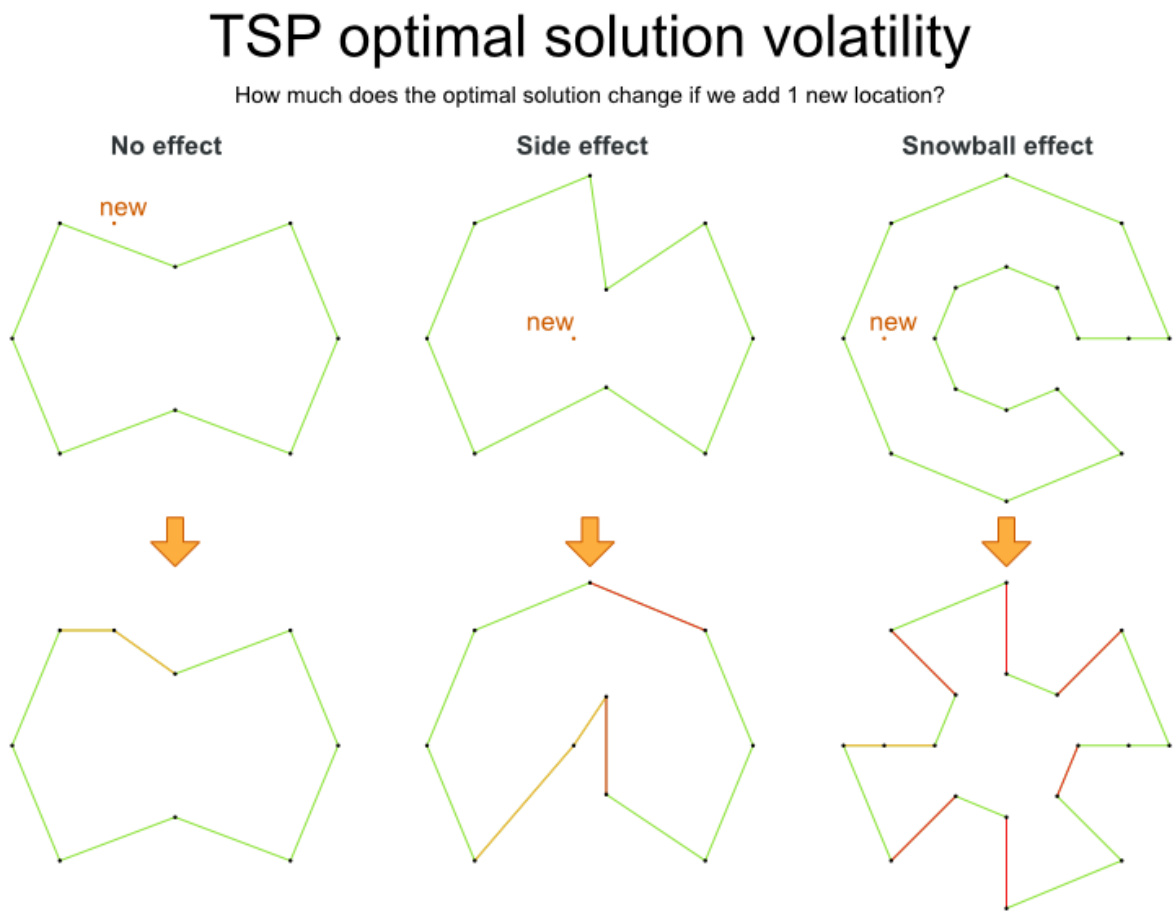
该问题由 Wikipedia 定义。它是计算 计算中最广泛调查的问题之一。然而，在现实世界中，这通常只是规划问题的一部分，以及其他约束，如员工切换限制。

问题大小

> *dj38 has 38 cities with a search space of 10^43.*
> *europe40 has 40 cities with a search space of 10^46.*
> *st70 has 70 cities with a search space of 10^98.*
> *pcb442 has 442 cities with a search space of 10^976.*
> *lu980 has 980 cities with a search space of 10^2504.*

问题困难

尽管 TSP 的简单定义，但问题很难解决。因为这是一个 NP-hard 问题（如大多数规划问题），当问题数据集中稍有改变时，特定问题 dataset 的最佳解决方案可能会改变：



31.9. TENNIS CLUB 调度

每周 10nis club 都有四个团队相互推断。将这四个点分配给团队。

硬约束：

- *冲突：团队每天只能每周一次。*
- *不可用：一些团队在一些日期不可用。*

Medium 约束：

- *公平分配：所有团队都应扮演（几乎）相同的时间。*

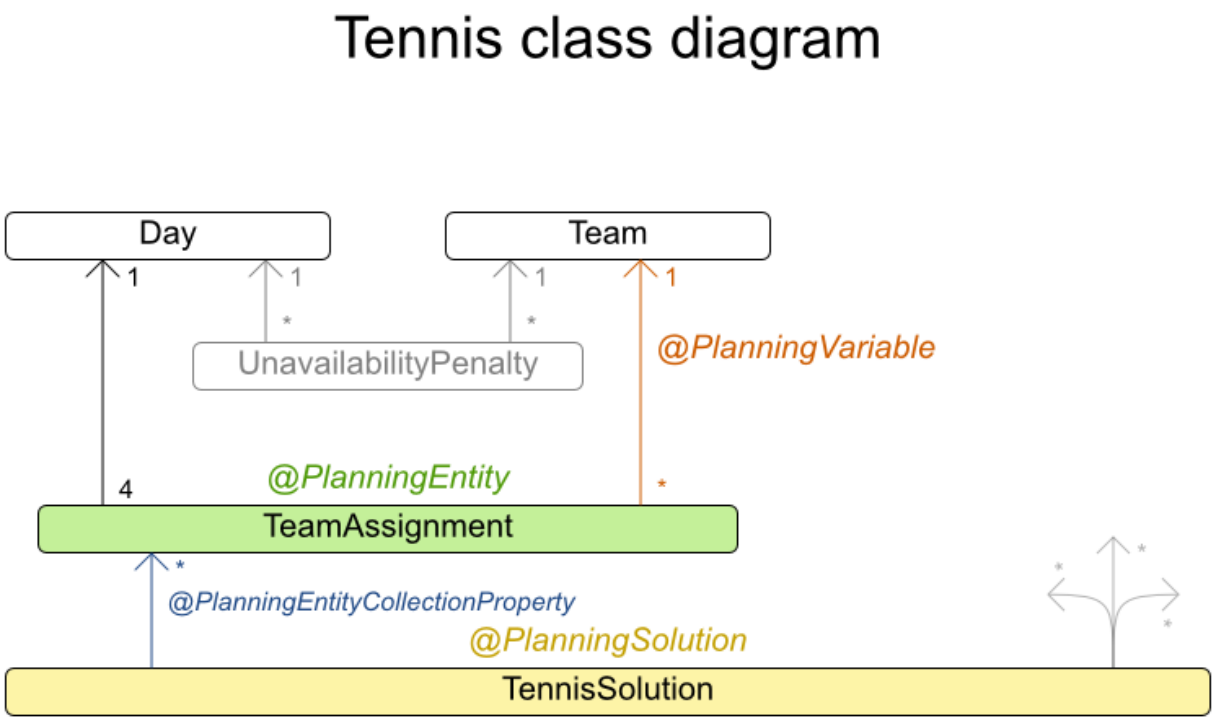
软限制：

- *均匀的前端：每个团队应针对其他每个团队都有一个相同的次数。*

问题大小

> *munich-7teams has 7 teams, 18 days, 12 unavailabilityPenalties and 72 teamAssignments with a search space of 10^60.*

图 31.4. 域模型



31.10. 会议调度

为启动时间和房间分配每个会议。会议有不同的持续时间。

硬约束：

- *间冲突：两个会议不能同时使用同一空间。*
- *所需参与者：一个人在同时无法进行两个所需的会议。*
- *必需的房间容量：会议不能适合于所有会议的参与者。*
- *在同一天开始和结束：无法安排在多天的会议。*

***Medium** 约束：*

- *首选参与人：一个人不能同时拥有两个首选会议，而且是首选，而且是需要同时进行的会议。*

软限制：

- *更快而不是之后：尽快计划所有会议。*
- *会议间的休息：任何两个会议至少应在这两间中断。*
- *重叠会议：为了尽量减少并行会议的数量，用户不必选择另外一种会议。*
- *首先分配较大的空间：如果有一个大的房间可用，应该分配给该会议，以便适应尽可能多的人，即使他们尚未签到该会议。*
- *房间稳定性：如果一个人连续有两个或小时间会议，它们之间的时间较差在同一房里。*

问题大小

> *50meetings-160timegrains-5rooms has 50 meetings, 160 timeGrains and 5 rooms with a search space of 10^145.*
> *100meetings-320timegrains-5rooms has 100 meetings, 320 timeGrains and 5 rooms with a search space of 10^320.*
> *200meetings-640timegrains-5rooms has 200 meetings, 640 timeGrains and 5 rooms with a search space of 10^701.*
> *400meetings-1280timegrains-5rooms has 400 meetings, 1280 timeGrains and 5 rooms with a search space of 10^1522.*
> *800meetings-2560timegrains-5rooms has 800 meetings, 2560 timeGrains and 5 rooms with a search space of 10^3285.*

31.11. 课程时间表（ITC 2007 年跟踪 3 - 日程表课程安排）

安排每个授课内容进入一个时间，并入一个房间。

硬约束：

- *老师冲突：老师不能在同一期限内没有两个讲义。*
- *课程冲突：课程不能在同一时间内拥有两个讲义。*
- *房间：两个讲话不能在同一时间段内。*
- *不可用周期（为每个数据集指定）：必须将特定的课程分配给特定的时间段。*

软限制：

- *房间容量：房间容量不应少于学员在演讲中的数量。*
- *最小工作日：在同一课程的讲义应分发至最少的天数。*

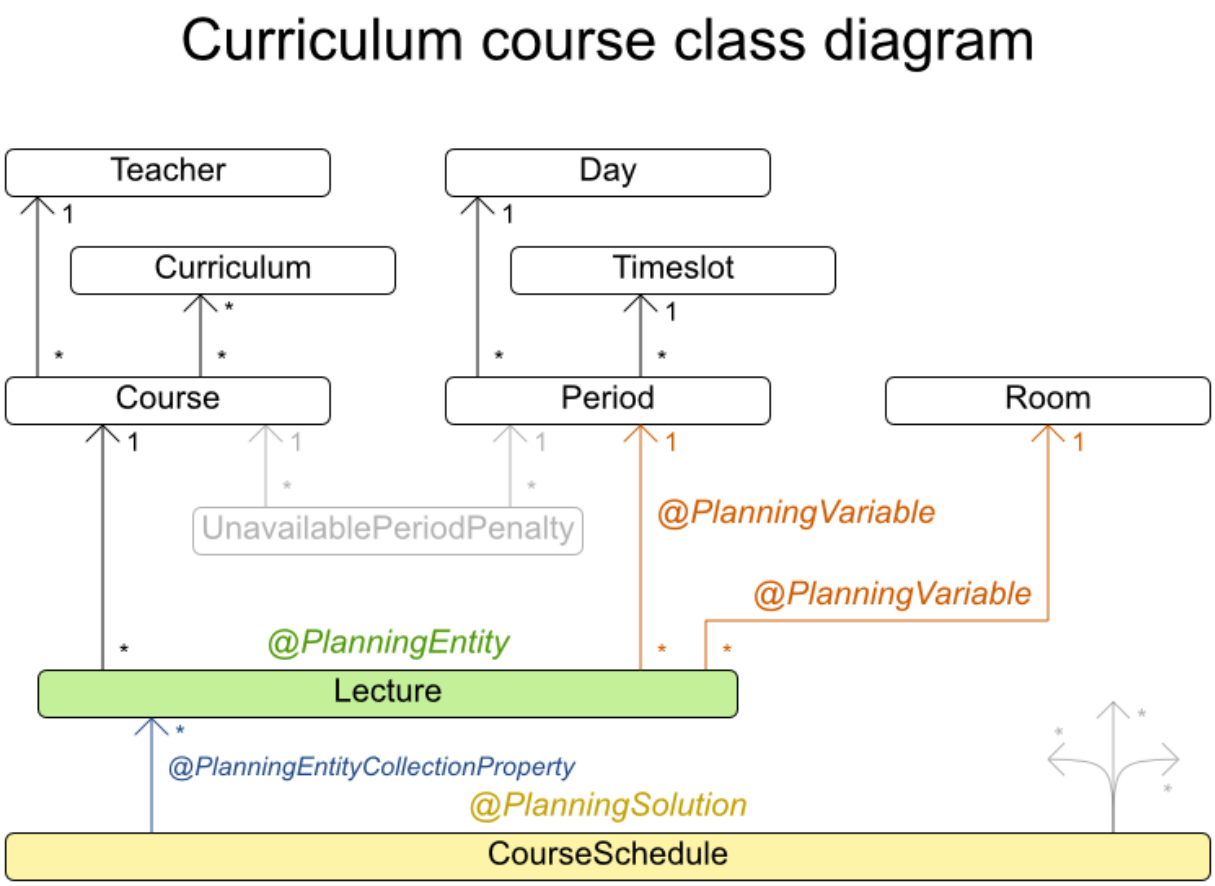
- *课程紧凑：属于相同课程的演讲应相互相邻（在连续的期间内）。*
- *房间稳定性：指示相同课程的演讲应该分配到相同的房间。*

这个问题由 国际时间选项卡 2007 年跟踪 3 定义。

问题大小

> *comp01 has 24 teachers, 14 curricula, 30 courses, 160 lectures, 30 periods, 6 rooms and 53 unavailable period constraints with a search space of 10^360.*
> *comp02 has 71 teachers, 70 curricula, 82 courses, 283 lectures, 25 periods, 16 rooms and 513 unavailable period constraints with a search space of 10^736.*
> *comp03 has 61 teachers, 68 curricula, 72 courses, 251 lectures, 25 periods, 16 rooms and 382 unavailable period constraints with a search space of 10^653.*
> *comp04 has 70 teachers, 57 curricula, 79 courses, 286 lectures, 25 periods, 18 rooms and 396 unavailable period constraints with a search space of 10^758.*
> *comp05 has 47 teachers, 139 curricula, 54 courses, 152 lectures, 36 periods, 9 rooms and 771 unavailable period constraints with a search space of 10^381.*
> *comp06 has 87 teachers, 70 curricula, 108 courses, 361 lectures, 25 periods, 18 rooms and 632 unavailable period constraints with a search space of 10^957.*
> *comp07 has 99 teachers, 77 curricula, 131 courses, 434 lectures, 25 periods, 20 rooms and 667 unavailable period constraints with a search space of 10^1171.*
> *comp08 has 76 teachers, 61 curricula, 86 courses, 324 lectures, 25 periods, 18 rooms and 478 unavailable period constraints with a search space of 10^859.*
> *comp09 has 68 teachers, 75 curricula, 76 courses, 279 lectures, 25 periods, 18 rooms and 405 unavailable period constraints with a search space of 10^740.*
> *comp10 has 88 teachers, 67 curricula, 115 courses, 370 lectures, 25 periods, 18 rooms and 694 unavailable period constraints with a search space of 10^981.*
> *comp11 has 24 teachers, 13 curricula, 30 courses, 162 lectures, 45 periods, 5 rooms and 94 unavailable period constraints with a search space of 10^381.*
> *comp12 has 74 teachers, 150 curricula, 88 courses, 218 lectures, 36 periods, 11 rooms and 1368 unavailable period constraints with a search space of 10^566.*
> *comp13 has 77 teachers, 66 curricula, 82 courses, 308 lectures, 25 periods, 19 rooms and 468 unavailable period constraints with a search space of 10^824.*
> *comp14 has 68 teachers, 60 curricula, 85 courses, 275 lectures, 25 periods, 17 rooms and 486 unavailable period constraints with a search space of 10^722.*

图 31.5. 域模型



31.12. 机器重新分配(GOOGLE ROADEF 2012)

为机器分配每个进程。所有进程已经有原始（未优化）分配。每个进程需要每个资源（如 CPU 或 RAM）的数量。这是 Cloud Balancing 示例的一个更为复杂的版本。

硬约束：

- *最大容量：不能超过每台机器的每个资源的最大容量。*
- *冲突：同一服务的进程必须在不同的机器上运行。*
- *分布：必须将同一服务的进程分散到各个位置。*
- *依赖项：取决于其他服务的进程必须在其他服务的邻居中运行。*

- 临时使用：有些资源是临时的，可以计入原始机器作为新分配机器的最大容量。

软限制：

- load：不应超过每台机器的每个资源的安全容量。
- balances：通过平衡每台机器上的可用资源，为以后分配空间。
- 流程移动成本：流程具有移动成本。
- 服务迁移成本：服务具有移动成本。
- 机器迁移成本：将进程从机器 A 移到机器 B 还有另一个特定于 A-B 的移动成本。

这个问题由 Google ROADEF/EURO Challenge 定义，2012 年。

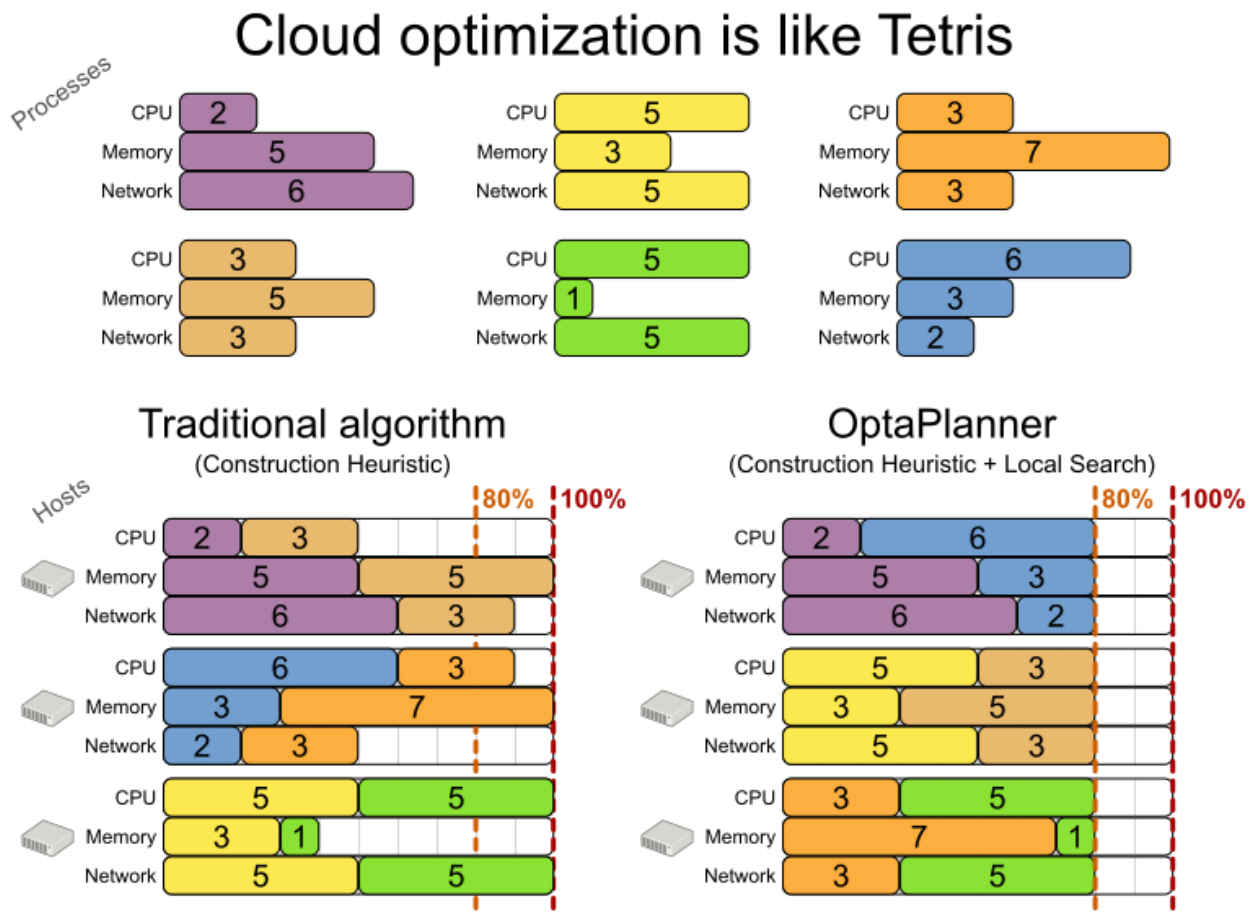
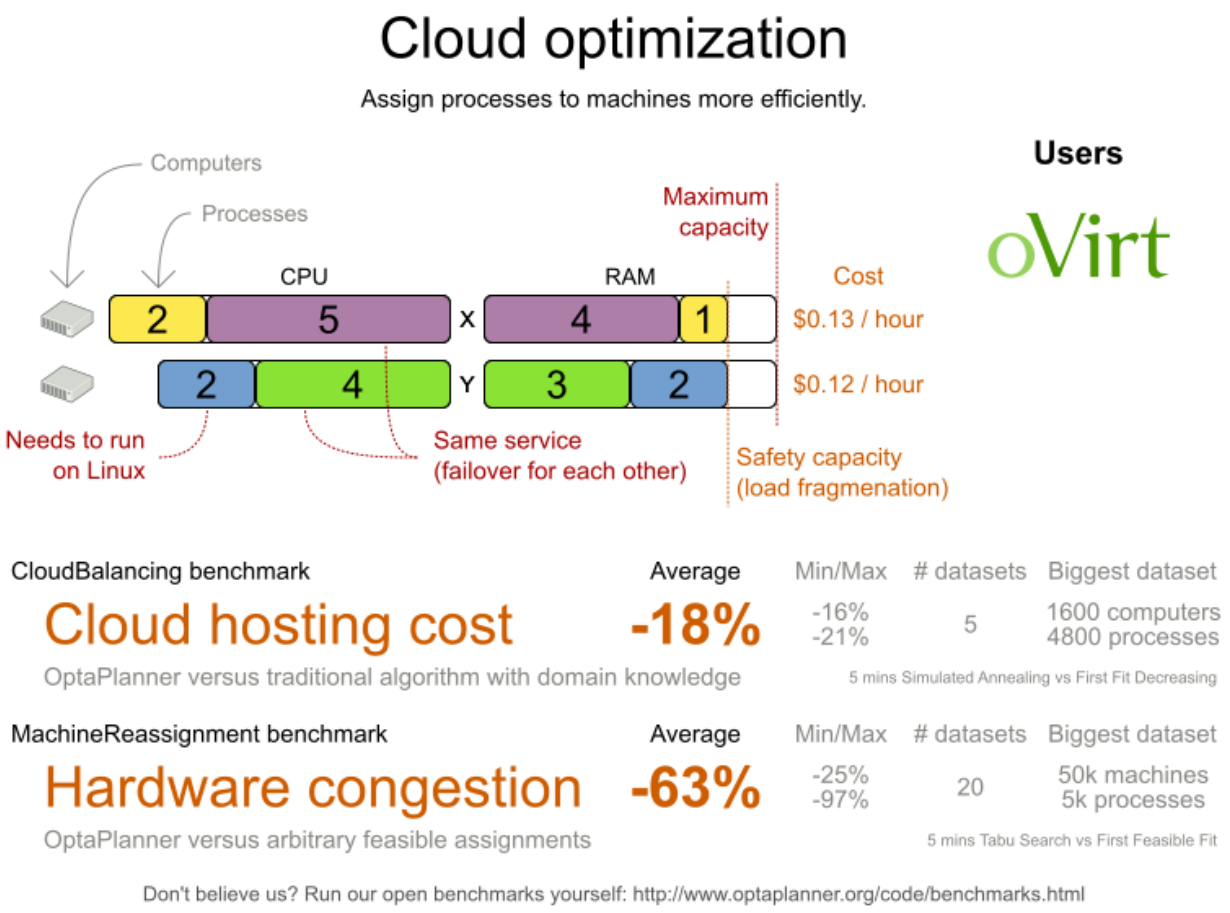

Cloud optimization is like Tetris
Processes
CPU 2
Memory 5
Network 6
CPU 5
Memory 3
Network 5
CPU 3
Memory 7
Network 3
CPU 3
Memory 5
Network 3
CPU 5
Memory 1
Network 5
CPU 6
Memory 3
Network 2
Traditional algorithm
(Construction Heuristic)
Hosts
80%
100%
CPU 2 3
Memory 5 5
Network 6 3
CPU 6 3
Memory 3 7
Network 2 3
CPU 5 5
Memory 3 1
Network 5 5
OptaPlanner
(Construction Heuristic + Local Search)
80%
100%
CPU 2 6
Memory 5 3
Network 6 2
CPU 5 3
Memory 3 5
Network 5 3
CPU 3 5
Memory 7 1
Network 3 5

图 31.6. 价值主张

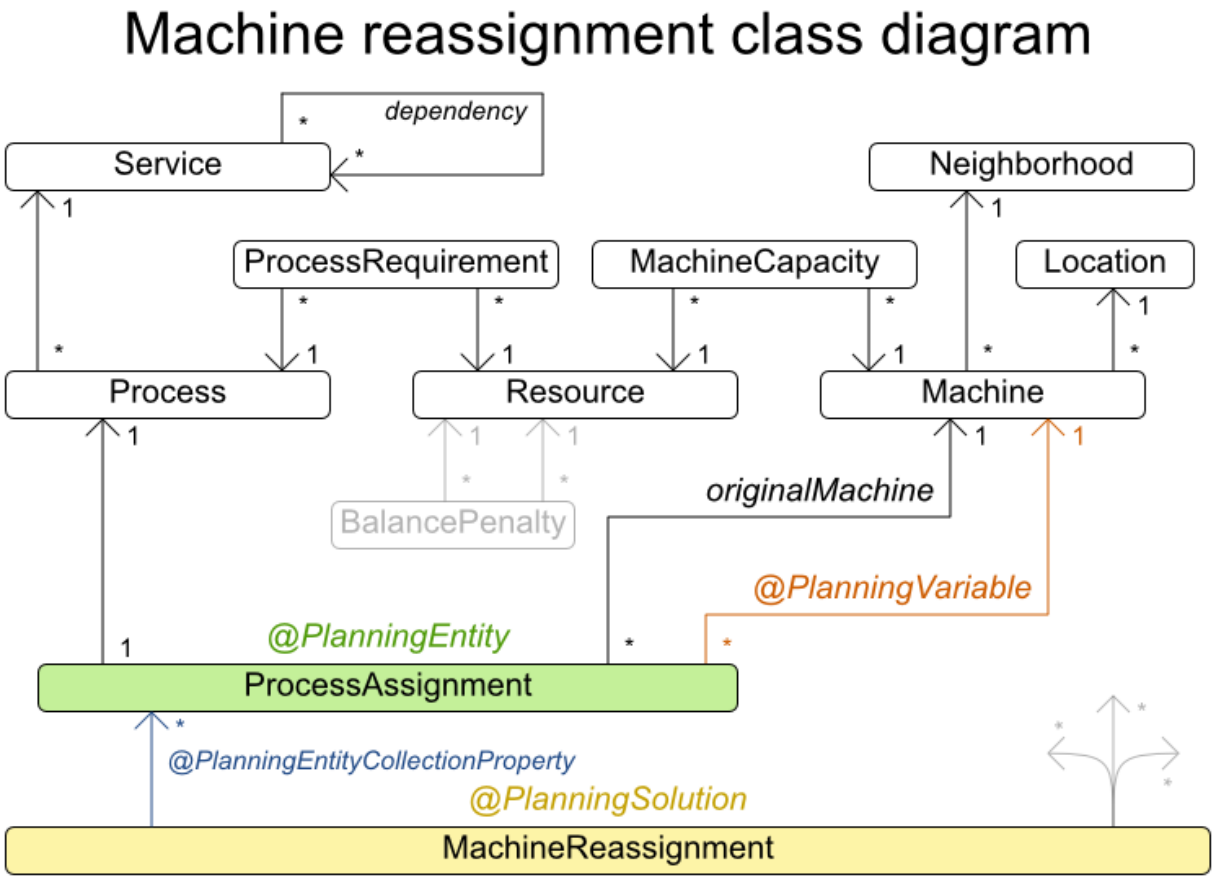


问题大小

> *model_a1_1 has 2 resources, 1 neighborhoods, 4 locations, 4 machines, 79 services, 100 processes and 1 balancePenalties with a search space of 10^60.*
> *model_a1_2 has 4 resources, 2 neighborhoods, 4 locations, 100 machines, 980 services, 1000 processes and 0 balancePenalties with a search space of 10^2000.*
> *model_a1_3 has 3 resources, 5 neighborhoods, 25 locations, 100 machines, 216 services, 1000 processes and 0 balancePenalties with a search space of 10^2000.*
> *model_a1_4 has 3 resources, 50 neighborhoods, 50 locations, 50 machines, 142 services, 1000 processes and 1 balancePenalties with a search space of 10^1698.*
> *model_a1_5 has 4 resources, 2 neighborhoods, 4 locations, 12 machines, 981 services, 1000 processes and 1 balancePenalties with a search space of 10^1079.*
> *model_a2_1 has 3 resources, 1 neighborhoods, 1 locations, 100 machines, 1000 services, 1000 processes and 0 balancePenalties with a search space of 10^2000.*
> *model_a2_2 has 12 resources, 5 neighborhoods, 25 locations, 100 machines, 170 services, 1000 processes and 0 balancePenalties with a search space of 10^2000.*
> *model_a2_3 has 12 resources, 5 neighborhoods, 25 locations, 100 machines, 129 services, 1000 processes and 0 balancePenalties with a search space of 10^2000.*
> *model_a2_4 has 12 resources, 5 neighborhoods, 25 locations, 50 machines, 180 services, 1000 processes and 1 balancePenalties with a search space of 10^1698.*
> *model_a2_5 has 12 resources, 5 neighborhoods, 25 locations, 50 machines, 153 services, 1000 processes and 0 balancePenalties with a search space of 10^1698.*
> *model_b_1 has 12 resources, 5 neighborhoods, 10 locations, 100 machines, 2512 services, 5000*

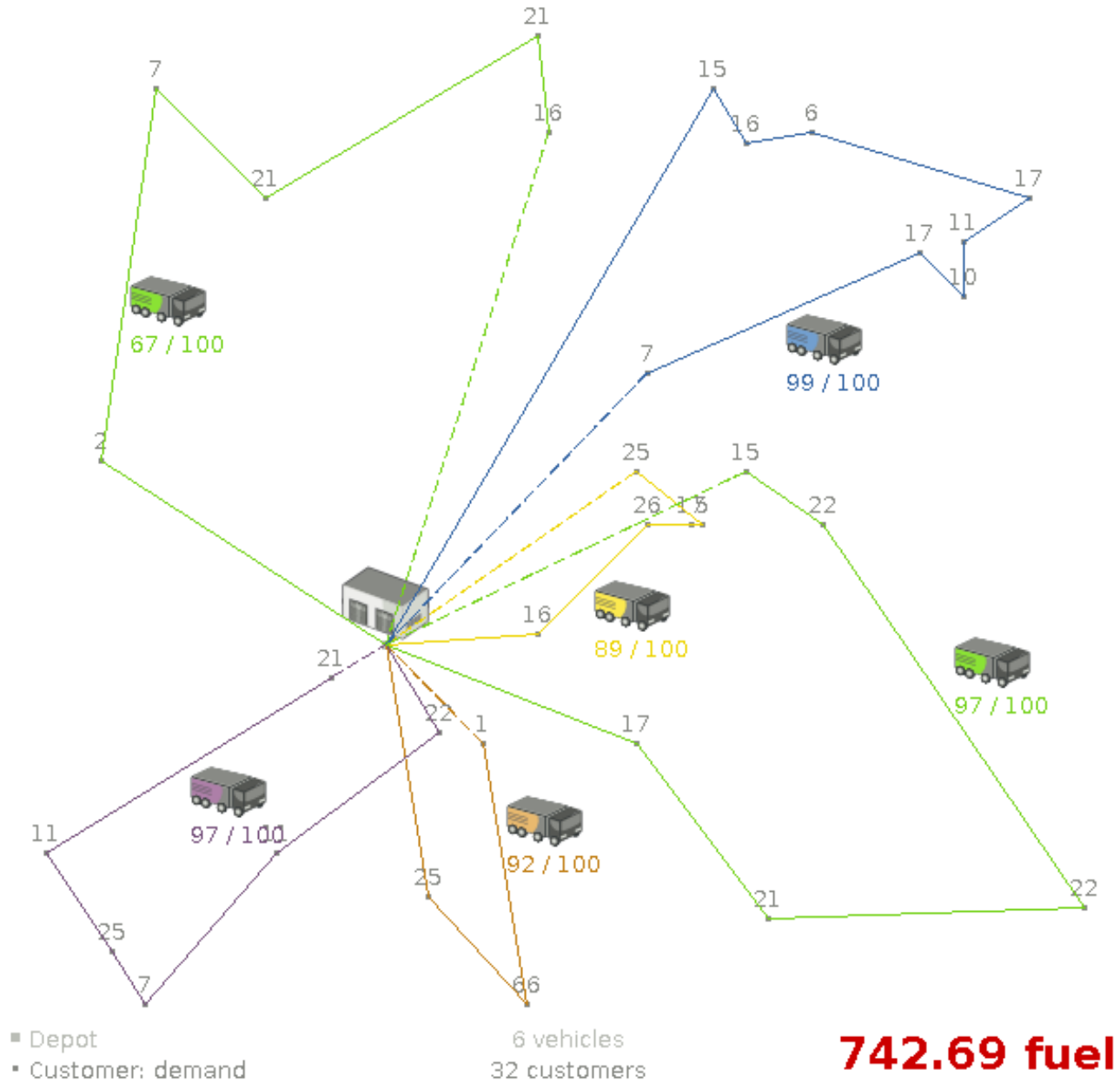
processes and 0 balancePenalties with a search space of 10^10000.
model_b_2 has 12 resources, 5 neighborhoods, 10 locations, 100 machines, 2462 services, 5000 processes and 1 balancePenalties with a search space of 10^10000.
model_b_3 has 6 resources, 5 neighborhoods, 10 locations, 100 machines, 15025 services, 20000 processes and 0 balancePenalties with a search space of 10^40000.
model_b_4 has 6 resources, 5 neighborhoods, 50 locations, 500 machines, 1732 services, 20000 processes and 1 balancePenalties with a search space of 10^53979.
model_b_5 has 6 resources, 5 neighborhoods, 10 locations, 100 machines, 35082 services, 40000 processes and 0 balancePenalties with a search space of 10^80000.
model_b_6 has 6 resources, 5 neighborhoods, 50 locations, 200 machines, 14680 services, 40000 processes and 1 balancePenalties with a search space of 10^92041.
model_b_7 has 6 resources, 5 neighborhoods, 50 locations, 4000 machines, 15050 services, 40000 processes and 1 balancePenalties with a search space of 10^144082.
model_b_8 has 3 resources, 5 neighborhoods, 10 locations, 100 machines, 45030 services, 50000 processes and 0 balancePenalties with a search space of 10^100000.
model_b_9 has 3 resources, 5 neighborhoods, 100 locations, 1000 machines, 4609 services, 50000 processes and 1 balancePenalties with a search space of 10^150000.
model_b_10 has 3 resources, 5 neighborhoods, 100 locations, 5000 machines, 4896 services, 50000 processes and 1 balancePenalties with a search space of 10^184948.

*图 **31.7. 域模型***



31.13. 载体路由

使用车队车队，选择每个客户的对象并将其带给它。每个车块都可以为多个客户提供服务，但其容量有限。



除了基本情况(CVRP)外，还有一个包含时间窗(CVRPTW)的变体。

硬约束：

- *载体容量：一种车块无法获取更多项目，然后它的容量。*
- *时间窗（仅在 CVRPTW 中）：*

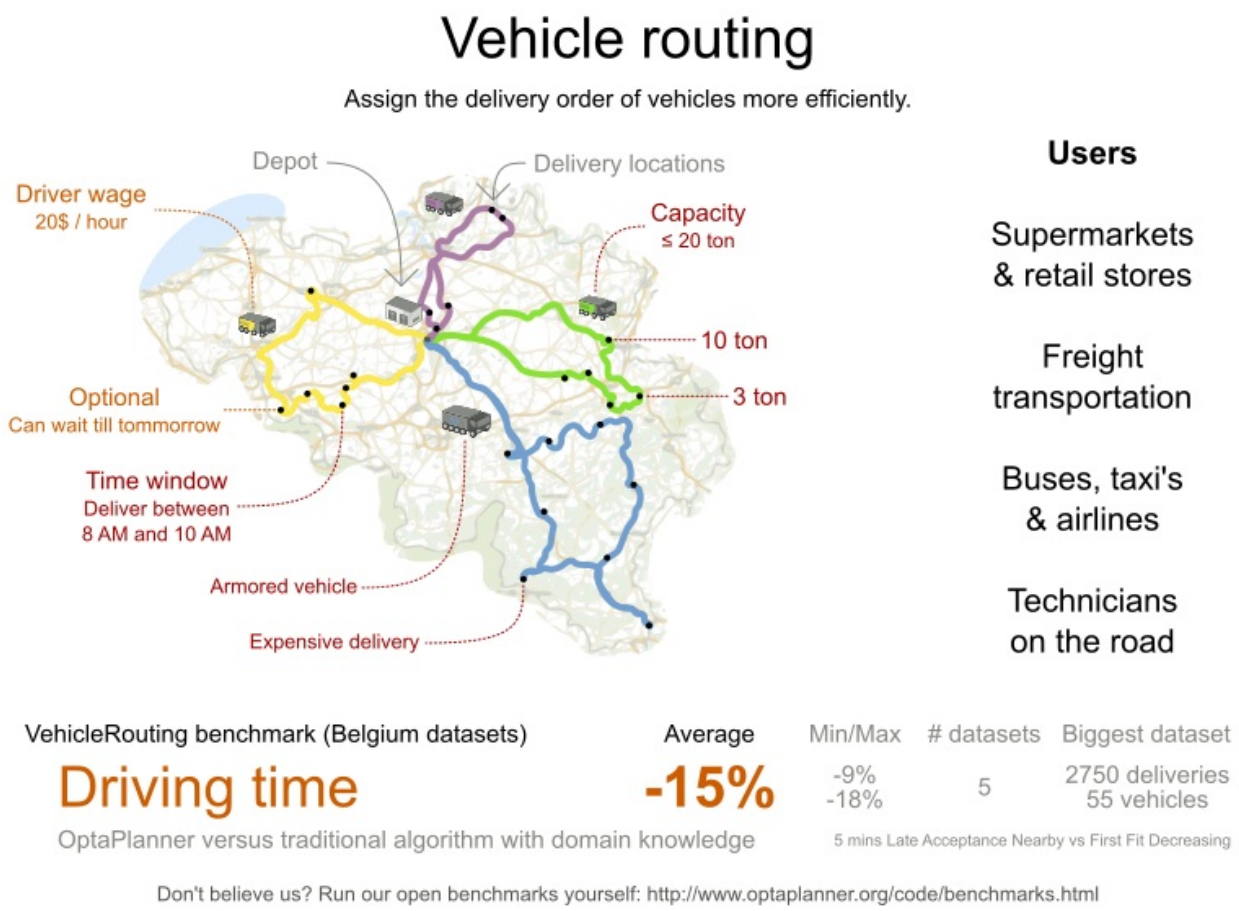
- *差时：从一个位置到另一个位置的差差时间。*
- *客户服务持续时间：消费者必须保持在服务期间的长度。*
- *客户准备时间：在客户准备时间之前，车辆可能已经进入，但必须等到提供时间后再提供服务。*
- *客户经过时限：在客户因时间而进行前，车辆必须经过时间。*

软限制：

- *总距离：让所有车上的总距离驱动（精简资源占用率）*

VRP Web 定义了容量设施设施路由问题(CVRP)及其时间序列(CVRPTW)。

图 31.8. 价值主张



问题大小

CVRP 实例（没有时间窗）：

> *belgium-n50-k10 has 1 depots, 10 vehicles and 49 customers with a search space of 10^74.*
> *belgium-n100-k10 has 1 depots, 10 vehicles and 99 customers with a search space of 10^170.*
> *belgium-n500-k20 has 1 depots, 20 vehicles and 499 customers with a search space of 10^1168.*
> *belgium-n1000-k20 has 1 depots, 20 vehicles and 999 customers with a search space of 10^2607.*
> *belgium-n2750-k55 has 1 depots, 55 vehicles and 2749 customers with a search space of 10^8380.*
> *belgium-road-km-n50-k10 has 1 depots, 10 vehicles and 49 customers with a search space of 10^74.*
> *belgium-road-km-n100-k10 has 1 depots, 10 vehicles and 99 customers with a search space of 10^170.*
> *belgium-road-km-n500-k20 has 1 depots, 20 vehicles and 499 customers with a search space of 10^1168.*
> *belgium-road-km-n1000-k20 has 1 depots, 20 vehicles and 999 customers with a search space of 10^2607.*
> *belgium-road-km-n2750-k55 has 1 depots, 55 vehicles and 2749 customers with a search space of 10^8380.*

belgium-road-time-n50-k10 has 1 depots, 10 vehicles and 49 customers with a search space of 10^74.
belgium-road-time-n100-k10 has 1 depots, 10 vehicles and 99 customers with a search space of 10^170.
belgium-road-time-n500-k20 has 1 depots, 20 vehicles and 499 customers with a search space of 10^1168.
belgium-road-time-n1000-k20 has 1 depots, 20 vehicles and 999 customers with a search space of 10^2607.
belgium-road-time-n2750-k55 has 1 depots, 55 vehicles and 2749 customers with a search space of 10^8380.
belgium-d2-n50-k10 has 2 depots, 10 vehicles and 48 customers with a search space of 10^74.
belgium-d3-n100-k10 has 3 depots, 10 vehicles and 97 customers with a search space of 10^170.
belgium-d5-n500-k20 has 5 depots, 20 vehicles and 495 customers with a search space of 10^1168.
belgium-d8-n1000-k20 has 8 depots, 20 vehicles and 992 customers with a search space of 10^2607.
belgium-d10-n2750-k55 has 10 depots, 55 vehicles and 2740 customers with a search space of 10^8380.

A-n32-k5 has 1 depots, 5 vehicles and 31 customers with a search space of 10^40.
A-n33-k5 has 1 depots, 5 vehicles and 32 customers with a search space of 10^41.
A-n33-k6 has 1 depots, 6 vehicles and 32 customers with a search space of 10^42.
A-n34-k5 has 1 depots, 5 vehicles and 33 customers with a search space of 10^43.
A-n36-k5 has 1 depots, 5 vehicles and 35 customers with a search space of 10^46.
A-n37-k5 has 1 depots, 5 vehicles and 36 customers with a search space of 10^48.
A-n37-k6 has 1 depots, 6 vehicles and 36 customers with a search space of 10^49.
A-n38-k5 has 1 depots, 5 vehicles and 37 customers with a search space of 10^49.
A-n39-k5 has 1 depots, 5 vehicles and 38 customers with a search space of 10^51.
A-n39-k6 has 1 depots, 6 vehicles and 38 customers with a search space of 10^52.
A-n44-k7 has 1 depots, 7 vehicles and 43 customers with a search space of 10^61.
A-n45-k6 has 1 depots, 6 vehicles and 44 customers with a search space of 10^62.
A-n45-k7 has 1 depots, 7 vehicles and 44 customers with a search space of 10^63.
A-n46-k7 has 1 depots, 7 vehicles and 45 customers with a search space of 10^65.
A-n48-k7 has 1 depots, 7 vehicles and 47 customers with a search space of 10^68.
A-n53-k7 has 1 depots, 7 vehicles and 52 customers with a search space of 10^77.
A-n54-k7 has 1 depots, 7 vehicles and 53 customers with a search space of 10^79.
A-n55-k9 has 1 depots, 9 vehicles and 54 customers with a search space of 10^82.
A-n60-k9 has 1 depots, 9 vehicles and 59 customers with a search space of 10^91.
A-n61-k9 has 1 depots, 9 vehicles and 60 customers with a search space of 10^93.
A-n62-k8 has 1 depots, 8 vehicles and 61 customers with a search space of 10^94.
A-n63-k9 has 1 depots, 9 vehicles and 62 customers with a search space of 10^97.
A-n63-k10 has 1 depots, 10 vehicles and 62 customers with a search space of 10^98.
A-n64-k9 has 1 depots, 9 vehicles and 63 customers with a search space of 10^99.
A-n65-k9 has 1 depots, 9 vehicles and 64 customers with a search space of 10^101.
A-n69-k9 has 1 depots, 9 vehicles and 68 customers with a search space of 10^108.
A-n80-k10 has 1 depots, 10 vehicles and 79 customers with a search space of 10^130.
F-n45-k4 has 1 depots, 4 vehicles and 44 customers with a search space of 10^60.
F-n72-k4 has 1 depots, 4 vehicles and 71 customers with a search space of 10^108.
F-n135-k7 has 1 depots, 7 vehicles and 134 customers with a search space of 10^240.

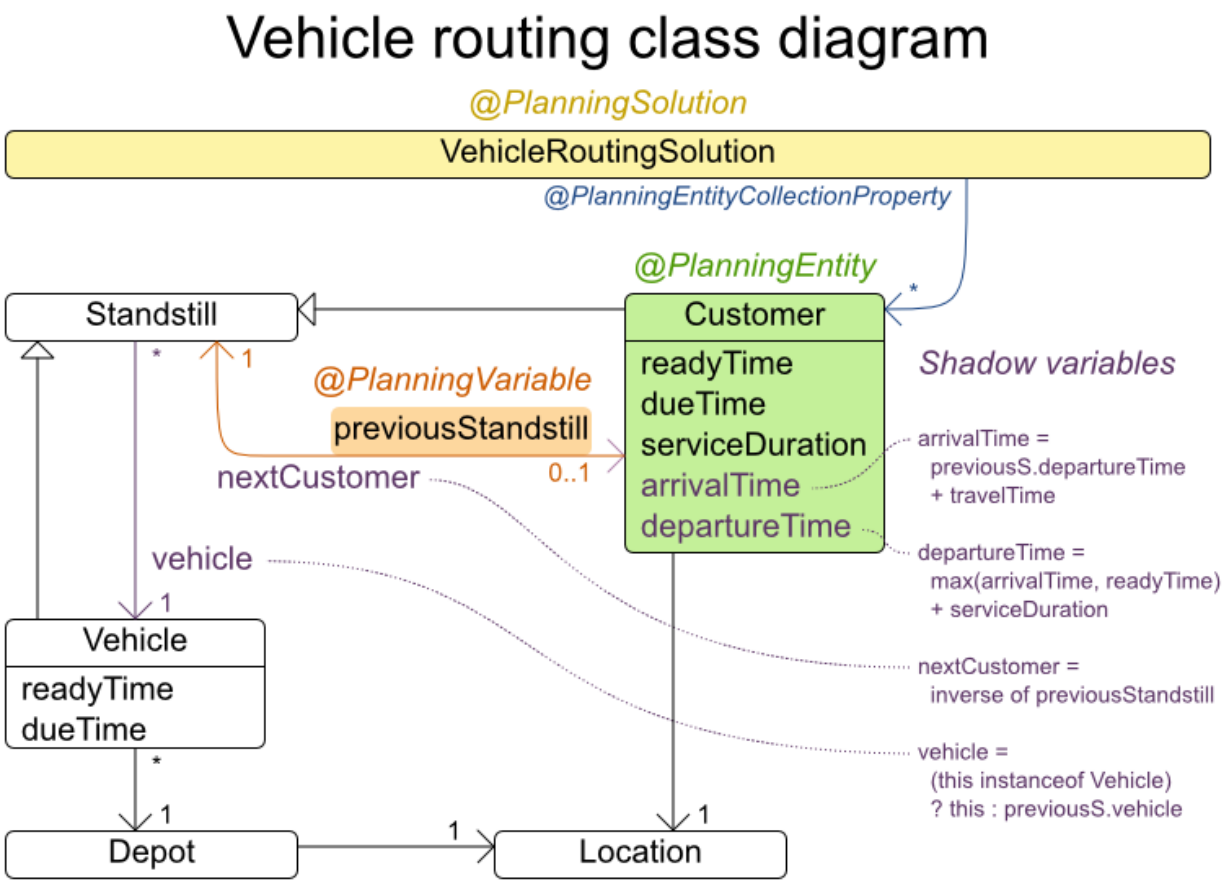
CVRP 两个实例（带时间窗）：

belgium-tw-d2-n50-k10 has 2 depots, 10 vehicles and 48 customers with a search space of 10^74.
belgium-tw-d3-n100-k10 has 3 depots, 10 vehicles and 97 customers with a search space of 10^170.
belgium-tw-d5-n500-k20 has 5 depots, 20 vehicles and 495 customers with a search space of 10^1168.
belgium-tw-d8-n1000-k20 has 8 depots, 20 vehicles and 992 customers with a search space of 10^2607.
belgium-tw-d10-n2750-k55 has 10 depots, 55 vehicles and 2740 customers with a search space of 10^8380.
belgium-tw-n50-k10 has 1 depots, 10 vehicles and 49 customers with a search space of 10^74.
belgium-tw-n100-k10 has 1 depots, 10 vehicles and 99 customers with a search space of 10^170.
belgium-tw-n500-k20 has 1 depots, 20 vehicles and 499 customers with a search space of 10^1168.
belgium-tw-n1000-k20 has 1 depots, 20 vehicles and 999 customers with a search space of 10^2607.
belgium-tw-n2750-k55 has 1 depots, 55 vehicles and 2749 customers with a search space of 10^8380.

Solomon_025_C101 has 1 depots, 25 vehicles and 25 customers with a search space of 10^40.
Solomon_025_C201 has 1 depots, 25 vehicles and 25 customers with a search space of 10^40.
Solomon_025_R101 has 1 depots, 25 vehicles and 25 customers with a search space of 10^40.
Solomon_025_R201 has 1 depots, 25 vehicles and 25 customers with a search space of 10^40.
Solomon_025_RC101 has 1 depots, 25 vehicles and 25 customers with a search space of 10^40.
Solomon_025_RC201 has 1 depots, 25 vehicles and 25 customers with a search space of 10^40.
Solomon_100_C101 has 1 depots, 25 vehicles and 100 customers with a search space of 10^185.
Solomon_100_C201 has 1 depots, 25 vehicles and 100 customers with a search space of 10^185.
Solomon_100_R101 has 1 depots, 25 vehicles and 100 customers with a search space of 10^185.
Solomon_100_R201 has 1 depots, 25 vehicles and 100 customers with a search space of 10^185.
Solomon_100_RC101 has 1 depots, 25 vehicles and 100 customers with a search space of 10^185.
Solomon_100_RC201 has 1 depots, 25 vehicles and 100 customers with a search space of 10^185.
Homberger_0200_C1_2_1 has 1 depots, 50 vehicles and 200 customers with a search space of 10^429.
Homberger_0200_C2_2_1 has 1 depots, 50 vehicles and 200 customers with a search space of 10^429.
Homberger_0200_R1_2_1 has 1 depots, 50 vehicles and 200 customers with a search space of 10^429.
Homberger_0200_R2_2_1 has 1 depots, 50 vehicles and 200 customers with a search space of 10^429.
Homberger_0200_RC1_2_1 has 1 depots, 50 vehicles and 200 customers with a search space of 10^429.
Homberger_0200_RC2_2_1 has 1 depots, 50 vehicles and 200 customers with a search space of

10^429.
Homberger_0400_C1_4_1 has 1 depots, 100 vehicles and 400 customers with a search space of 10^978.
Homberger_0400_C2_4_1 has 1 depots, 100 vehicles and 400 customers with a search space of 10^978.
Homberger_0400_R1_4_1 has 1 depots, 100 vehicles and 400 customers with a search space of 10^978.
Homberger_0400_R2_4_1 has 1 depots, 100 vehicles and 400 customers with a search space of 10^978.
Homberger_0400_RC1_4_1 has 1 depots, 100 vehicles and 400 customers with a search space of 10^978.
Homberger_0400_RC2_4_1 has 1 depots, 100 vehicles and 400 customers with a search space of 10^978.
Homberger_0600_C1_6_1 has 1 depots, 150 vehicles and 600 customers with a search space of 10^1571.
Homberger_0600_C2_6_1 has 1 depots, 150 vehicles and 600 customers with a search space of 10^1571.
Homberger_0600_R1_6_1 has 1 depots, 150 vehicles and 600 customers with a search space of 10^1571.
Homberger_0600_R2_6_1 has 1 depots, 150 vehicles and 600 customers with a search space of 10^1571.
Homberger_0600_RC1_6_1 has 1 depots, 150 vehicles and 600 customers with a search space of 10^1571.
Homberger_0600_RC2_6_1 has 1 depots, 150 vehicles and 600 customers with a search space of 10^1571.
Homberger_0800_C1_8_1 has 1 depots, 200 vehicles and 800 customers with a search space of 10^2195.
Homberger_0800_C2_8_1 has 1 depots, 200 vehicles and 800 customers with a search space of 10^2195.
Homberger_0800_R1_8_1 has 1 depots, 200 vehicles and 800 customers with a search space of 10^2195.
Homberger_0800_R2_8_1 has 1 depots, 200 vehicles and 800 customers with a search space of 10^2195.
Homberger_0800_RC1_8_1 has 1 depots, 200 vehicles and 800 customers with a search space of 10^2195.
Homberger_0800_RC2_8_1 has 1 depots, 200 vehicles and 800 customers with a search space of 10^2195.
Homberger_1000_C110_1 has 1 depots, 250 vehicles and 1000 customers with a search space of 10^2840.
Homberger_1000_C210_1 has 1 depots, 250 vehicles and 1000 customers with a search space of 10^2840.
Homberger_1000_R110_1 has 1 depots, 250 vehicles and 1000 customers with a search space of 10^2840.
Homberger_1000_R210_1 has 1 depots, 250 vehicles and 1000 customers with a search space of 10^2840.
Homberger_1000_RC110_1 has 1 depots, 250 vehicles and 1000 customers with a search space of 10^2840.
Homberger_1000_RC210_1 has 1 depots, 250 vehicles and 1000 customers with a search space of 10^2840.

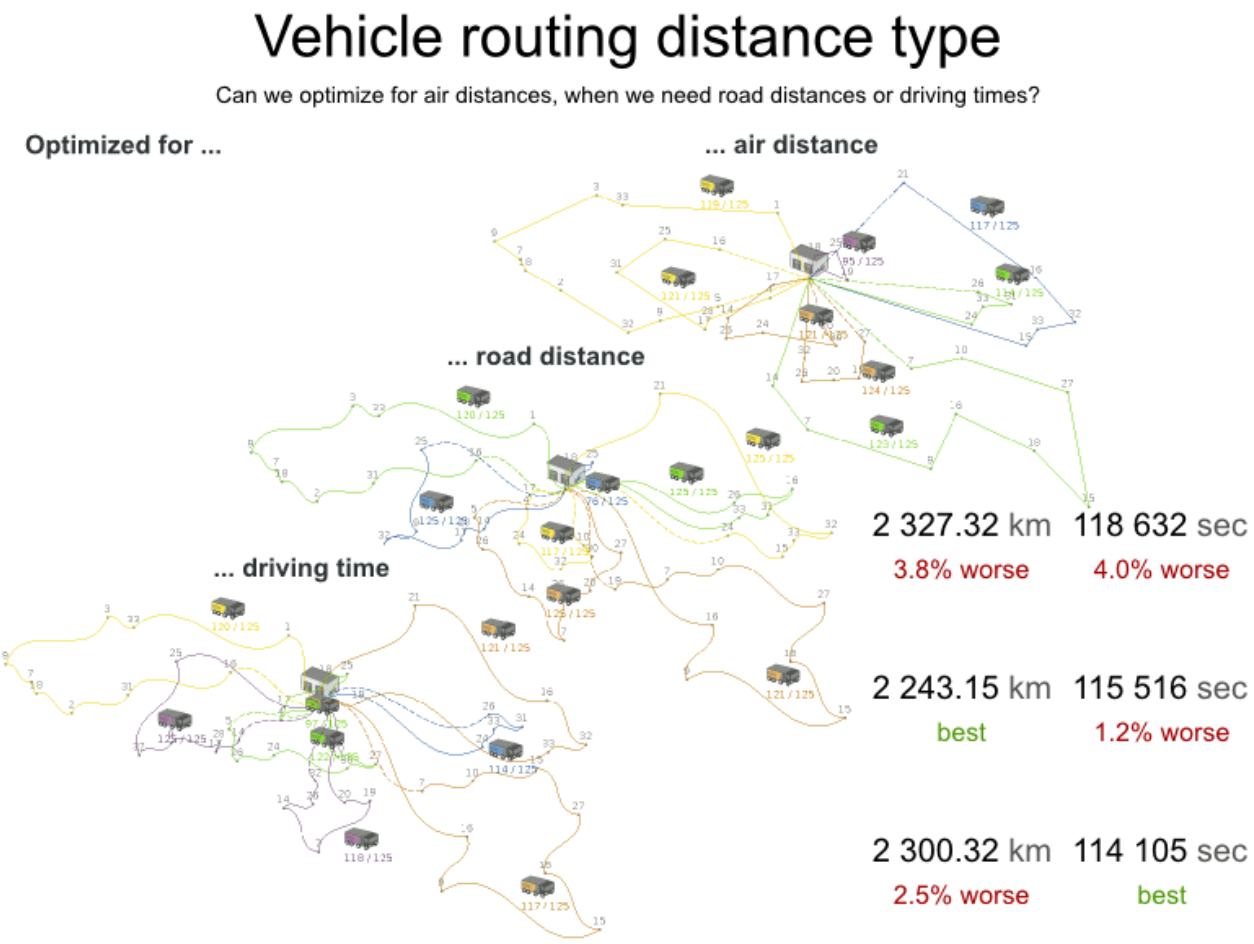
31.13.1. Vehicle 路由的域模型



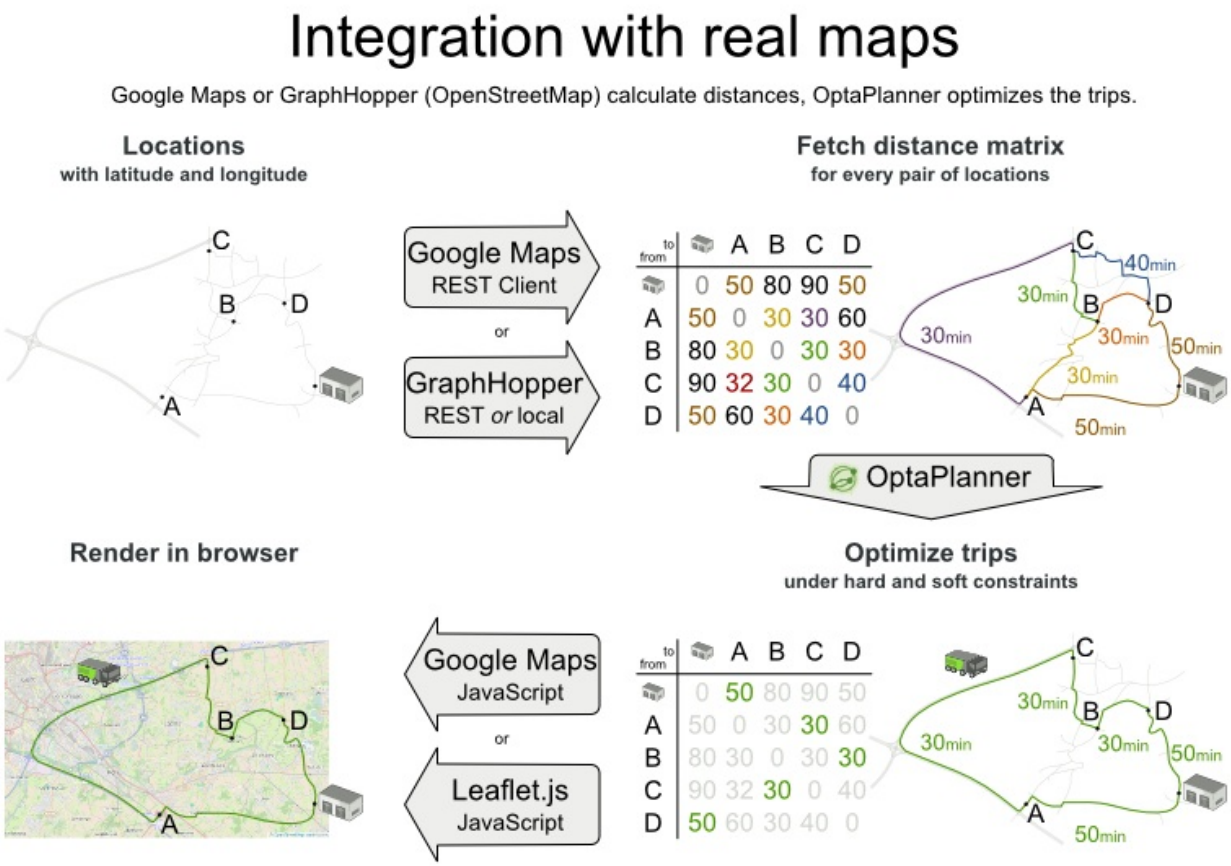
使用时间窗域模型的载体路由大量使用 shadow 变量功能。这使得它能够更自然地表达其限制，因为 arrivalTime 和 departureTime 等属性可以在域模型上直接可用。

巴林林特利斯特利斯特区

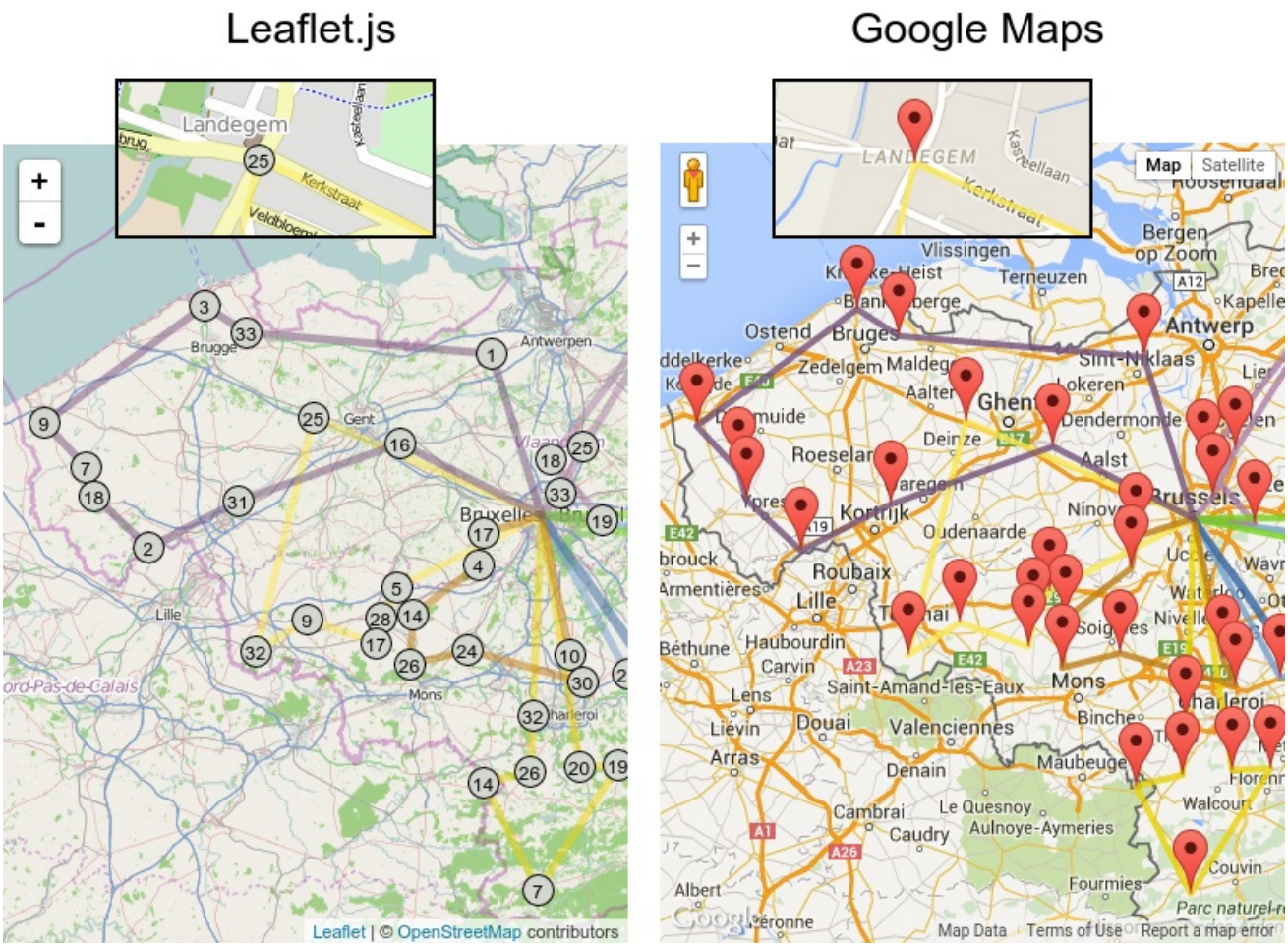
在现实世界中，车辆不能直接跟随位置到位置：他们必须使用路路和高路。从业务的角度来说，这很重要：



对于优化算法，这并不重要，只要可以查找两个点之间的距离（并最好预先计算）。未来成本甚至不需要成为距离。它还可以是差时间、增加成本或更加权的功能。有几种方法可以预先计算出您的成本，如 GraphHopper （嵌入式、离线 Java 引擎）、 Open MapQuest （Web 服务）和 Google Maps 客户端 API （Web 服务）。

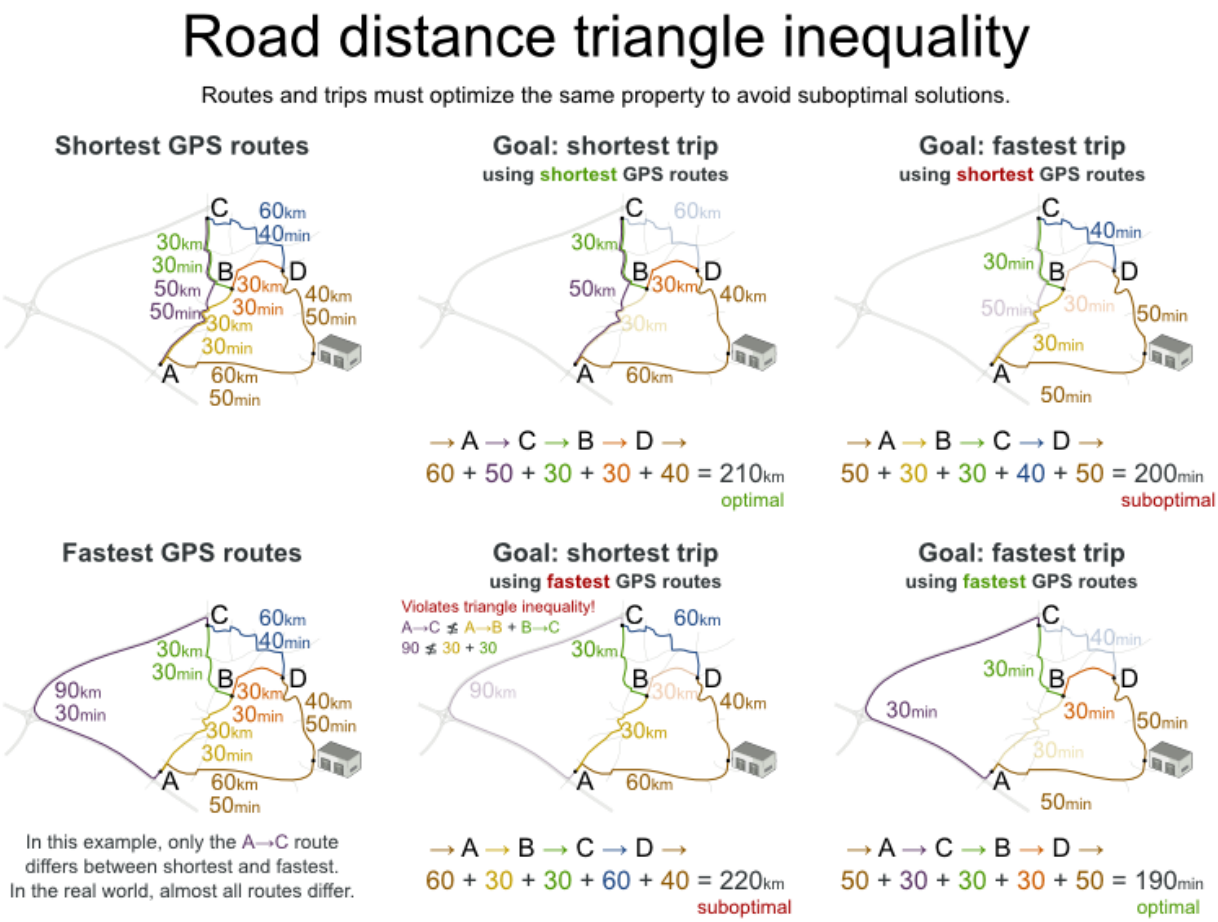


还有一些技术 可以呈现它，例如针对开发人员的 Leaflet 和 Google Maps。



甚至可以使用 GraphHopper 或 Google Map Directions 呈现实际规划路由，但是由于路由在高路上的重叠，它可能会变得难以看到反引号：

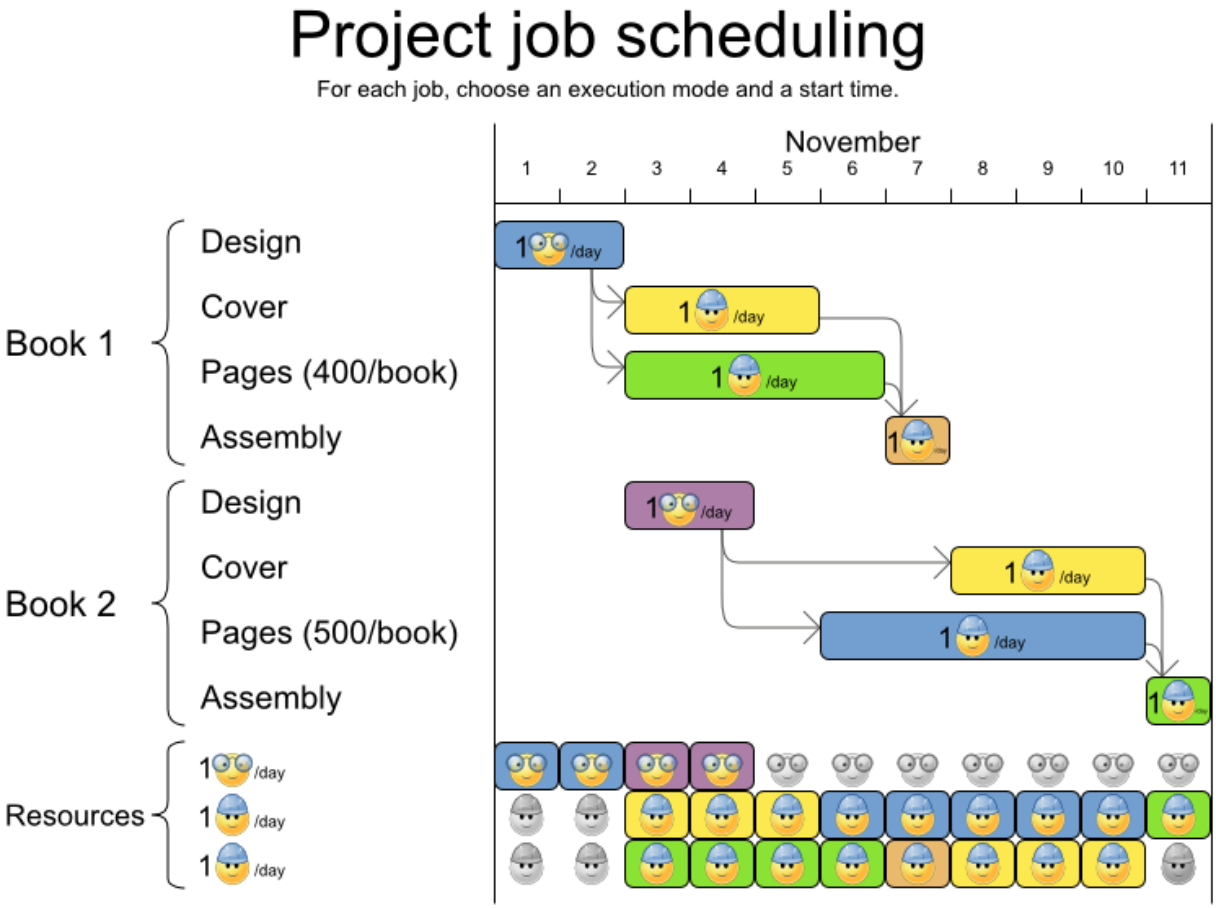
请特别小心，在两个点之间采用与 OptaPlanner 中使用的相同优化标准。例如： GraphHopper 默认返回最快的路由，而不是最短的路由。不要使用 km（或 miles）最快的 GPS 路由来优化 OptaPlanner 中最短的差差：



与热门相对应，大多数用户都不希望最短的路由：他们想要使用最快的路由。它们比一般的路高。它们优先于有变化的障碍。在现实世界中，最快且最短的路由很少是相同的。

31.14. 项目作业调度

以时间和执行模式调度所有作业，以最小化项目延迟。每个作业都是项目的一部分。作业可以通过不同的方式执行：每个方法都是一种执行模式，它代表了不同的持续时间，但也可以使用不同的资源。这是一种灵活的 作业商店调度 的一种形式。



硬约束：

- *作业优先级：作业只能在其所有前身作业完成后启动。*
- *资源容量：不要使用超过可用的资源。*
  - *资源是本地（在同一项目中的作业之间共享）或全局（在所有作业间共享）*
  - *资源是可续订（每天可用容量）或不可续订（所有天可用的容量）*

***Medium** 约束：*

- *项目延迟总量：尽量减少每个项目的持续时间(makespan)。*

软限制：

- ***Total makespan：最小化整个多项目调度的持续时间。***

该问题由 MISTA 2013 挑战 定义。

问题大小

> *Schedule A-1 has 2 projects, 24 jobs, 64 execution modes, 7 resources and 150 resource requirements.*
> *Schedule A-2 has 2 projects, 44 jobs, 124 execution modes, 7 resources and 420 resource requirements.*
> *Schedule A-3 has 2 projects, 64 jobs, 184 execution modes, 7 resources and 630 resource requirements.*
> *Schedule A-4 has 5 projects, 60 jobs, 160 execution modes, 16 resources and 390 resource requirements.*
> *Schedule A-5 has 5 projects, 110 jobs, 310 execution modes, 16 resources and 900 resource requirements.*
> *Schedule A-6 has 5 projects, 160 jobs, 460 execution modes, 16 resources and 1440 resource requirements.*
> *Schedule A-7 has 10 projects, 120 jobs, 320 execution modes, 22 resources and 900 resource requirements.*
> *Schedule A-8 has 10 projects, 220 jobs, 620 execution modes, 22 resources and 1860 resource requirements.*
> *Schedule A-9 has 10 projects, 320 jobs, 920 execution modes, 31 resources and 2880 resource requirements.*
> *Schedule A-10 has 10 projects, 320 jobs, 920 execution modes, 31 resources and 2970 resource requirements.*
> *Schedule B-1 has 10 projects, 120 jobs, 320 execution modes, 31 resources and 900 resource requirements.*
> *Schedule B-2 has 10 projects, 220 jobs, 620 execution modes, 22 resources and 1740 resource requirements.*
> *Schedule B-3 has 10 projects, 320 jobs, 920 execution modes, 31 resources and 3060 resource requirements.*
> *Schedule B-4 has 15 projects, 180 jobs, 480 execution modes, 46 resources and 1530 resource requirements.*
> *Schedule B-5 has 15 projects, 330 jobs, 930 execution modes, 46 resources and 2760 resource requirements.*
> *Schedule B-6 has 15 projects, 480 jobs, 1380 execution modes, 46 resources and 4500 resource requirements.*
> *Schedule B-7 has 20 projects, 240 jobs, 640 execution modes, 61 resources and 1710 resource requirements.*
> *Schedule B-8 has 20 projects, 440 jobs, 1240 execution modes, 42 resources and 3180 resource requirements.*
> *Schedule B-9 has 20 projects, 640 jobs, 1840 execution modes, 61 resources and 5940 resource*

> requirements.
> Schedule B-10 has 20 projects, 460 jobs, 1300 execution modes, 42 resources and 4260 resource requirements.

31.15. 任务分配

将每个任务分配到员工队列中的 spot。每个任务都有一个持续时间，它受到员工的关联性级别的影响。

硬约束：

- 技能：每个任务都需要一个或多个技能。员工必须拥有所有这些技能。

软级别 0 约束：

- Critical 任务：首先完成关键的任务，比主要任务和次要任务快。

软级别 1 的限制：

- 最小化 makespan：缩短完成所有任务的时间。
  - 首先，首先从工作最多的工作员工开始，即员工数量最长，从而创建公平和负载平衡。

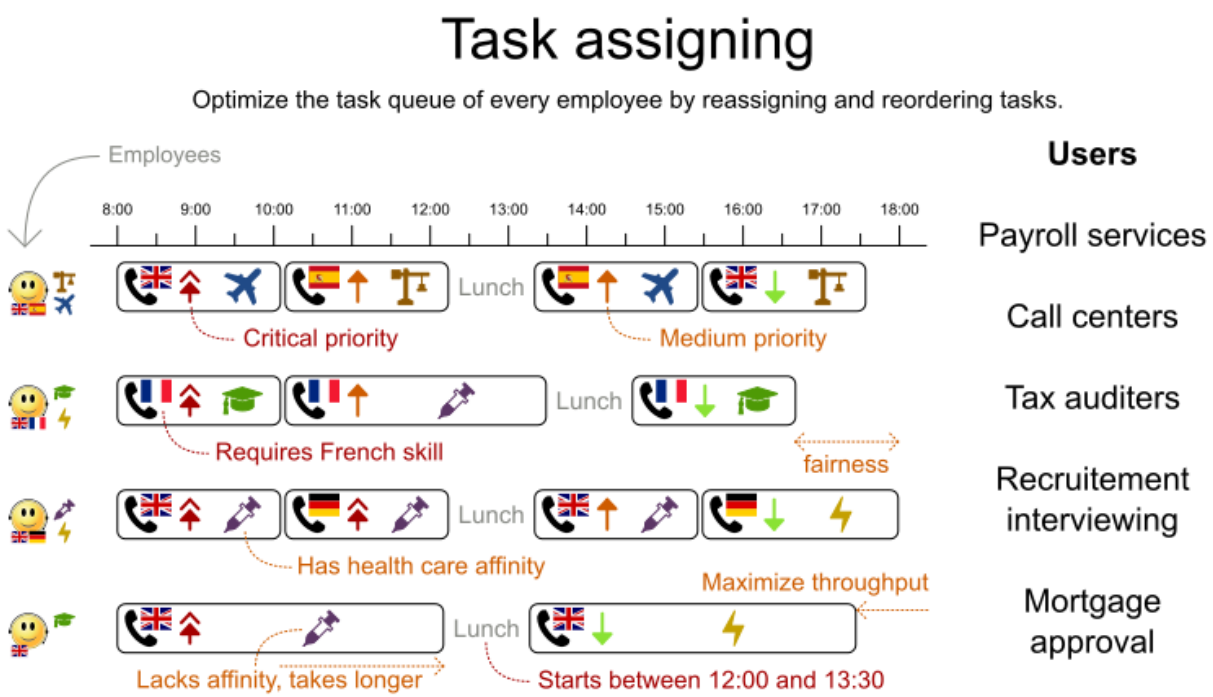
软级别 2 约束：

- 主要任务：在可能的情况下尽快完成主要任务，比小任务更快。

软级别 3 的限制：

- *次要任务：尽快完成次要任务。*

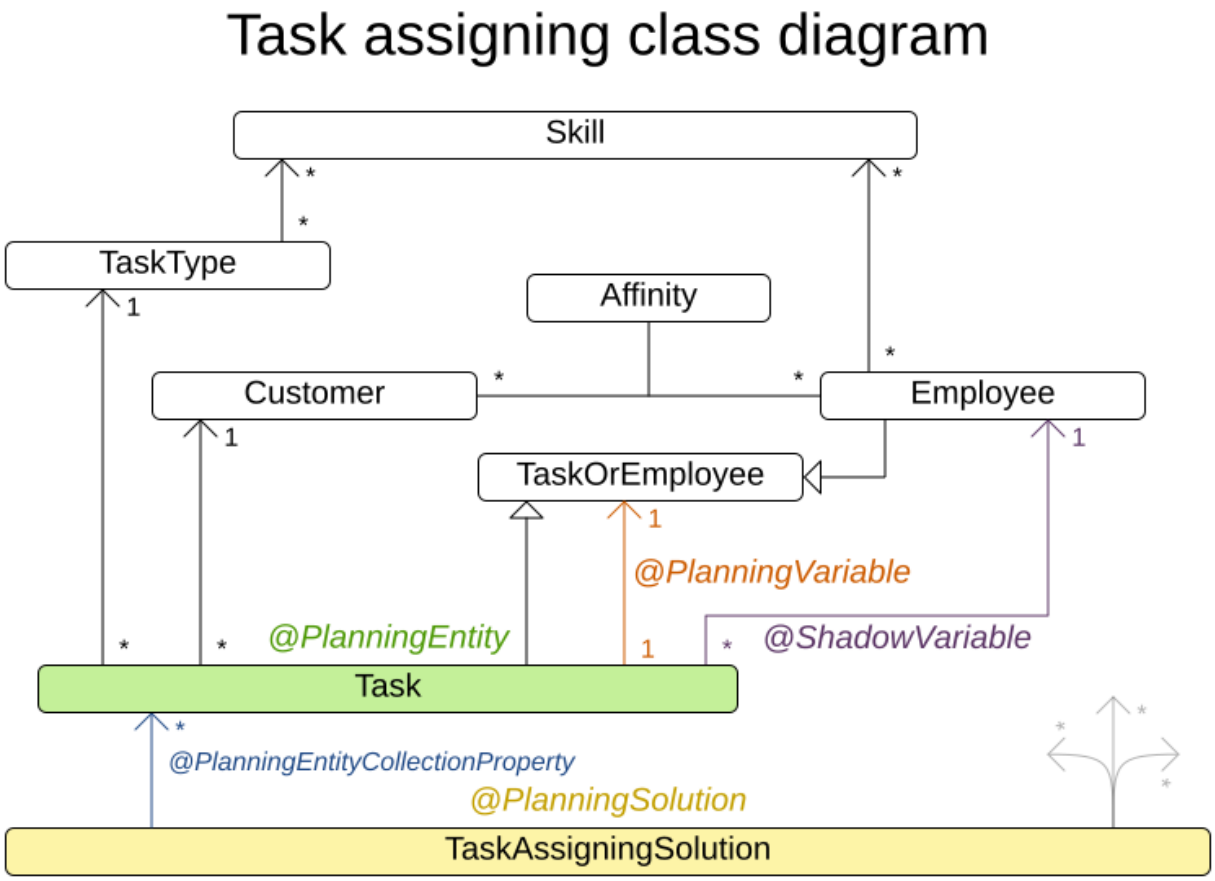
图 31.9. 价值主张



问题大小

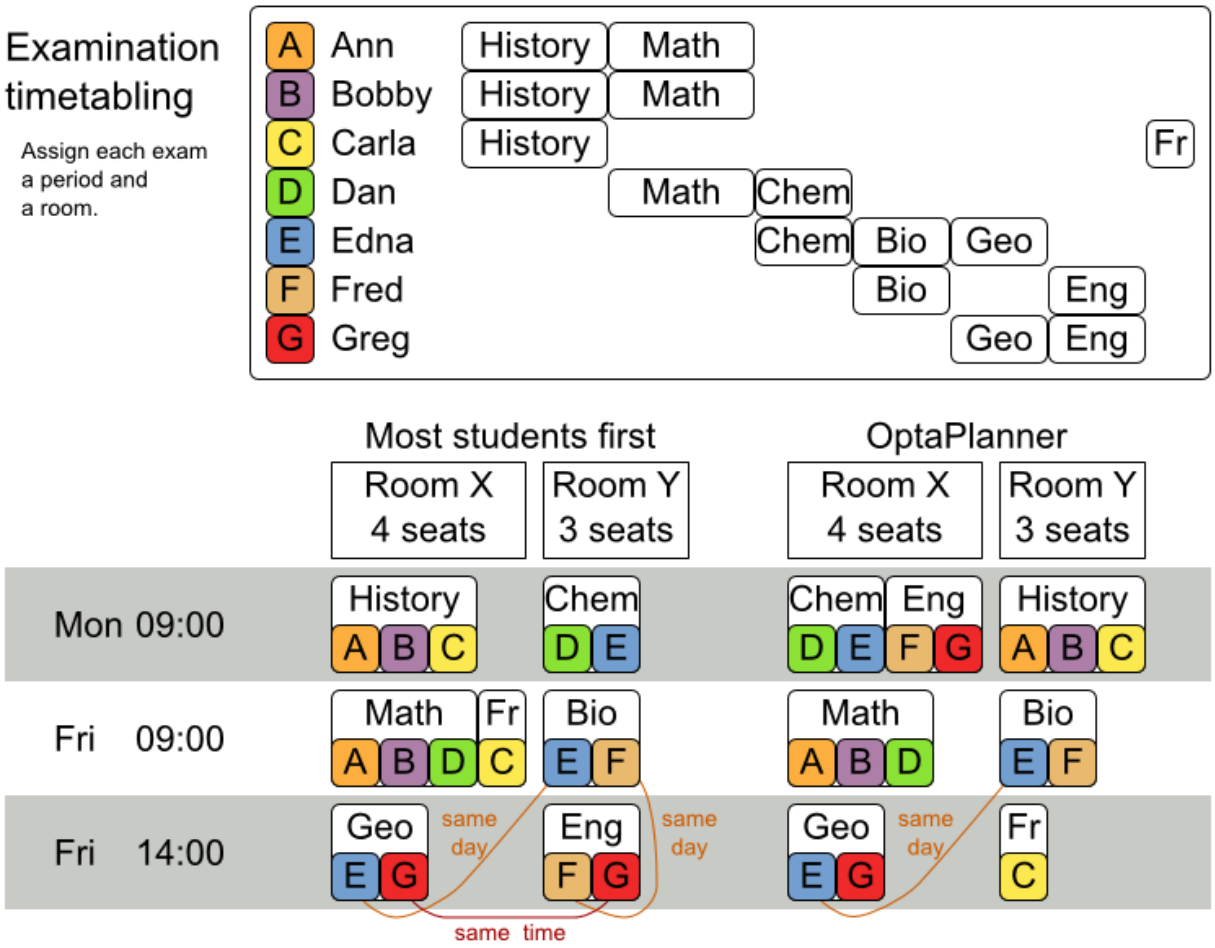
> *24tasks-8employees has 24 tasks, 6 skills, 8 employees, 4 task types and 4 customers with a search space of 10^30.*
> *50tasks-5employees has 50 tasks, 5 skills, 5 employees, 10 task types and 10 customers with a search space of 10^69.*
> *100tasks-5employees has 100 tasks, 5 skills, 5 employees, 20 task types and 15 customers with a search space of 10^164.*
> *500tasks-20employees has 500 tasks, 6 skills, 20 employees, 100 task types and 60 customers with a search space of 10^1168.*

图 31.10. 域模型



31.16. 考试时间表（ITC 2007 年跟踪 1 - 考试）

将各考试安排为时段和上门。多个考试可以在同一期间内共享相同的空间。



硬约束：

- *考试冲突：必须在同一时间段内共享学员的两个考试。*
- *房间容量：房座容量必须随时可用。*
- *期间持续时间：其所有考试的持续时间必须有效。*
- *相关的硬限制（为每个数据集指定）：*
  - *coindence：两个指定的考试必须使用相同的期限（但可能还会使用其它课程）。*
  - *排除：两个指定的考试不得使用相同的期限。*

- 后：在另一个指定的考试期后，必须在一段时间内进行指定的考试。

- 相关硬约束（为每个数据集指定）：
  - 排行：一个指定的考试不必与任何其他考试共享其空间。

软限制（其各自具有重要优势）：

- 同一人不能连续有两个考试。
- 同一人不能同时拥有两个考试。
- 周期分布：共享学员的两个考试应该是间隔很多句点。
- 混合持续时间：共享房间应该没有不同持续时间的两种考试。
- 前端加载：在计划前面应预先调度大型考试。
- 周期 penalty（每个数据集指定）：有些句点（使用时指定句点）。
- room penalty（每个数据集指定）：有些房间在使用时有感激。

它使用大量实时测试数据集。

这个问题由 国际时间选项卡 2007 年跟踪 1 定义。Geoffrey De Smet 在竞争中，使用非常早的 OptaPlanner 版本。然后，从那时起已进行了很多改进。

问题大小

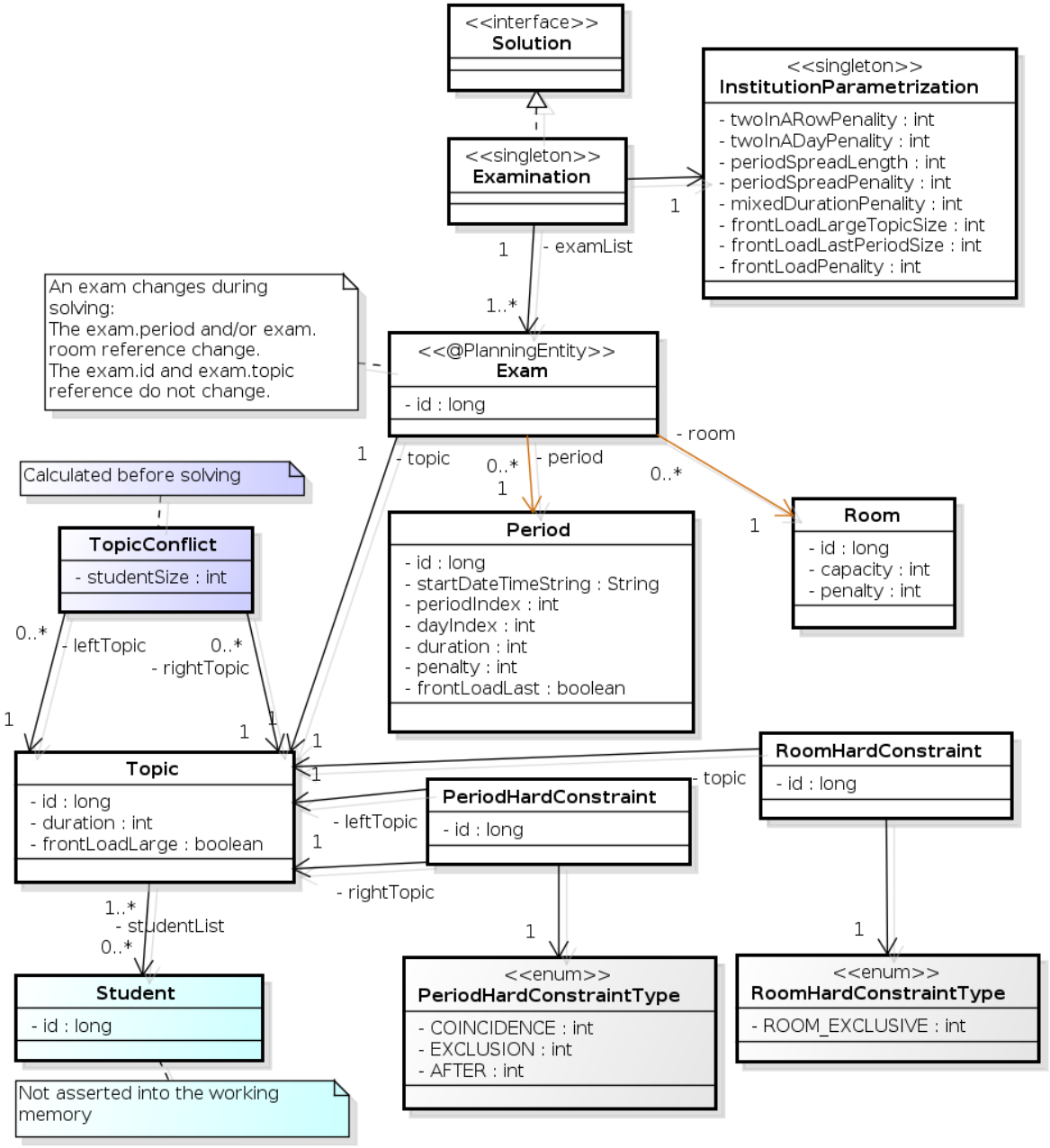
-

exam_comp_set1 has 7883 students, 607 exams, 54 periods, 7 rooms, 12 period constraints and 0 room constraints with a search space of 10^1564.
exam_comp_set2 has 12484 students, 870 exams, 40 periods, 49 rooms, 12 period constraints and 2 room constraints with a search space of 10^2864.
exam_comp_set3 has 16365 students, 934 exams, 36 periods, 48 rooms, 168 period constraints and 15 room constraints with a search space of 10^3023.
exam_comp_set4 has 4421 students, 273 exams, 21 periods, 1 rooms, 40 period constraints and 0 room constraints with a search space of 10^360.
exam_comp_set5 has 8719 students, 1018 exams, 42 periods, 3 rooms, 27 period constraints and 0 room constraints with a search space of 10^2138.
exam_comp_set6 has 7909 students, 242 exams, 16 periods, 8 rooms, 22 period constraints and 0 room constraints with a search space of 10^509.
exam_comp_set7 has 13795 students, 1096 exams, 80 periods, 15 rooms, 28 period constraints and 0 room constraints with a search space of 10^3374.
exam_comp_set8 has 7718 students, 598 exams, 80 periods, 8 rooms, 20 period constraints and 1 room constraints with a search space of 10^1678.

31.16.1. 用于考试时间设置的域模型

下图显示了考试的主要课程：

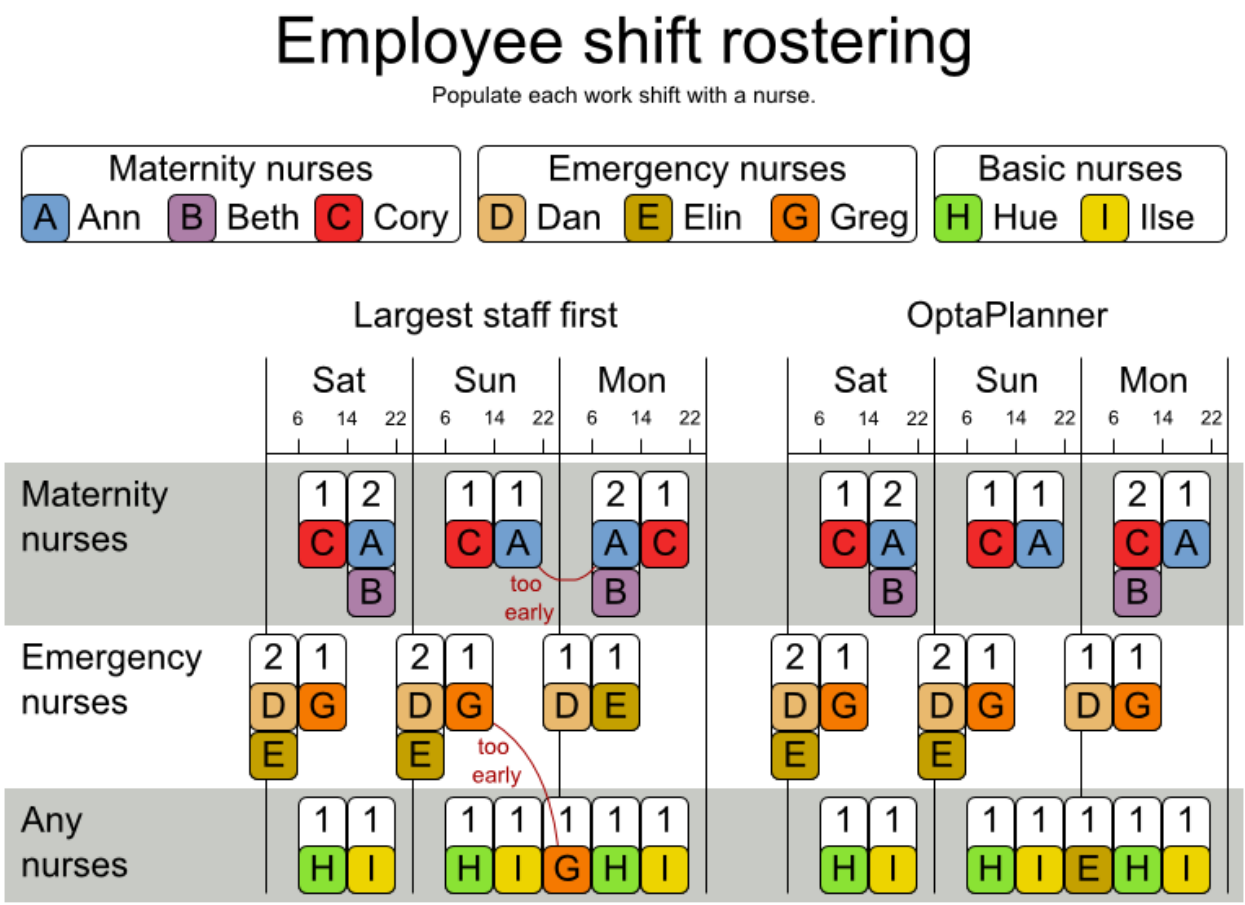
图 31.11. 检查域类图



请注意，我们已将考试概念分成 考试 课程和主题 课程。在解决过程中，考试 实例会改变（这是计划 实体类），当它们的期间或房间属性改变时。主题、Period 和 Room 实例在解决过程中不会发生改变（它们有问题的事实，就像其他一些课程一样）。

31.17. NURSE ROSTER(2010INRC 2010)

对于每个转变，请安排相关变化。



硬约束：

- *无未分配的转变（内置）：所有变化都需要分配给员工。*
- *发生冲突：员工每天只能进行一次转换。*

软限制：

- *合同义务。业务经常违反这些情况，因此他们决定将这些内容定义为软限制，而不是硬约束。*
  - *最小和最大分配：每个员工需要工作超过 x 个变化，超过 y 个变化（取决于其合约）。*
  - *最少和连续工作天数：每个员工需要在一行中的 x 到 y 天（取决于其合约）之间工*

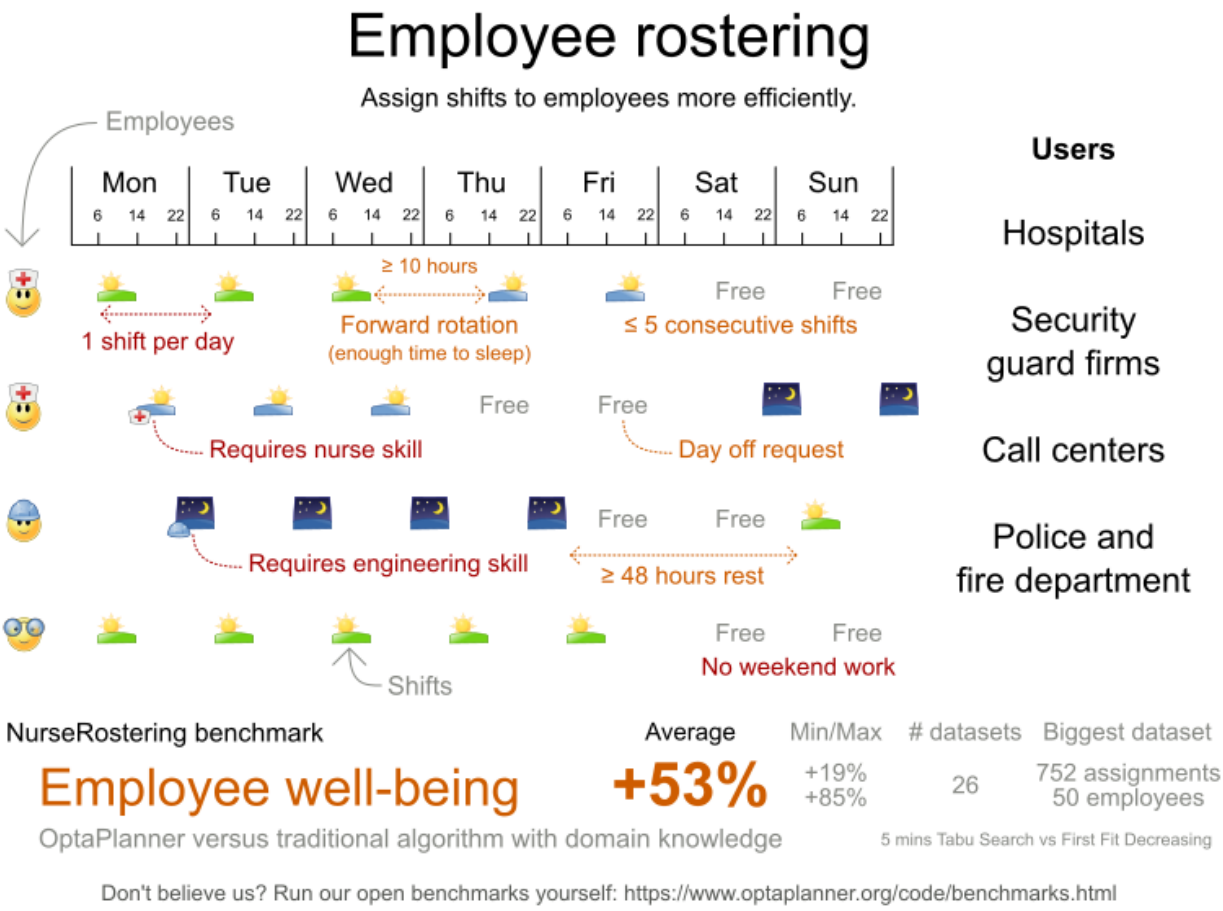
作。

- 最少和最多的空闲天 ：每个员工都需要在一行中的 x 到 y 天之间释放（取决于其合约）。
- 最小和最大连续工作每周 ：每个员工在一行中的 x 和 y 周期限之间工作（取决于其合约）。
- 完整的周末 ：每个员工都需要在一个周末或根本上每天工作。
- 周末时相同的转换类型 ：对于同一员工同一周末的每周发生变化都必须是相同的转变类型。
- 不需要的模式 ：一行中不需要的转变类型组合，例如，较早的转变，后再进行较晚的转变。

- 雇员：
  - 申请 当日：员工希望处理某个特定天。
  - 天下请求 ：员工不希望在特定一天上工作。
  - 申请 转变：员工希望分配到特定的转变。
  - 转移请求 ：员工不想分配到特定的转变。
- 备选技能 ：分配给技能的员工应掌握这种转变所需的各项技能。

这个问题由 2010 年国际 Nurse Rosterionion 定义。

图 31.12. 价值主张



问题大小

有三种 dataset 类型：

- ***Sprint：必须以秒为单位解决。***
- ***Medium: 必须以分钟为单位解决。***
- ***long：必须在几小时内解决。***

> *toy1 has 1 skills, 3 shiftTypes, 2 patterns, 1 contracts, 6 employees, 7 shiftDates, 35 shiftAssignments and 0 requests with a search space of 10^27.*
> *toy2 has 1 skills, 3 shiftTypes, 3 patterns, 2 contracts, 20 employees, 28 shiftDates, 180 shiftAssignments and 140 requests with a search space of 10^234.*
>
> *sprint01 has 1 skills, 4 shiftTypes, 3 patterns, 4 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 150 requests with a search space of 10^152.*
> *sprint02 has 1 skills, 4 shiftTypes, 3 patterns, 4 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 150 requests with a search space of 10^152.*

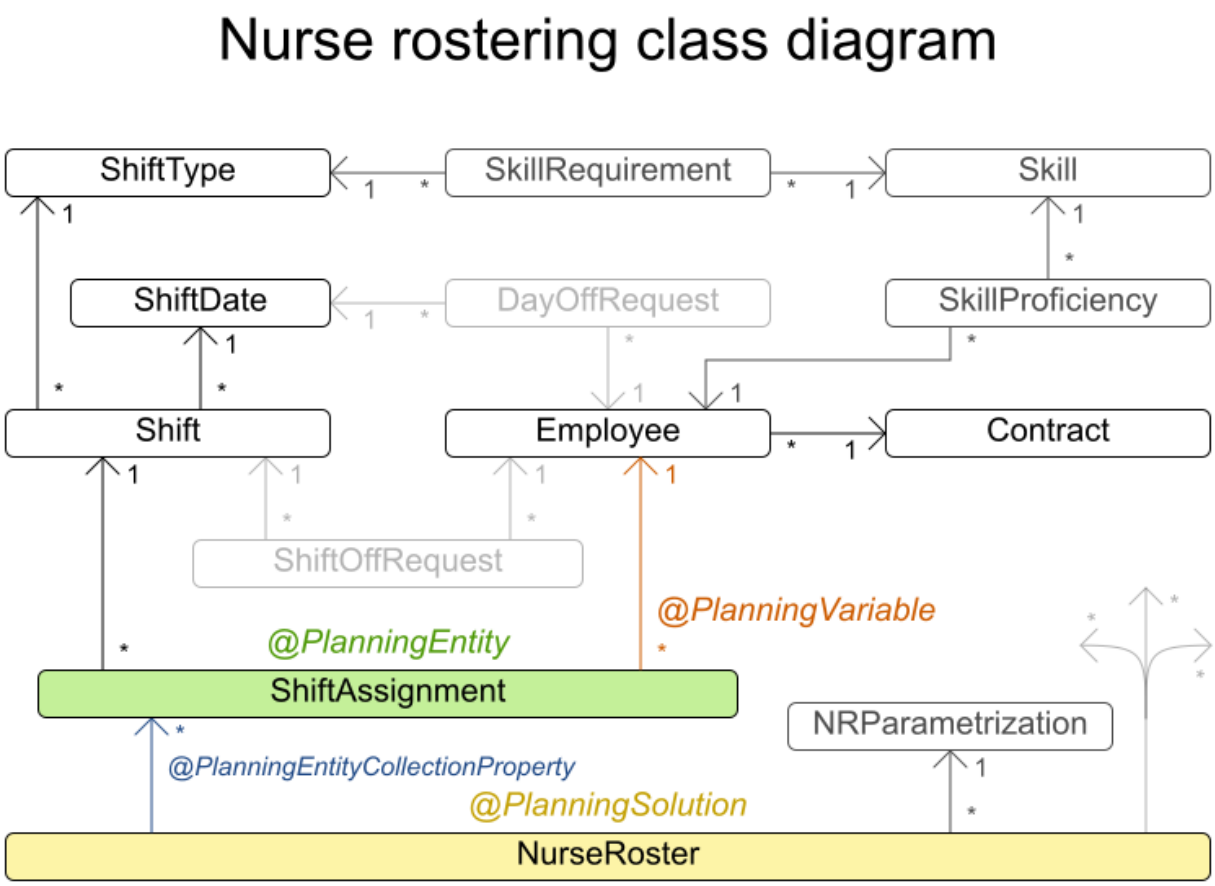
sprint03 has 1 skills, 4 shiftTypes, 3 patterns, 4 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 150 requests with a search space of 10^152.
sprint04 has 1 skills, 4 shiftTypes, 3 patterns, 4 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 150 requests with a search space of 10^152.
sprint05 has 1 skills, 4 shiftTypes, 3 patterns, 4 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 150 requests with a search space of 10^152.
sprint06 has 1 skills, 4 shiftTypes, 3 patterns, 4 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 150 requests with a search space of 10^152.
sprint07 has 1 skills, 4 shiftTypes, 3 patterns, 4 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 150 requests with a search space of 10^152.
sprint08 has 1 skills, 4 shiftTypes, 3 patterns, 4 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 150 requests with a search space of 10^152.
sprint09 has 1 skills, 4 shiftTypes, 3 patterns, 4 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 150 requests with a search space of 10^152.
sprint10 has 1 skills, 4 shiftTypes, 3 patterns, 4 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 150 requests with a search space of 10^152.
sprint_hint01 has 1 skills, 4 shiftTypes, 8 patterns, 3 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 150 requests with a search space of 10^152.
sprint_hint02 has 1 skills, 4 shiftTypes, 0 patterns, 3 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 150 requests with a search space of 10^152.
sprint_hint03 has 1 skills, 4 shiftTypes, 8 patterns, 3 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 150 requests with a search space of 10^152.
sprint_late01 has 1 skills, 4 shiftTypes, 8 patterns, 3 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 150 requests with a search space of 10^152.
sprint_late02 has 1 skills, 3 shiftTypes, 4 patterns, 3 contracts, 10 employees, 28 shiftDates, 144 shiftAssignments and 139 requests with a search space of 10^144.
sprint_late03 has 1 skills, 4 shiftTypes, 8 patterns, 3 contracts, 10 employees, 28 shiftDates, 160 shiftAssignments and 150 requests with a search space of 10^160.
sprint_late04 has 1 skills, 4 shiftTypes, 8 patterns, 3 contracts, 10 employees, 28 shiftDates, 160 shiftAssignments and 150 requests with a search space of 10^160.
sprint_late05 has 1 skills, 4 shiftTypes, 8 patterns, 3 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 150 requests with a search space of 10^152.
sprint_late06 has 1 skills, 4 shiftTypes, 0 patterns, 3 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 150 requests with a search space of 10^152.
sprint_late07 has 1 skills, 4 shiftTypes, 0 patterns, 3 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 150 requests with a search space of 10^152.
sprint_late08 has 1 skills, 4 shiftTypes, 0 patterns, 3 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 0 requests with a search space of 10^152.
sprint_late09 has 1 skills, 4 shiftTypes, 0 patterns, 3 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 0 requests with a search space of 10^152.
sprint_late10 has 1 skills, 4 shiftTypes, 0 patterns, 3 contracts, 10 employees, 28 shiftDates, 152 shiftAssignments and 150 requests with a search space of 10^152.

medium01 has 1 skills, 4 shiftTypes, 0 patterns, 4 contracts, 31 employees, 28 shiftDates, 608 shiftAssignments and 403 requests with a search space of 10^906.
medium02 has 1 skills, 4 shiftTypes, 0 patterns, 4 contracts, 31 employees, 28 shiftDates, 608 shiftAssignments and 403 requests with a search space of 10^906.
medium03 has 1 skills, 4 shiftTypes, 0 patterns, 4 contracts, 31 employees, 28 shiftDates, 608 shiftAssignments and 403 requests with a search space of 10^906.
medium04 has 1 skills, 4 shiftTypes, 0 patterns, 4 contracts, 31 employees, 28 shiftDates, 608 shiftAssignments and 403 requests with a search space of 10^906.
medium05 has 1 skills, 4 shiftTypes, 0 patterns, 4 contracts, 31 employees, 28 shiftDates, 608 shiftAssignments and 403 requests with a search space of 10^906.
medium_hint01 has 1 skills, 4 shiftTypes, 7 patterns, 4 contracts, 30 employees, 28 shiftDates, 428 shiftAssignments and 390 requests with a search space of 10^632.
medium_hint02 has 1 skills, 4 shiftTypes, 7 patterns, 3 contracts, 30 employees, 28 shiftDates, 428

shiftAssignments and 390 requests with a search space of 10^632.
medium_hint03 has 1 skills, 4 shiftTypes, 7 patterns, 4 contracts, 30 employees, 28 shiftDates, 428 shiftAssignments and 390 requests with a search space of 10^632.
medium_late01 has 1 skills, 4 shiftTypes, 7 patterns, 4 contracts, 30 employees, 28 shiftDates, 424 shiftAssignments and 390 requests with a search space of 10^626.
medium_late02 has 1 skills, 4 shiftTypes, 7 patterns, 3 contracts, 30 employees, 28 shiftDates, 428 shiftAssignments and 390 requests with a search space of 10^632.
medium_late03 has 1 skills, 4 shiftTypes, 0 patterns, 4 contracts, 30 employees, 28 shiftDates, 428 shiftAssignments and 390 requests with a search space of 10^632.
medium_late04 has 1 skills, 4 shiftTypes, 7 patterns, 3 contracts, 30 employees, 28 shiftDates, 416 shiftAssignments and 390 requests with a search space of 10^614.
medium_late05 has 2 skills, 5 shiftTypes, 7 patterns, 4 contracts, 30 employees, 28 shiftDates, 452 shiftAssignments and 390 requests with a search space of 10^667.

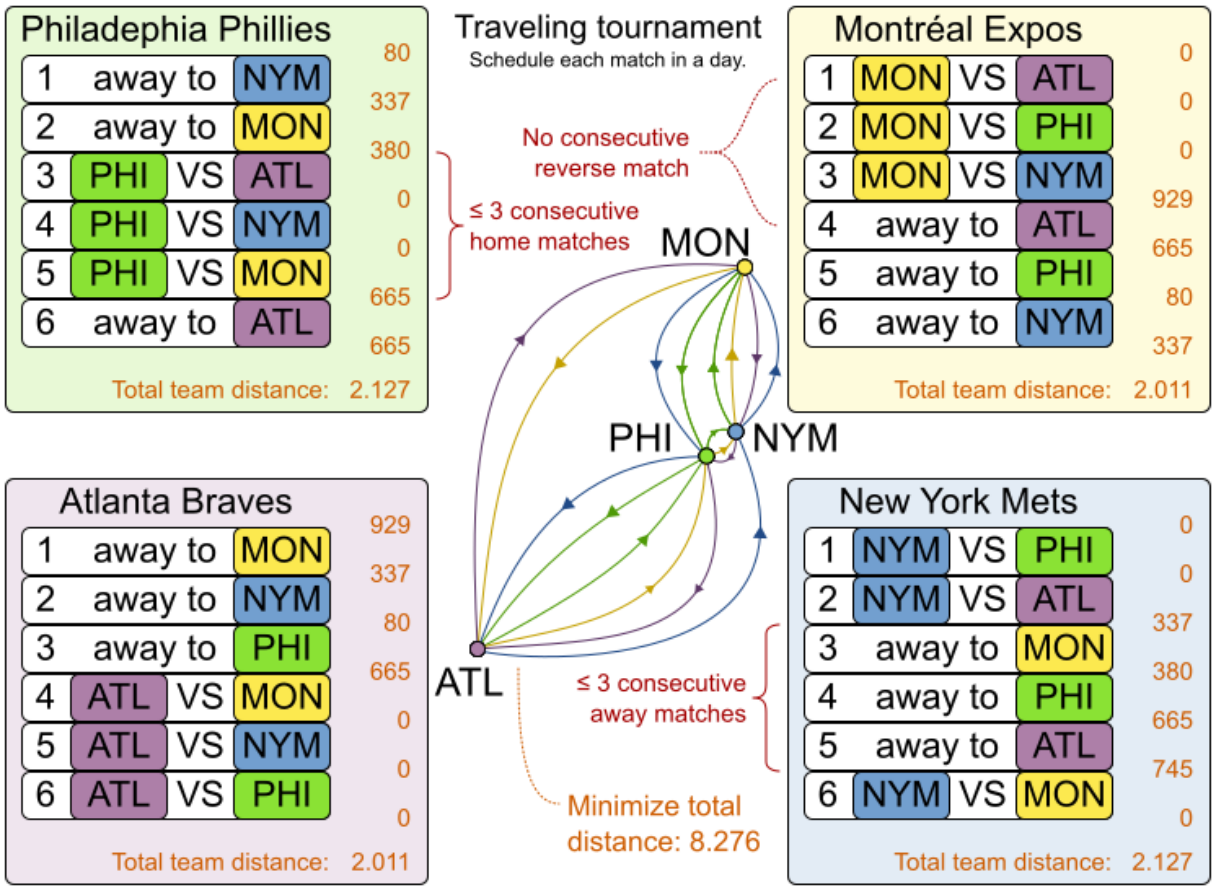
long01 has 2 skills, 5 shiftTypes, 3 patterns, 3 contracts, 49 employees, 28 shiftDates, 740 shiftAssignments and 735 requests with a search space of 10^1250.
long02 has 2 skills, 5 shiftTypes, 3 patterns, 3 contracts, 49 employees, 28 shiftDates, 740 shiftAssignments and 735 requests with a search space of 10^1250.
long03 has 2 skills, 5 shiftTypes, 3 patterns, 3 contracts, 49 employees, 28 shiftDates, 740 shiftAssignments and 735 requests with a search space of 10^1250.
long04 has 2 skills, 5 shiftTypes, 3 patterns, 3 contracts, 49 employees, 28 shiftDates, 740 shiftAssignments and 735 requests with a search space of 10^1250.
long05 has 2 skills, 5 shiftTypes, 3 patterns, 3 contracts, 49 employees, 28 shiftDates, 740 shiftAssignments and 735 requests with a search space of 10^1250.
long_hint01 has 2 skills, 5 shiftTypes, 9 patterns, 3 contracts, 50 employees, 28 shiftDates, 740 shiftAssignments and 0 requests with a search space of 10^1257.
long_hint02 has 2 skills, 5 shiftTypes, 7 patterns, 3 contracts, 50 employees, 28 shiftDates, 740 shiftAssignments and 0 requests with a search space of 10^1257.
long_hint03 has 2 skills, 5 shiftTypes, 7 patterns, 3 contracts, 50 employees, 28 shiftDates, 740 shiftAssignments and 0 requests with a search space of 10^1257.
long_late01 has 2 skills, 5 shiftTypes, 9 patterns, 3 contracts, 50 employees, 28 shiftDates, 752 shiftAssignments and 0 requests with a search space of 10^1277.
long_late02 has 2 skills, 5 shiftTypes, 9 patterns, 4 contracts, 50 employees, 28 shiftDates, 752 shiftAssignments and 0 requests with a search space of 10^1277.
long_late03 has 2 skills, 5 shiftTypes, 9 patterns, 3 contracts, 50 employees, 28 shiftDates, 752 shiftAssignments and 0 requests with a search space of 10^1277.
long_late04 has 2 skills, 5 shiftTypes, 9 patterns, 4 contracts, 50 employees, 28 shiftDates, 752 shiftAssignments and 0 requests with a search space of 10^1277.
long_late05 has 2 skills, 5 shiftTypes, 9 patterns, 3 contracts, 50 employees, 28 shiftDates, 740 shiftAssignments and 0 requests with a search space of 10^1257.

图 31.13. 域模型



31.18. TRAVELING TOURNAMENT 问题(TTP)

计划在 n 个团队数之间的匹配。



硬约束：

- *每个团队会针对其他每个团队进行两次操作：一次，一次。*
- *每个团队在每个时间上都有完全匹配。*
- *团队不能连续三个以上的主页或连续三部的匹配。*
- *无重复者：与团队相比，不连续匹配两个。*

软限制：

- *最大程度降低所有团队的距离。*

在 Michael Trick 网站（其中包含世界记录） 上也定义了问题。

问题大小

```
1-nl04    has  6 days,  4 teams and   12 matches with a search space of    10^5.
1-nl06    has 10 days,  6 teams and   30 matches with a search space of   10^19.
1-nl08    has 14 days,  8 teams and   56 matches with a search space of   10^43.
1-nl10    has 18 days, 10 teams and   90 matches with a search space of   10^79.
1-nl12    has 22 days, 12 teams and  132 matches with a search space of  10^126.
1-nl14    has 26 days, 14 teams and  182 matches with a search space of  10^186.
1-nl16    has 30 days, 16 teams and  240 matches with a search space of  10^259.
2-bra24   has 46 days, 24 teams and  552 matches with a search space of  10^692.
3-nfl16   has 30 days, 16 teams and  240 matches with a search space of  10^259.
3-nfl18   has 34 days, 18 teams and  306 matches with a search space of  10^346.
3-nfl20   has 38 days, 20 teams and  380 matches with a search space of  10^447.
3-nfl22   has 42 days, 22 teams and  462 matches with a search space of  10^562.
3-nfl24   has 46 days, 24 teams and  552 matches with a search space of  10^692.
3-nfl26   has 50 days, 26 teams and  650 matches with a search space of  10^838.
3-nfl28   has 54 days, 28 teams and  756 matches with a search space of  10^999.
3-nfl30   has 58 days, 30 teams and  870 matches with a search space of 10^1175.
3-nfl32   has 62 days, 32 teams and  992 matches with a search space of 10^1367.
4-super04  has  6 days,  4 teams and   12 matches with a search space of    10^5.
4-super06  has 10 days,  6 teams and   30 matches with a search space of   10^19.
4-super08  has 14 days,  8 teams and   56 matches with a search space of   10^43.
4-super10  has 18 days, 10 teams and   90 matches with a search space of   10^79.
4-super12  has 22 days, 12 teams and  132 matches with a search space of  10^126.
4-super14  has 26 days, 14 teams and  182 matches with a search space of  10^186.
5-galaxy04 has  6 days,  4 teams and   12 matches with a search space of    10^5.
5-galaxy06 has 10 days,  6 teams and   30 matches with a search space of   10^19.
5-galaxy08 has 14 days,  8 teams and   56 matches with a search space of   10^43.
5-galaxy10 has 18 days, 10 teams and   90 matches with a search space of   10^79.
5-galaxy12 has 22 days, 12 teams and  132 matches with a search space of  10^126.
5-galaxy14 has 26 days, 14 teams and  182 matches with a search space of  10^186.
5-galaxy16 has 30 days, 16 teams and  240 matches with a search space of  10^259.
5-galaxy18 has 34 days, 18 teams and  306 matches with a search space of  10^346.
5-galaxy20 has 38 days, 20 teams and  380 matches with a search space of  10^447.
5-galaxy22 has 42 days, 22 teams and  462 matches with a search space of  10^562.
5-galaxy24 has 46 days, 24 teams and  552 matches with a search space of  10^692.
5-galaxy26 has 50 days, 26 teams and  650 matches with a search space of  10^838.
5-galaxy28 has 54 days, 28 teams and  756 matches with a search space of  10^999.
5-galaxy30 has 58 days, 30 teams and  870 matches with a search space of 10^1175.
5-galaxy32 has 62 days, 32 teams and  992 matches with a search space of 10^1367.
5-galaxy34 has 66 days, 34 teams and 1122 matches with a search space of 10^1576.
5-galaxy36 has 70 days, 36 teams and 1260 matches with a search space of 10^1801.
5-galaxy38 has 74 days, 38 teams and 1406 matches with a search space of 10^2042.
5-galaxy40 has 78 days, 40 teams and 1560 matches with a search space of 10^2301.
```

31.19. 更低的时间调度

以时间和方式调度所有任务，以最大程度降低电源成本。电源价格因时间而异。这是 作业权利调度的一种形式。

硬约束：

- 开始时间限制：每个任务必须在最早的开始和最新开始限制之间启动。
- 最大容量：不能超过每台机器的每个资源的最大容量。
- 启动和关闭：在分配任务的期间，每台机器都必须处于激活状态。在任务之间，可以闲置它以避免启动和关闭成本。

Medium 约束：

- 电源成本：降低整个计划的总功耗。
  - 机器电源成本：每个活跃或闲置机器都会消耗电源成本，这导致了电源成本（取决于该期间的电源价格）。
  - 任务电源成本：每个任务都消耗了电源成本，这导致了电源成本（取决于一段时间内的电源价格）。
  - 机器启动和关闭成本：计算机启动或关闭的所有时间都会产生额外的成本。

软限制（在原始问题定义中添加）：

- 早开始：请参阅立即启动任务，而不是稍后启动。

这个问题由 ICON 质询 定义。

问题大小

> *sample01 has 3 resources, 2 machines, 288 periods and 25 tasks with a search space of 10^53.*
> *sample02 has 3 resources, 2 machines, 288 periods and 50 tasks with a search space of 10^114.*
> *sample03 has 3 resources, 2 machines, 288 periods and 100 tasks with a search space of 10^226.*
> *sample04 has 3 resources, 5 machines, 288 periods and 100 tasks with a search space of 10^266.*
> *sample05 has 3 resources, 2 machines, 288 periods and 250 tasks with a search space of 10^584.*
> *sample06 has 3 resources, 5 machines, 288 periods and 250 tasks with a search space of 10^673.*
> *sample07 has 3 resources, 2 machines, 288 periods and 1000 tasks with a search space of 10^2388.*
> *sample08 has 3 resources, 5 machines, 288 periods and 1000 tasks with a search space of 10^2748.*
> *sample09 has 4 resources, 20 machines, 288 periods and 2000 tasks with a search space of 10^6668.*
> *instance00 has 1 resources, 10 machines, 288 periods and 200 tasks with a search space of 10^595.*
> *instance01 has 1 resources, 10 machines, 288 periods and 200 tasks with a search space of 10^599.*
> *instance02 has 1 resources, 10 machines, 288 periods and 200 tasks with a search space of 10^599.*
> *instance03 has 1 resources, 10 machines, 288 periods and 200 tasks with a search space of 10^591.*
> *instance04 has 1 resources, 10 machines, 288 periods and 200 tasks with a search space of 10^590.*
> *instance05 has 2 resources, 25 machines, 288 periods and 200 tasks with a search space of 10^667.*
> *instance06 has 2 resources, 25 machines, 288 periods and 200 tasks with a search space of 10^660.*
> *instance07 has 2 resources, 25 machines, 288 periods and 200 tasks with a search space of 10^662.*
> *instance08 has 2 resources, 25 machines, 288 periods and 200 tasks with a search space of 10^651.*
> *instance09 has 2 resources, 25 machines, 288 periods and 200 tasks with a search space of 10^659.*
> *instance10 has 2 resources, 20 machines, 288 periods and 500 tasks with a search space of 10^1657.*
> *instance11 has 2 resources, 20 machines, 288 periods and 500 tasks with a search space of 10^1644.*
> *instance12 has 2 resources, 20 machines, 288 periods and 500 tasks with a search space of 10^1637.*
> *instance13 has 2 resources, 20 machines, 288 periods and 500 tasks with a search space of 10^1659.*
> *instance14 has 2 resources, 20 machines, 288 periods and 500 tasks with a search space of 10^1643.*
> *instance15 has 3 resources, 40 machines, 288 periods and 500 tasks with a search space of 10^1782.*
> *instance16 has 3 resources, 40 machines, 288 periods and 500 tasks with a search space of*

10^1778.
instance17 has 3 resources, 40 machines, 288 periods and 500 tasks with a search space of 10^1764.
instance18 has 3 resources, 40 machines, 288 periods and 500 tasks with a search space of 10^1769.
instance19 has 3 resources, 40 machines, 288 periods and 500 tasks with a search space of 10^1778.
instance20 has 3 resources, 50 machines, 288 periods and 1000 tasks with a search space of 10^3689.
instance21 has 3 resources, 50 machines, 288 periods and 1000 tasks with a search space of 10^3678.
instance22 has 3 resources, 50 machines, 288 periods and 1000 tasks with a search space of 10^3706.
instance23 has 3 resources, 50 machines, 288 periods and 1000 tasks with a search space of 10^3676.
instance24 has 3 resources, 50 machines, 288 periods and 1000 tasks with a search space of 10^3681.
instance25 has 3 resources, 60 machines, 288 periods and 1000 tasks with a search space of 10^3774.
instance26 has 3 resources, 60 machines, 288 periods and 1000 tasks with a search space of 10^3737.
instance27 has 3 resources, 60 machines, 288 periods and 1000 tasks with a search space of 10^3744.
instance28 has 3 resources, 60 machines, 288 periods and 1000 tasks with a search space of 10^3731.
instance29 has 3 resources, 60 machines, 288 periods and 1000 tasks with a search space of 10^3746.
instance30 has 4 resources, 70 machines, 288 periods and 2000 tasks with a search space of 10^7718.
instance31 has 4 resources, 70 machines, 288 periods and 2000 tasks with a search space of 10^7740.
instance32 has 4 resources, 70 machines, 288 periods and 2000 tasks with a search space of 10^7686.
instance33 has 4 resources, 70 machines, 288 periods and 2000 tasks with a search space of 10^7672.
instance34 has 4 resources, 70 machines, 288 periods and 2000 tasks with a search space of 10^7695.
instance35 has 4 resources, 80 machines, 288 periods and 2000 tasks with a search space of 10^7807.
instance36 has 4 resources, 80 machines, 288 periods and 2000 tasks with a search space of 10^7814.
instance37 has 4 resources, 80 machines, 288 periods and 2000 tasks with a search space of 10^7764.
instance38 has 4 resources, 80 machines, 288 periods and 2000 tasks with a search space of 10^7736.
instance39 has 4 resources, 80 machines, 288 periods and 2000 tasks with a search space of 10^7783.
instance40 has 4 resources, 90 machines, 288 periods and 4000 tasks with a search space of 10^15976.
instance41 has 4 resources, 90 machines, 288 periods and 4000 tasks with a search space of 10^15935.
instance42 has 4 resources, 90 machines, 288 periods and 4000 tasks with a search space of 10^15887.
instance43 has 4 resources, 90 machines, 288 periods and 4000 tasks with a search space of 10^15896.
instance44 has 4 resources, 90 machines, 288 periods and 4000 tasks with a search space of

> *10^15885.*
> *instance45 has 4 resources, 100 machines, 288 periods and 5000 tasks with a search space of 10^20173.*
> *instance46 has 4 resources, 100 machines, 288 periods and 5000 tasks with a search space of 10^20132.*
> *instance47 has 4 resources, 100 machines, 288 periods and 5000 tasks with a search space of 10^20126.*
> *instance48 has 4 resources, 100 machines, 288 periods and 5000 tasks with a search space of 10^20110.*
> *instance49 has 4 resources, 100 machines, 288 periods and 5000 tasks with a search space of 10^20078.*

31.20. 投资资产类分配（PORTFOLIO 优化）

决定在每个资产类投资的相对数量。

硬约束：

- *风险最大：标准开发总量不得高于标准开发最大值。*
  - *标准开发总体计算通过应用 Markowitz 组合 Theory 将资产类关联起来考虑起来。*
- ***地区上限：每个地区具有最大数量。***
- ***最大扇区数：每个扇区具有最大数量。***

软限制：

- ***最大化预期的回报。***

问题大小

> *de_smet_1 has 1 regions, 3 sectors and 11 asset classes with a search space of 10^4.*
> *irrinki_1 has 2 regions, 3 sectors and 6 asset classes with a search space of 10^3.*

较大的数据集尚未创建或测试，但不应构成问题。好的数据来源是 这一资产关联网站。

31.21. 会议调度

*将每个会议分配给一个小时和房间。Timeslots 可能会重叠。在 *.xlsx 文件中读取和写入，该文件可使用 libreoffice 或 Excel 编辑。*

硬约束：

- *talk type of timeslot：讨论的类型必须与 timeslot 的对话类型匹配。*
- *房间不可用：讨论的房间必须在对话期间可用。*
- *间冲突：两个对话在重叠时不能使用相同的空间。*
- *发言人不可用时间：所有对话的发言人必须在对话期间可用。*
- *发言人冲突：两个对话在重叠时无法共享发言人。*
- *通用用途和房间标签：*
  - *发言人所需时间标签：如果发言人具有所需的 timelot 标签，则所有其或她的讨论都必须使用该标签分配给一个小时。*
  - *发言人禁止标记：如果发言人具有禁止的时间板板，那么他或她的所有话都不能被分配给具有该标签的时期。*

- *talk required timelot tag：如果一个 talk 具有所需的 timelot 标签，则必须将其分配给带有该标签的 timeslot。*
- *禁止时间标签：如果对话具有禁止的计时标签，则无法使用该标签将其分配给时间。*
- *speaker required room tag：如果发言人具有必需的 room 标签，则必须将其所有或她的讨论分配到具有该标签的房间。*
- *演讲者禁止房间标签：如果发言人有禁止的房间标签，那么他或她的所有客户都不能分配给具有该标签的房间。*
- *talk required room tag： 如果 talk 具有必需的 room 标签，则必须将其分配给具有该标签的房间。*
- *谈话空间标签：如果对话有一个禁止的房间标签，则无法将其分配给具有该标签的房间。*

- *互斥标签：共享这样的标签不能调度在重叠的时间里。*
- *对话：所有预备讨论之后必须调度 talk talk。*

软限制：

- *me track 冲突：减少在重叠时间内共享主题标签的讨论数量。*
- *扇区冲突：减少在重叠期内共享相同的扇区标签的讨论数量。*
- *内容受众级别流违反：对于每内容标签，在高级讨论前，安排简介。*
- *受众级多样性：对于每个时间，请尽量减少与不同受众级别的讨论数量。*

- *语言多样性：对于每一时间来说，最大程度地提高与不同语言的沟通数量。*

- *通用用途和房间标签：*

  - *发言人首选的时间标签：如果发言人具有首选 timelot 标签，则本标签中的所有讨论都应分配给一个小时。*

  - *发言人不需要的时间lot 标签： 如果发言人具有不所需时间标签，那么他或她的任何客户都不能被分配给具有该标签的时间段。*

  - *talk preferred timeslot tag：如果一个 talk 具有首选 timeslot 标签，则应使用该标签将其分配给一个 timeslot。*

  - *talk undesired timeslot tag: 如果 talk 带有不良的 timelot 标签，则不应将其分配给具有该标签的运行时间。*

  - *演讲者首选的房间标签：如果发言人拥有首选房间标签，则本人或她的所有讨论都应分配给具有该标签的房间标签。*

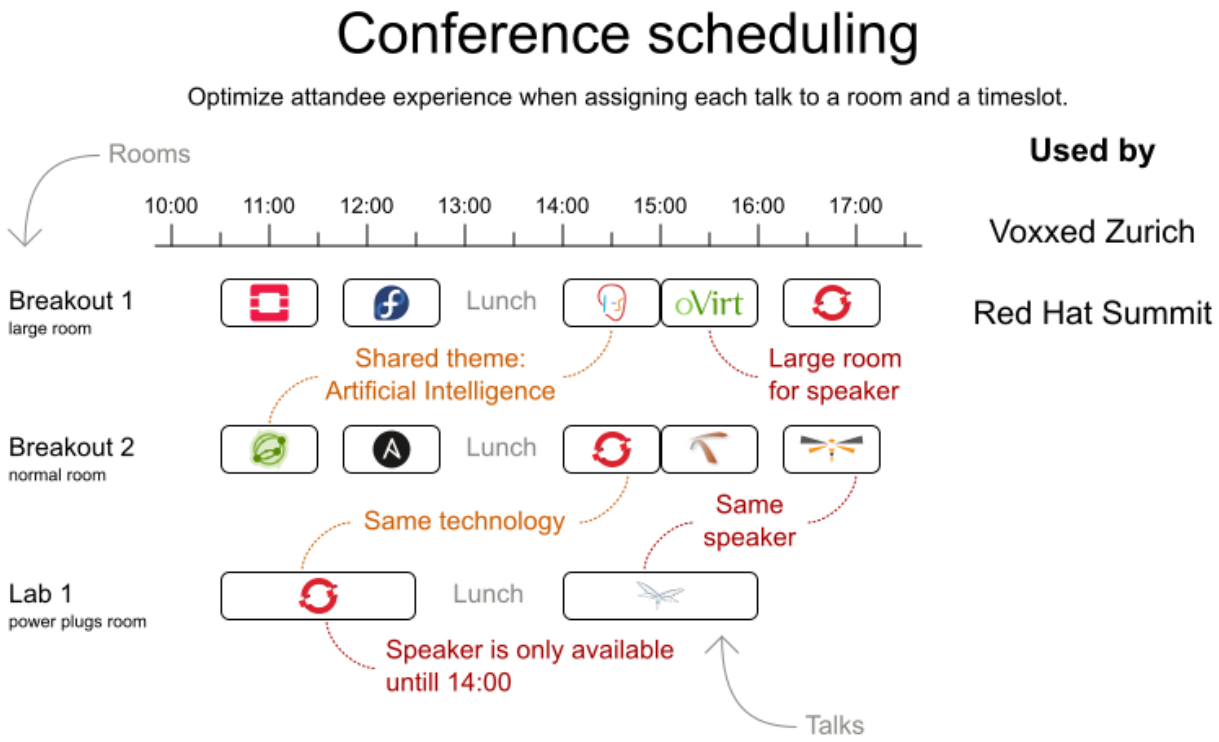
  - *speaker undesired room 标签： 如果发言人具有不需要的房间标签，则不应将任何其讨论分配给具有该标签的房间。*

  - *talk preferred room tag：如果 talk 具有首选空间标签，则应将其分配给具有该标签的房间。*

  - *talk undesired room tag: 如果 talk anirdesired room 标签，则不应将其分配给具有该标签的房间。*

- *第二天讨论：所有共享主题标签或内容标签都应以最少的天数调度（在同一天内）。*

图 31.14. 价值主张



问题大小

> *18talks-6timeslots-5rooms has 18 talks, 6 timeslots and 5 rooms with a search space of 10^26.*
> *36talks-12timeslots-5rooms has 36 talks, 12 timeslots and 5 rooms with a search space of 10^64.*
> *72talks-12timeslots-10rooms has 72 talks, 12 timeslots and 10 rooms with a search space of 10^149.*
> *108talks-18timeslots-10rooms has 108 talks, 18 timeslots and 10 rooms with a search space of 10^243.*
> *216talks-18timeslots-20rooms has 216 talks, 18 timeslots and 20 rooms with a search space of 10^552.*

31.22. STTOUR

从展到演示到展示的车车，但计划仅显示可用日。

硬约束：

- *安排每个必需显示。*
- *计划尽可能多的显示。*

Medium 约束：

- *最大化收入机会。*
- *最小化驱动时间。*
- *快于稍后访问。*

软限制：

- *避免长时间推动时间。*

问题大小

> *47shows has 47 shows with a search space of 10^59.*

31.23. FLIGHT CREW 调度

为试点和航班出员指派航班。

硬约束：

- 所需技能：每个班级分配都具有必要的技能。例如，flight AB0001 需要 2 个试点和 3 班级参加。

- 动态冲突：每个员工只能同时参加一个动态

- 在两个航班之间传输：在两个机之间，员工必须能够从 arrival airport 转让到 departure airport。例如，An 到达 Brussels at 10:00，在位于 15:00 的阿姆斯特区。

- 员工不可用：该员工必须在航班的某一天可用。例如，An 是 1-Feb 上的 PTO。

软限制：

- 首次分配来自家

- 最后分配到达家

- 每个员工的负载均衡持续时间

问题大小

> 175flights-7days-Europe has 2 skills, 50 airports, 150 employees, 175 flights and 875 flight assignments with a search space of 10^1904.
> 700flights-28days-Europe has 2 skills, 50 airports, 150 employees, 700 flights and 3500 flight assignments with a search space of 10^7616.
> 875flights-7days-Europe has 2 skills, 50 airports, 750 employees, 875 flights and 4375 flight assignments with a search space of 10^12578.
> 175flights-7days-US has 2 skills, 48 airports, 150 employees, 175 flights and 875 flight assignments with a search space of 10^1904.

第 32 章 下载红帽构建的 OPTAPLANNER 示例

您可以下载 Red Hat build of OptaPlanner 示例，作为红帽客户门户网站中提供的 {PRODUCTPAM} 附加组件软件包的一部分。

流程

1. 进入红帽客户门户网站中的 Software Downloads 页面（需要登录），然后从下拉列表中选择产品和版本：
   - 产品： 流程自动化管理器
   - Version: 7.13.5
2. 下载 Red Hat Process Automation Manager 7.13 Add Ons。
3. 提取 rhpam-7.13.5-add-ons.zip 文件。提取的 add-ons 文件夹包含 rhpam-7.13.5-planner-engine.zip 文件。
4. 提取 rhpam-7.13.5-planner-engine.zip 文件。

结果

提取的 rhpam-7.13.5-planner-engine 目录包含以下子目录下的示例源代码：

- 示例/源/src/main/java/org/optaplanner/examples
- 示例/源/src/main/resources/org/optaplanner/examples

32.1. 运行 OPTAPLANNER 示例

红帽构建的 OptaPlanner 包含多个示例，演示各种计划用例。下载并使用示例来探索不同类型的规划解决方案。

先决条件

- *您已下载并提取示例，如 第 32 章 下载红帽构建的 OptaPlanner 示例 所述。*

流程

1. *要运行示例，请在 rhpam-7.13.5-planner-engine/examples 目录中输入以下命令之一：*

   Linux 或 Mac:

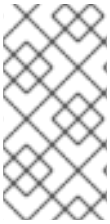
   ```
   $ ./runExamples.sh
   ```

   Windows：

   ```
   $ runExamples.bat
   ```

   OptaPlanner Examples 窗口将打开。

2. *选择一个示例来运行该示例。*



注意

Red Hat build of OptaPlanner 没有 GUI 依赖项。它还可在服务器或移动 JVM 上运行，就像桌面一样。

32.2. 在 IDE 中运行 OPTAPLANNER 示例构建（INTELLIJ、ECLIPSE 或 NETBEANS）

如果您使用集成开发环境(IDE)，如 IntelliJ、Eclipse 或 Netbeans，您可以在开发环境中运行下载的 OptaPlanner 示例。

先决条件

- *您已下载并提取了 OptaPlanner 示例，如 第 32 章 下载红帽构建的 OptaPlanner 示例 所述。*

流程

1. 打开 OptaPlanner 示例作为新项目：

   a. 对于 IntelliJ 或 Netbeans，开放 示例/源/pom.xml 作为新项目。Maven 集成指南，指导您完成剩余的安装。跳过这个过程中的其余步骤。

   b. 对于 Eclipse，为 /examples/binaries 目录打开一个新项目，它位于 rhpam-7.13.5-planner-engine 目录下。

2. 将所有位于 二进制文件 目录中的所有 JAR 文件添加到 classpath 中，但 examples/binaries/optaplanner-examples-7.67.0.Final-redhat-00024.jar 文件除外。

3. 添加 Java 源目录 src/main/java 和 Java 资源目录 src/main/resources，它位于 rhpam-7.13.5-planner-engine/examples/sources/ 目录下。

4. 创建运行配置：

   - main class: org.optaplanner.examples.app.OptaPlannerExamplesApp
   - VM 参数（可选）： -Xmx512M -server -Dorg.optaplanner.examples.dataDir=examples/sources/data
   - 工作目录： example/sources

5. 运行 run 配置。

第 33 章 BUSINESS CENTRAL 中的 OPTAPLANNER 入门：员工名单示例

您可以在 Business Central 中构建和部署员工 模板示例项目。该项目演示了如何创建解决传统规划问题所需的每个业务中心资产，并使用红帽构建的 OptaPlanner 来查找可能的解决方案。

您可以在 Business Central 中部署预配置的 员工 -rostering 项目。或者，您可以使用 Business Central 自行创建项目。

注意

Business Central 中的 employees -rostering 示例项目不包括数据集。您必须使用 REST API 调用以 XML 格式提供数据集。

33.1. 在 BUSINESS CENTRAL 中部署 EMPLOYEES ROSTERING 示例项目

Business Central 包括很多可用于熟悉产品和其功能的示例项目。员工的 rostering 示例项目经过设计和创建，以展示红帽构建 OptaPlanner 的变速用例。使用以下步骤部署和运行 Business Central 中的员工降级示例。

先决条件

- Red Hat Process Automation Manager 下载并安装。有关安装选项，请参阅 规划 Red Hat Process Automation Manager 安装。
- 您已启动了 Red Hat Process Automation Manager，如安装文档，且您使用具有 admin 权限的用户身份登录 Business Central。

流程

1. 在 Business Central 中，点击 Menu → Design → Projects。
2. 在预配置的 MySpace 空间中，点 Try Samples。
3. 从示例项目列表中选择 employees -rostering，然后单击右上角的 Ok 来导入项目。

4. 资产列表变得复杂后，请单击 **Build & Deploy** 来部署员工的 rostering 示例。

本文档的其余部分介绍了各个项目资产及其配置。

33.2. 重新排序员工降级示例项目

员工的 rostering 示例项目是 Business Central 中提供的一个预配置的项目。您可以在 第 33.1 节 “在 Business Central 中部署 employees rostering 示例项目” 中了解如何部署此项目。

您可以创建员工的名册示例"全新"。您可以使用本例中的工作流在 Business Central 中创建您自己的类似项目。

33.2.1. 设置员工的 rostering 项目

要在 Business Central 中开始开发解决方法，您必须设置该项目。

先决条件

- Red Hat Process Automation Manager 下载并安装。
- 您已部署了 Business Central，并使用具有管理员角色的用户登录。

流程

1. 在 Business Central 中点 **Menu** → **Design** → **Projects** → **Add Project** 创建一个新项目。
2. 在 **Add Project** 窗口中，填写以下字段：
   - **Name** : staff-rostering
   - **Description**（可选）：使用 OptaPlanner 的 Employee rostering 问题优化。可根据自身技能分配员工进行转变。

可选：点 Configure Advanced Options 填充 组 ID、Artifact ID 和版本信息。

- 组 ID ：员工
- 工件 ID ：员工
- 版本 1：1NAPSHOT

3. 单击 Add，将项目添加到 Business Central 项目存储库。

33.2.2. 问题事实和规划实体

员工时间表问题中的每个域类都归类为以下一种：

- 不相关的类：未由任何分数限制使用。从规划角度来说，这个数据已过时。
- 问题事实 类：在分数约束下使用，但在规划期间不会改变（只要问题仍保持相同），例如 Shift 和 Employee。问题事实类的所有属性都是问题属性。
- 计划实体类：在规划 过程中对分数约束和更改使用，例如 ShiftAssignment。规划期间更改的属性是 规划变量。其他属性是问题属性。

询问以下问题：

- 计划过程中发生了哪些类变化？
- 哪个类具有我想要 更改 的变量？

该类是规划实体。

计划实体类需要使用 @PlanningEntity 注释注释，或使用红帽在域设计器中的

OptaPlanner dock 进行定义。

每个规划实体类都有一个或多个 规划变量，并且还必须有一个或多个定义属性。

大多数用例只有一个规划实体类，每个计划实体类只有一个规划变量。

33.2.3. 为员工降级项目创建数据模型

使用这个部分，创建在 Business Central 中运行员工指定示例项目所需的数据对象。

先决条件

- 您已完成 第 33.2.1 节 “设置员工的 rostering 项目” 中描述的项目设置。

流程

1. 使用新项目时，点击项目视角中的 Data Object，或者点击 Add Asset → Data Object 以创建新的数据对象。
2. 将第一个数据对象 时间 命名为，然后选择 employees rostering.employeerostering 作为软件包。

   点 确定。
3. 在 Data Objects 视角中，点 +add 字段将字段 添加到 Timeslot data 对象。
4. 在 id 字段中，键入 endTime。
5. 单击 Type 旁边的下拉菜单，然后选择 LocalDateTime。
6. 点击 Create 并继续 添加另一个字段。

7. *添加另一个字段，其中包含 id startTime 和 Type LocalDateTime。*

8. *点 Create。*

9. *单击右上角的 Save，以保存 Timeslot 数据对象。*

10. *单击右上角的 x 以关闭 Data Objects 透视图，再返回到 Assets 菜单。*

11. *使用前面的步骤创建以下数据对象及其属性：*

表 33.1. 技巧

id	类型
name	**字符串**

表 33.2. 员工

id	类型
name	**字符串**
技能	**employeerostering.employeerostering.Skill[List]**

表 33.3. 改变

id	类型
requiredSkill	**employeerostering.employeerostering.Skill**
timeslot	**employeerostering.employeerostering.Timeslot**

表 33.4. DayOffRequest

id	类型
date	**LocalDate**

id	类型
员工	employeerostering.employeerostering.Employee

表 33.5. ShiftAssignment

id	类型
员工	employeerostering.employeerostering.Employee
改变	employeerostering.employeerostering.Shift

有关创建数据对象的更多示例，请参阅开始使用决策服务。

33.2.3.1. 创建员工的 roster 计划实体

为了解决员工问答规划问题，您必须创建一个计划实体和解决方法。计划实体在域设计人员中使用 OptaPlanner dock 中可用的属性定义。

使用以下步骤将 ShiftAssignment 数据对象定义为员工名册的规划实体。

先决条件

- *您已通过完成 第 33.2.3 节 “为员工降级项目创建数据模型” 中的步骤创建运行员工代理示例所需的相关数据对象和规划实体。*

流程

1. *在项目 Assets 菜单中，打开 ShiftAssignment data 对象。*
2. *在 Data Objects 视角中，点右侧的*

   来打开 OptaPlanner dock。

3. 选择 计划实体。

4. 从 ShiftAssignment data 对象下的字段列表中选择 employees。

5. 在 OptaPlanner dock 中，选择 规划变量。

   在 Value Range Id 输入字段中，键入 employeesRange。这会在计划实体中添加 @ValueRangeProvider 注释，您可以通过单击设计者中的 Source 选项卡来查看。

   planning 变量的值范围通过 @ValueRangeProvider 注释定义。@ValueRangeProvider 注释始终具有属性 id，它被 @PlanningVariable 属性值RangeProviderRefs 引用。

6. 关闭 dock 并单击 Save 以保存数据对象。

33.2.3.2. 创建员工的 roster 计划解决方案

员工的 roster 问题依赖于定义的规划解决方案。计划解决方案由红帽构建 OptaPlanner dock 中的属性在域设计器中定义。

先决条件

- 您已通过完成 第 33.2.3 节 “为员工降级项目创建数据模型” 和 第 33.2.3.1 节 “创建员工的 roster 计划实体” 中的步骤创建运行员工指定示例所需的相关数据对象和规划实体。

流程

1. 使用标识符 EmployeeRoster 创建一个新的数据对象。

2. 创建以下字段：

表 33.6. EmployeeRoster

id	类型
dayOffRequestList	employeerostering.employeerostering.DayOffRequest[List]

id	类型
shiftAssignmentList	**employeerostering.employeerostering.ShiftAssignment[List]**
shiftList	**employeerostering.employeerostering.Shift[List]**
skillList	**employeerostering.employeerostering.Skill[List]**
timeslotList	**employeerostering.employeerostering.Timeslot[List]**

3. *在 Data Objects 视角中，点右侧的*

来打开 OptaPlanner dock。

4. *选择 规划解决方案.*

5. *将默认的 hard soft 分数 保留为 Solution Score Type。这会在 EmployeeRoster 数据对象中自动生成 分数，该分数作为类型。*

6. *使用以下属性添加新字段：*

id	类型
employeeList	**employeerostering.employeerostering.Employee[List]**

7. *选择 employees List 字段后，打开 OptaPlanner dock，再选择 Planning Value Range Provider 框。*

在 id 字段中，键入 staff Range。关闭 dock。

8. *点击右上角的 Save 保存资产。*

33.2.4. 员工的降级限制

员工名单是一个规划问题。所有规划问题都包括必须满足约束才能找到最佳解决方案。

Business Central 中的员工指定示例项目包括以下硬和软限制：

硬约束

- 员工仅针对每天进行一次转变。
- 所有需要特定员工技能的改变都会为拥有该特定技能的员工分配。

软限制

- 所有员工都被分配了一项转变。
- 如果雇员请求一天，其变化会被重新分配给其他员工。

硬和软限制在 Business Central 中通过自由-form DRL 设计器定义，也可以使用指导规则。

33.2.4.1. DRL（Drools 规则语言）规则

DRL(Drools Rule)规则是您在 .drl 文本文件中直接定义的业务规则。这些 DRL 文件是 Business Central 中所有其他规则资产渲染的源。您可以在 Business Central 界面中创建和管理 DRL 文件，或使用 Red Hat CodeReady Studio 或其他集成开发环境(IDE)在外部创建它们。DRL 文件可以包含一个或多个规则，它们至少定义规则条件（在时）和操作（然后再）。Business Central 中的 DRL 设计器为 Java、DRL 和 XML 提供语法高亮显示。

DRL 文件由以下组件组成：

DRL 文件中的组件

> *package*

```
import

function  // Optional

query  // Optional

declare   // Optional

global   // Optional

rule "rule name"
    // Attributes
    when
        // Conditions
    then
        // Actions
end

rule "rule2 name"

...
```

以下示例 DRL 规则决定了 loan 应用程序决策服务中的年龄限制：

loan Application age 限制的规则示例

```
rule "Underage"
  salience 15
  agenda-group "applicationGroup"
  when
    $application : LoanApplication()
    Applicant( age < 21 )
  then
    $application.setApproved( false );
    $application.setExplanation( "Underage" );
end
```

DRL 文件包含单个或多个规则、查询和函数，并可以定义由规则和查询分配和使用的属性等资源声明。DRL 软件包必须在 DRL 文件的顶部列出，规则通常最后列出。所有其他 DRL 组件可遵循任何顺序。

每个规则必须在规则软件包中具有唯一的名称。如果您在软件包中的任何 DRL 文件中使用相同的规则名称多次，则规则无法编译。始终使用双引号括起规则名称（规则"rule name"），以防止可能出现的编译错误，特别是在规则名称中使用空格。

与 DRL 规则相关的所有数据对象都必须位于与 Business Central 中的 DRL 文件相同的项目软件包中。默认导入同一软件包中的资产。其他软件包中的现有资产可以通过 DRL 规则导入。

33.2.4.2. 使用 DRL designer 定义员工指定限制

您可以使用 Business Central 中的自由-form DRL 设计器为员工名册创建约束定义。

使用此流程创建一个 硬约束，使其没有员工分配在之前的 10 小时后开始的转换。

流程

1. 在 Business Central 中，前往 Menu → Design → Projects 并点项目名称。
2. 点 Add Asset → DRL 文件。
3. 在 DRL 文件名 字段中，键入 complexScoreRules。
4. 选择 employees rostering.employeerostering 软件包。
5. 点 +Ok 创建 DRL 文件。
6. 在 DRL 设计器的 Model 选项卡中，将 Employee10HourShiftSpace 规则定义为 DRL 文件：

```
package employeerostering.employeerostering;

rule "Employee10HourShiftSpace"
    when
        $shiftAssignment : ShiftAssignment( $employee : employee != null, $shiftEndDateTime :
shift.timeslot.endTime)
        ShiftAssignment( this != $shiftAssignment, $employee == employee, $shiftEndDateTime
<= shift.timeslot.endTime,
```

```
            $shiftEndDateTime.until(shift.timeslot.startTime,
java.time.temporal.ChronoUnit.HOURS) <10)
    then
        scoreHolder.addHardConstraintMatch(kcontext, -1);
end
```

7. 点 Save 保存 DRL 文件。

有关创建 DRL 文件的更多信息，请参阅使用 DRL 规则设计决策服务。

33.2.5. 使用指导规则为员工名单创建规则

您可以使用 Business Central 中的指导规则设计人员创建为员工名单定义硬和软约束的规则。

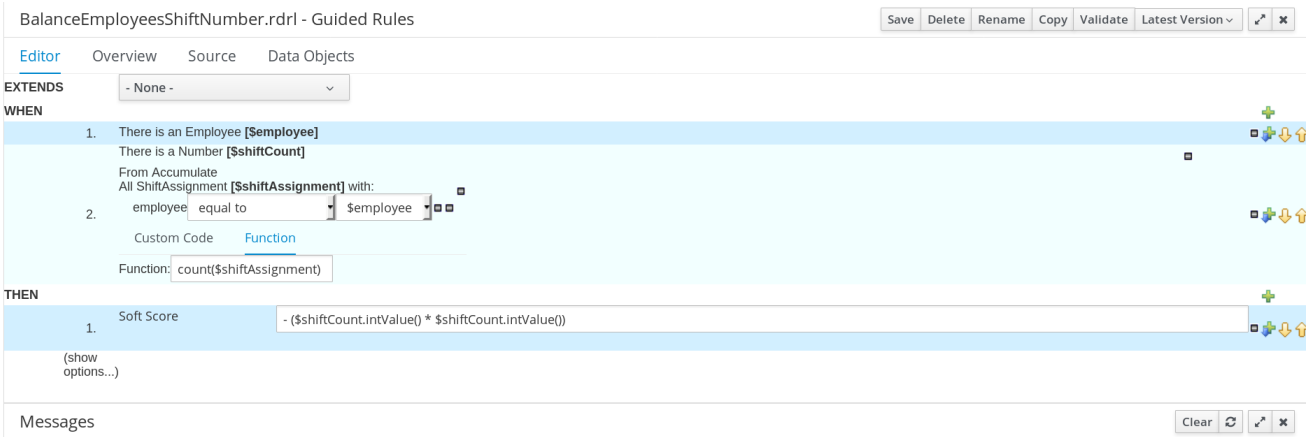
33.2.5.1. 指导规则

指导规则是您通过规则创建在业务中心基于 UI 的指导规则设计者中创建的业务规则。指导规则设计器根据所定义规则的数据对象，提供可接受输入的字段和选项。您定义的指南规则与所有其他规则资产一样编译为 Drools Rule Language(DRL)规则。

与指导规则相关的所有数据对象都必须位于与指导规则相同的项目软件包中。默认导入同一软件包中的资产。创建必要的数据对象和指导规则后，您可以使用指南规则设计器的 Data Objects 选项卡来验证所有所需数据对象是否已列出或导入其他现有数据对象，方法是添加新项目。

33.2.5.2. 创建用于平衡员工切换数量的指导规则

BalanceEmployeesShiftNumber 指导规则创建一个软约束，确保以尽可能均匀地平衡的方式为员工分配切换。它通过创建一个分数影响，在变送变小时增加。分数公式由规则实施，使 Solver 以更均衡的方式分发转换。



流程

1. 在 Business Central 中，前往 Menu → Design → Projects 并点项目名称。

2. 点 Add Asset → Guided Rule。

3. 输入 BalanceEmployeesShiftNumber 作为 引导规则 名称，再选择 employees rostering.employeerostering 软件包。

4. 点 Ok 创建规则资产。

5. 点 WHEN 字段中的

来添加 WHEN 条件。

6. 在规则窗口的 Add a 条件 中选择 Employee。单击 +Ok。

7. 单击 Employee 条件来修改限制，并添加变量名称 $employee。

8. 添加来自 加速 的 WHEN 条件。

    a. 在 From Accumulate 条件 上方点击 添加模式 并选择 Number 作为事实类型，从下拉列表中选择 Number。

    b. 将变量名称 $shiftCount 添加到 Number 条件中。

    c. 在 From Accumulate 条件下，单击添加模式，然后从下拉列表中选择 ShiftAssignment fact 类型。

    d. 将变量名称 $shiftAssignment 添加到 ShiftAssignment fact 类型。

    e. 再次单击 ShiftAssignment 条件，然后从 Add a restrictions on a field 下拉列表中

选择 **employees**。

f. 从 **employees** 约束旁边的下拉列表中选择 **等于**。

g. 点击下拉按钮旁的  图标添加变量，然后点击 **Field value** 窗口中的 **Bound** 变量。

h. 从下拉列表中选择 **$employee**。

i. 在 **Function** 框中，键入 **count($shiftAssignment)**。

9. 点 **wordpress N** 字段中的  来添加 **wordpressN** 条件。

10. 在 **Add a new action** 窗口中选择 **Modify Soft Score**。单击 **+Ok**。

a. 在方框中输入以下表达式：
-（$shiftCount.intValue（）*$shiftCount.intValue（））

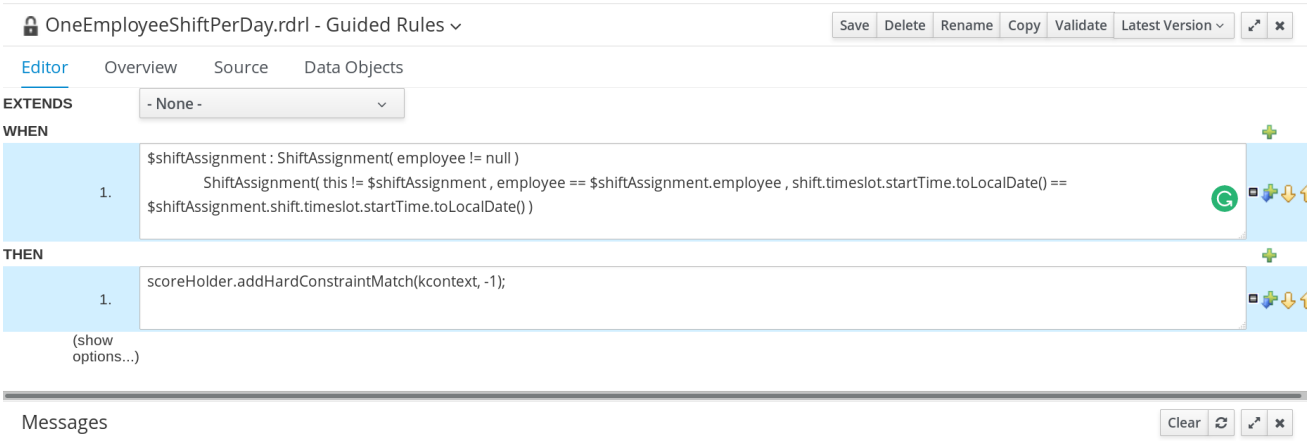
11. 单击右上角的 **Validate** 来检查所有规则条件是否有效。如果规则验证失败，解决错误消息中描述的任何问题，查看规则中的所有组件，然后重试验证规则直到规则通过为止。

12. 单击 **Save** 以保存该规则。

有关创建指南规则的更多信息，请参阅使用指导规则 设计决策服务。

33.2.5.3. 为每天不一换个变化创建指导规则

OneEmployeeShiftPerDay 指导规则创建了一个硬约束，员工不会超过一天的转变。在员工指定示例中，这个约束是使用指导的规则设计人员创建的。



流程

1. ***在 Business Central 中，前往 Menu → Design → Projects 并点项目名称。***

2. ***点 Add Asset → Guided Rule。***

3. ***输入 OneEmployeeShiftPerDay 作为 引导规则 名称，再选择 employees rostering.employeerostering 软件包。***

4. ***点 Ok 创建规则资产。***

5. ***点 WHEN 字段中的***

来添加 WHEN 条件。

6. ***从 Add a condition to rule 窗口中选择 Free form DRL。***

7. ***在自由表单 DRL 框中，输入以下条件：***

```
$shiftAssignment : ShiftAssignment( employee != null )
  ShiftAssignment( this != $shiftAssignment , employee == $shiftAssignment.employee
, shift.timeslot.startTime.toLocalDate() ==
$shiftAssignment.shift.timeslot.startTime.toLocalDate() )
```

此条件指出，不能分配给已经在同一天进行另一个切换分配的员工。

8. 点 wordpress N 字段中的

来添加 wordpressN 条件。

9. 从 Add a new action 窗口中选择 Add free form DRL。

10. 在自由表单 DRL 框中，输入以下条件：

```
scoreHolder.addHardConstraintMatch(kcontext, -1);
```

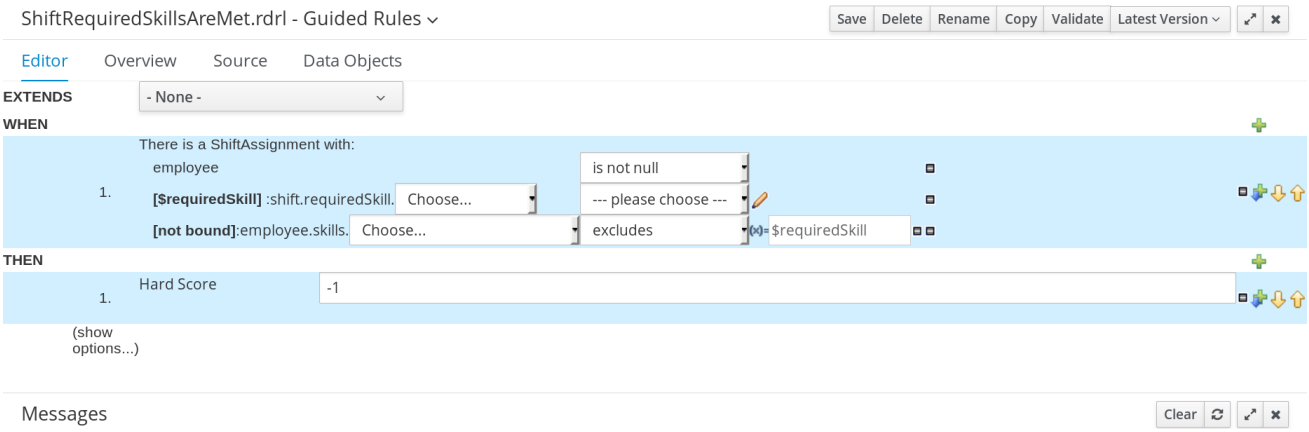
11. 单击右上角的 Validate 来检查所有规则条件是否有效。如果规则验证失败，解决错误消息中描述的任何问题，查看规则中的所有组件，然后重试验证规则直到规则通过为止。

12. 单击 Save 以保存该规则。

有关创建指南规则的更多信息，请参阅使用指导规则 设计决策服务。

33.2.5.4. 创建一条指导规则以满足不断变化的要求技能

ShiftReqiredSkillsAreMet guided 规则创建了一个硬约束，以确保所有转换都被分配了一组正确技能的员工。在员工指定示例中，这个约束是使用指导的规则设计人员创建的。



流程

1. 在 Business Central 中，前往 Menu → Design → Projects 并点项目名称。

2. 点 Add Asset → Guided Rule。

3. 输入 ShiftReqiredSkillsAreMet 作为 引导规则 名称并选择 employees rostering.employeerostering 软件包。

4. 点 Ok 创建规则资产。

5. 点 WHEN 字段中的

来添加 WHEN 条件。

6. 选择 Add a condition to rule 窗口中的 ShiftAssignment。单击 +Ok。

7. 单击 ShiftAssignment 条件，然后从 Add a limits on a field 下拉列表中选择 employees。

8. 在设计人员中，单击 员工 旁边的下拉列表，然后选择 不是 null。

9. 单击 ShiftAssignment 条件，然后单击 Expression 编辑器。

   a. 在设计人员中，单击 [not bound] 以打开 Expression 编辑器，并将表达式绑定到变量 $requiredSkill。点 Set。

   b. 在设计人员（ /requiredSkill 旁边），从第一个下拉列表中选择 转换，然后从下一下拉列表中选择 requiredSkill。

10. 单击 ShiftAssignment 条件，然后单击 Expression 编辑器。

    a. 在设计人员（ [not bound] 旁边），从第一个下拉列表中选择 员工，然后从下一下拉列表中选择 技能。

'

b. 将下一个下拉列表保留为 选择。

c. 在下一个下拉菜单中，请更改 以排除 。

d. 点击 排除 的

图标，在 Field value 窗口中点击 New formula 按钮。

e. 在公式框中输入 $requiredSkill。

11. 点 wordpress N 字段中的

来添加 wordpressN 条件。

12. 在 Add a new action 窗口中选择 Modify Hard Score。单击 +Ok。

13. 在分数操作框中输入 -1。

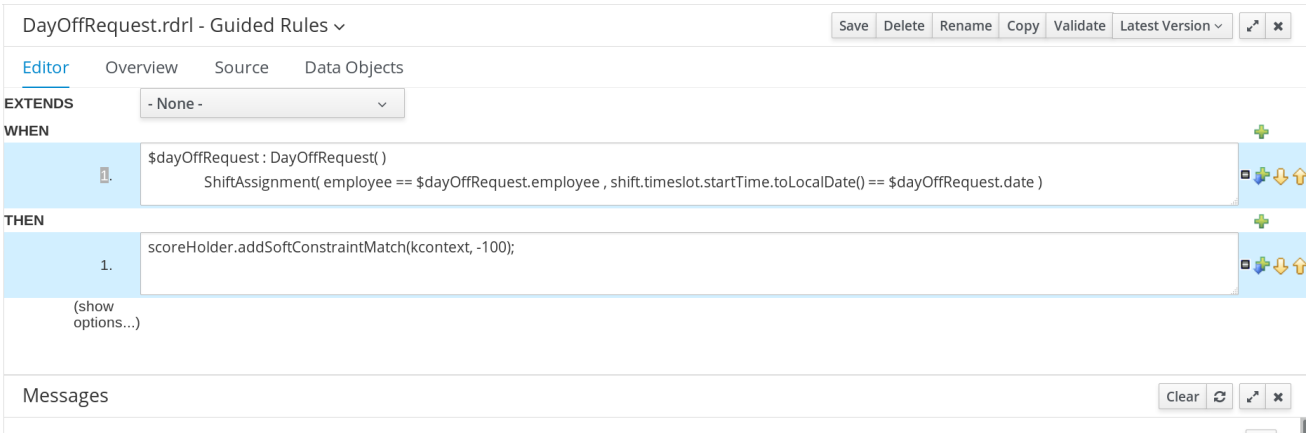
14. 单击右上角的 Validate 来检查所有规则条件是否有效。如果规则验证失败，解决错误消息中描述的任何问题，查看规则中的所有组件，然后重试验证规则直到规则通过为止。

15. 单击 Save 以保存该规则。

有关创建指南规则的更多信息，请参阅使用指导规则 设计决策服务。

33.2.5.5. 创建用于管理开箱即用请求的指南规则

DayOffRequest guided 规则会创建一个软约束。如果最初被分配了该转换的人，则这个限制可以重新分配给另一位员工。在员工指定示例中，这个约束是使用指导的规则设计人员创建的。



流程

1. ***在 Business Central 中，前往 Menu → Design → Projects 并点项目名称。***

2. ***点 Add Asset → Guided Rule。***

3. ***输入 DayOffRequest 作为 引导规则 名称，再选择 employees rostering.employeerostering 软件包。***

4. ***点 Ok 创建规则资产。***

5. ***点 WHEN 字段中的***

来添加 WHEN 条件。

6. ***从 Add a condition to rule 窗口中选择 Free form DRL。***

7. ***在自由表单 DRL 框中，输入以下条件：***

```
$dayOffRequest : DayOffRequest( )
  ShiftAssignment( employee == $dayOffRequest.employee ,
shift.timeslot.startTime.toLocalDate() == $dayOffRequest.date )
```

如果转移被分配给一个进行当天发出的某个员工，则该条件将指出，该条件可以取消对当天的转换进行取消签名。

8.

点 wordpress N 字段中的

来添加 wordpressN 条件。

9. 从 Add a new action 窗口中选择 Add free form DRL。

10. 在自由表单 DRL 框中，输入以下条件：

```
scoreHolder.addSoftConstraintMatch(kcontext, -100);
```

11. 单击右上角的 Validate 来检查所有规则条件是否有效。如果规则验证失败，解决错误消息中描述的任何问题，查看规则中的所有组件，然后重试验证规则直到规则通过为止。

12. 单击 Save 以保存该规则。

有关创建指南规则的更多信息，请参阅使用指导规则 设计决策服务。

33.2.6. 为员工降级创建解决器配置

您可以在 Business Central 中创建并编辑 Solver 配置。Solver 配置设计器会创建一个解决器配置，可在项目部署后运行。

先决条件

- Red Hat Process Automation Manager 下载并安装。
- 您已为员工名录示例创建并配置了所有相关资产。

流程

1. 在 Business Central 中，点击 Menu → Projects，然后点击您的项目打开它。
2. 在 Assets 视角中，点 Add Asset → Solver 配置

3. 在 Create new Solver 配置 窗口中，为您的 Solver 键入名称 EmployeeRosteringSolverConfig，然后点击 Ok。

   这会打开 Solv er 配置 设计器。

4. 在 Score Director Factory 配置部分中，定义包含评分规则定义的 KIE 基础。员工的 rostering 示例项目使用 defaultKieBase。

   a. 选择在 KIE 基本中定义的一个 KIE 会话。员工的 rostering 示例项目使用 defaultKieSession。

5. 单击右上角的 Validate，以检查 Score Director Factory 配置是否正确。如果验证失败，解决错误消息中描述的任何问题，然后重试进行验证，直到配置通过。

6. 点 Save 保存 Solver 配置。

33.2.7. 为员工降级项目配置 Solver 终止

您可以将 Solver 配置为在指定时间后终止。默认情况下，计划引擎会提供无限的时间段来解决问题实例。

员工的 rostering 示例项目设置为运行 30 秒。

先决条件

- 您已为员工名册项目创建了所有相关资产，并在 Business Central 中创建了 EmployeeRosteringSolverConfig solver 配置，如 第 33.2.6 节 “为员工降级创建解决器配置” 所述。

流程

1. 从 Assets 视角打开 EmployeeRosteringSolverConfig。这将打开 Solv er 配置 设计器。

2. 在 Termination 部分，点 Add 在所选逻辑组中创建新终止元素。

3. 从下拉列表中选择所花费的 终止类型。这会在终止配置中添加为输入字段。

4. 使用时间元素旁边的箭头来调整 30 秒的时间。

5. 单击右上角的 Validate，以检查 Score Director Factory 配置是否正确。如果验证失败，解决错误消息中描述的任何问题，然后重试进行验证，直到配置通过。

6. 点 Save 保存 Solver 配置。

33.3. 使用 REST API 访问解决问题

在部署或重新创建示例地址后，您可以使用 REST API 访问它。

您必须使用 REST API 注册解决器实例。然后，您可以提供数据集并检索优化的解决方案。

先决条件

- 根据本文档中前面的章节设置和部署员工的 rostering 项目。您可以部署示例项目，如 第 33.1 节 “在 Business Central 中部署 employees rostering 示例项目” 所述，或者重新创建项目，如 第 33.2 节 “重新排序员工降级示例项目” 所述。

33.3.1. 使用 REST API 注册 Solver

您必须使用 REST API 注册 solver 实例，然后才能使用 solver。

每个临时解决方案都能够一次优化一个规划问题。

流程

1. 使用以下标头创建 HTTP 请求：

```
authorization: admin:admin
X-KIE-ContentType: xstream
content-type: application/xml
```

2. 使用以下请求注册 Solver：

PUT

http://localhost:8080/kie-server/services/rest/server/containers/employeerostering_1.0.0-SNAPSHOT/solvers/EmployeeRosteringSolver

请求正文

```
<solver-instance>
  <solver-config-file>employeerostering/employeerostering/EmployeeRosteringSolverConfig.solver.xml</solver-config-file>
</solver-instance>
```

33.3.2. 使用 REST API 调用 Solver

在注册了解决器实例后，您可以使用 REST API 将数据设置为解决方法，并检索优化的解决方案。

流程

1. 使用以下标头创建 HTTP 请求：

```
authorization: admin:admin
X-KIE-ContentType: xstream
content-type: application/xml
```

2. 使用数据收集向 Solver 提交请求，如下例所示：

POST

http://localhost:8080/kie-server/services/rest/server/containers/employeerostering_1.0.0-SNAPSHOT/solvers/EmployeeRosteringSolver/state/solving

请求正文

```
<employeerostering.employeerostering.EmployeeRoster>
  <employeeList>
    <employeerostering.employeerostering.Employee>
      <name>John</name>
      <skills>
        <employeerostering.employeerostering.Skill>
          <name>reading</name>
```

```
      </employeerostering.employeerostering.Skill>
    </skills>
  </employeerostering.employeerostering.Employee>
  <employeerostering.employeerostering.Employee>
    <name>Mary</name>
    <skills>
      <employeerostering.employeerostering.Skill>
        <name>writing</name>
      </employeerostering.employeerostering.Skill>
    </skills>
  </employeerostering.employeerostering.Employee>
  <employeerostering.employeerostering.Employee>
    <name>Petr</name>
    <skills>
      <employeerostering.employeerostering.Skill>
        <name>speaking</name>
      </employeerostering.employeerostering.Skill>
    </skills>
  </employeerostering.employeerostering.Employee>
 </employeeList>
 <shiftList>
  <employeerostering.employeerostering.Shift>
    <timeslot>
      <startTime>2017-01-01T00:00:00</startTime>
      <endTime>2017-01-01T01:00:00</endTime>
    </timeslot>
    <requiredSkill
reference="../../../employeeList/employeerostering.employeerostering.Employee/skills/emplo
yeerostering.employeerostering.Skill"/>
  </employeerostering.employeerostering.Shift>
  <employeerostering.employeerostering.Shift>
    <timeslot reference="../../employeerostering.employeerostering.Shift/timeslot"/>
    <requiredSkill
reference="../../../employeeList/employeerostering.employeerostering.Employee[3]/skills/emp
loyeerostering.employeerostering.Skill"/>
  </employeerostering.employeerostering.Shift>
  <employeerostering.employeerostering.Shift>
    <timeslot reference="../../employeerostering.employeerostering.Shift/timeslot"/>
    <requiredSkill
reference="../../../employeeList/employeerostering.employeerostering.Employee[2]/skills/emp
loyeerostering.employeerostering.Skill"/>
  </employeerostering.employeerostering.Shift>
 </shiftList>
 <skillList>
  <employeerostering.employeerostering.Skill
reference="../../employeeList/employeerostering.employeerostering.Employee/skills/employe
erostering.employeerostering.Skill"/>
  <employeerostering.employeerostering.Skill
reference="../../employeeList/employeerostering.employeerostering.Employee[3]/skills/emplo
yeerostering.employeerostering.Skill"/>
  <employeerostering.employeerostering.Skill
reference="../../employeeList/employeerostering.employeerostering.Employee[2]/skills/emplo
yeerostering.employeerostering.Skill"/>
 </skillList>
 <timeslotList>
  <employeerostering.employeerostering.Timeslot
```

```
reference="../../shiftList/employeerostering.employeerostering.Shift/timeslot"/>
  </timeslotList>
  <dayOffRequestList/>
  <shiftAssignmentList>
    <employeerostering.employeerostering.ShiftAssignment>
      <shift reference="../../../shiftList/employeerostering.employeerostering.Shift"/>
    </employeerostering.employeerostering.ShiftAssignment>
    <employeerostering.employeerostering.ShiftAssignment>
      <shift reference="../../../shiftList/employeerostering.employeerostering.Shift[3]"/>
    </employeerostering.employeerostering.ShiftAssignment>
    <employeerostering.employeerostering.ShiftAssignment>
      <shift reference="../../../shiftList/employeerostering.employeerostering.Shift[2]"/>
    </employeerostering.employeerostering.ShiftAssignment>
  </shiftAssignmentList>
</employeerostering.employeerostering.EmployeeRoster>
```

3. *请求最佳解决方案以寻求规划问题：*

GET

http://localhost:8080/kie-server/services/rest/server/containers/employeerostering_1.0.0-SNAPSHOT/solvers/EmployeeRosteringSolver/bestsolution

响应示例

```
<solver-instance>
  <container-id>employee-rostering</container-id>
  <solver-id>solver1</solver-id>
  <solver-config-file>employeerostering/employeerostering/EmployeeRosteringSolverConfig.solver.xml</solver-config-file>
  <status>NOT_SOLVING</status>
  <score scoreClass="org.optaplanner.core.api.score.buildin.hardsoft.HardSoftScore">0hard/0soft</score>
  <best-solution class="employeerostering.employeerostering.EmployeeRoster">
    <employeeList>
      <employeerostering.employeerostering.Employee>
        <name>John</name>
        <skills>
          <employeerostering.employeerostering.Skill>
            <name>reading</name>
          </employeerostering.employeerostering.Skill>
        </skills>
      </employeerostering.employeerostering.Employee>
      <employeerostering.employeerostering.Employee>
        <name>Mary</name>
        <skills>
```

```
        <employeerostering.employeerostering.Skill>
          <name>writing</name>
        </employeerostering.employeerostering.Skill>
      </skills>
    </employeerostering.employeerostering.Employee>
    <employeerostering.employeerostering.Employee>
      <name>Petr</name>
      <skills>
        <employeerostering.employeerostering.Skill>
          <name>speaking</name>
        </employeerostering.employeerostering.Skill>
      </skills>
    </employeerostering.employeerostering.Employee>
  </employeeList>
  <shiftList>
    <employeerostering.employeerostering.Shift>
      <timeslot>
        <startTime>2017-01-01T00:00:00</startTime>
        <endTime>2017-01-01T01:00:00</endTime>
      </timeslot>
      <requiredSkill
reference="../../../employeeList/employeerostering.employeerostering.Employee/skills/emplo
yeerostering.employeerostering.Skill"/>
    </employeerostering.employeerostering.Shift>
    <employeerostering.employeerostering.Shift>
      <timeslot reference="../../employeerostering.employeerostering.Shift/timeslot"/>
      <requiredSkill
reference="../../../employeeList/employeerostering.employeerostering.Employee[3]/skills/emp
loyeerostering.employeerostering.Skill"/>
    </employeerostering.employeerostering.Shift>
    <employeerostering.employeerostering.Shift>
      <timeslot reference="../../employeerostering.employeerostering.Shift/timeslot"/>
      <requiredSkill
reference="../../../employeeList/employeerostering.employeerostering.Employee[2]/skills/emp
loyeerostering.employeerostering.Skill"/>
    </employeerostering.employeerostering.Shift>
  </shiftList>
  <skillList>
    <employeerostering.employeerostering.Skill
reference="../../employeeList/employeerostering.employeerostering.Employee/skills/employe
erostering.employeerostering.Skill"/>
    <employeerostering.employeerostering.Skill
reference="../../employeeList/employeerostering.employeerostering.Employee[3]/skills/emplo
yeerostering.employeerostering.Skill"/>
    <employeerostering.employeerostering.Skill
reference="../../employeeList/employeerostering.employeerostering.Employee[2]/skills/emplo
yeerostering.employeerostering.Skill"/>
  </skillList>
  <timeslotList>
    <employeerostering.employeerostering.Timeslot
reference="../../shiftList/employeerostering.employeerostering.Shift/timeslot"/>
  </timeslotList>
  <dayOffRequestList/>
  <shiftAssignmentList/>
```

```
    <score>0hard/0soft</score>
  </best-solution>
</solver-instance>
```

第 34 章 OPTAPLANNER 和 QUARKUS 入门

您可以使用 https://code.quarkus.redhat.com 网站生成红帽构建的 OptaPlanner Quarkus Maven 项目，并自动添加并配置要在应用程序中使用的扩展。然后您可以下载 Quarkus Maven 存储库，或使用您项目中的在线 Maven 存储库。

34.1. APACHE MAVEN 和 RED HAT BUILD OF QUARKUS

Apache Maven 是 Java 应用程序开发中使用的分布式构建自动化工具，用于创建、管理和构建软件项目。Maven 使用名为 Project Object Model(POM)文件的标准配置文件来定义项目并管理构建流程。POM 文件描述使用 XML 文件生成的项目打包和输出的模块和组件依赖关系、构建顺序和目标。这可确保以正确、一致的方式构建项目。

Maven 存储库

Maven 存储库存储 Java 库、插件和其他构建构件。默认公共存储库是 Maven 2 中央存储库，但存储库可以是专有仓库和内部的，以在开发团队之间共享常见工件。还可从第三方提供存储库。

您可以将在线 Maven 存储库与 Quarkus 项目一起使用，也可以下载红帽 Quarkus Maven 存储库构建。

Maven 插件

Maven 插件是定义实现一个或多个目标的 POM 文件的组成部分。Quarkus 应用程序使用以下 Maven 插件：

- Quarkus Maven 插件(quarkus-maven-plugin): Enables Maven 创建 Quarkus 项目，支持生成 uber-JAR 文件，并提供开发模式。
- Maven Surefire 插件(maven-surefire-plugin): 在构建生命周期的测试阶段用来对应用程序执行单元测试。插件会生成包含测试报告的文本和 XML 文件。

34.1.1. 为在线存储库配置 Maven settings.xml 文件

您可以通过配置用户 settings.xml 文件，将在线 Maven 存储库用于 Maven 项目。这是推荐的方法。与共享服务器上的存储库管理器或存储库一起使用的 Maven 设置可以提供更好的控制和管理项目。

注意

当您通过修改 Maven settings.xml 文件来配置存储库时，更改会应用到所有 Maven 项目。

流程

1. 在文本编辑器中打开 Maven ~/.m2/settings.xml 文件或集成开发环境(IDE)。

注意

如果 ~/.m2/ 目录中没有 settings.xml 文件，请将 $MAVEN_HOME/.m2/conf/ 目录中的 settings.xml 文件复制到 ~/.m2/ 目录中。

2. 在 settings.xml 文件的 <profiles > 元素中添加以下行：

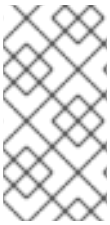
```
<!-- Configure the Maven repository -->
<profile>
 <id>red-hat-enterprise-maven-repository</id>
 <repositories>
  <repository>
   <id>red-hat-enterprise-maven-repository</id>
   <url>https://maven.repository.redhat.com/ga/</url>
   <releases>
    <enabled>true</enabled>
   </releases>
   <snapshots>
    <enabled>false</enabled>
   </snapshots>
  </repository>
 </repositories>
 <pluginRepositories>
  <pluginRepository>
   <id>red-hat-enterprise-maven-repository</id>
   <url>https://maven.repository.redhat.com/ga/</url>
   <releases>
    <enabled>true</enabled>
   </releases>
   <snapshots>
    <enabled>false</enabled>
   </snapshots>
  </pluginRepository>
 </pluginRepositories>
</profile>
```

3. 将以下行添加到 settings.xml 文件的 < activeProfiles > 元素，并保存文件。

```
<activeProfile>red-hat-enterprise-maven-repository</activeProfile>
```

34.1.2. 下载并配置 Quarkus Maven 存储库

如果您不想使用在线 Maven 存储库，您可以下载并配置 Quarkus Maven 存储库，以使用 Maven 创建 Quarkus 应用程序。Quarkus Maven 存储库包含许多 Java 开发人员通常用来构建应用程序的要求。这个步骤描述了如何编辑 settings.xml 文件来配置 Quarkus Maven 存储库。



注意

当您通过修改 Maven settings.xml 文件来配置存储库时，更改会应用到所有 Maven 项目。

流程

1. 从红帽客户门户网站的 Software Downloads 页面（需要登录）下载 Quarkus Maven repository ZIP 文件。

2. 展开下载的存档。

3. 将目录改为 ~/.m2/ 目录，并在文本编辑器或集成开发环境(IDE)中打开 Maven settings.xml 文件。

4. 将以下行添加到 settings.xml 文件的 <profiles > 元素，其中 QUARKUS_MAVEN_REPOSITORY 是您下载的 Quarkus Maven 存储库的路径。QUARKUS_MAVEN_REPOSITORY 的格式必须是 file://$PATH，如 file:///home/userX/rh-quarkus-2.13.GA-maven-repository/maven-repository。

```
<!-- Configure the Quarkus Maven repository -->
<profile>
  <id>red-hat-enterprise-maven-repository</id>
  <repositories>
    <repository>
      <id>red-hat-enterprise-maven-repository</id>
      <url>QUARKUS_MAVEN_REPOSITORY</url>
      <releases>
        <enabled>true</enabled>
      </releases>
      <snapshots>
```

```
      <enabled>false</enabled>
    </snapshots>
  </repository>
 </repositories>
 <pluginRepositories>
  <pluginRepository>
   <id>red-hat-enterprise-maven-repository</id>
   <url>QUARKUS_MAVEN_REPOSITORY</url>
   <releases>
    <enabled>true</enabled>
   </releases>
   <snapshots>
    <enabled>false</enabled>
   </snapshots>
  </pluginRepository>
 </pluginRepositories>
</profile>
```

5. 将以下行添加到 settings.xml 文件的 < activeProfiles > 元素，并保存文件。

```
<activeProfile>red-hat-enterprise-maven-repository</activeProfile>
```

重要

如果您的 Maven 存储库包含过时的工件，您可能会在构建或部署项目时遇到以下 Maven 错误消息之一，其中 ARTIFACT_NAME 是缺少的工件和 PROJECT_NAME 的名称，即您要构建的项目的名称：

- 缺少工件 PROJECT_NAME
- [ERROR] Failed on project ARTIFACT_NAME; Could not resolve dependencies for PROJECT_NAME

要解决这个问题，删除位于 ~/.m2/repository 目录中的本地存储库的缓存版本，以强制下载最新的 Maven 工件。

34.2. 使用 MAVEN 插件创建 OPTAPLANNER RED HAT BUILD OF QUARKUS MAVEN 项目

您可以使用 Apache Maven 和 Quarkus Maven 插件，使用红帽构建的 OptaPlanner 和 Quarkus 应用程序启动并运行。

先决条件

- 安装了 OpenJDK 11 或更高版本。红帽构建的 Open JDK 可从红帽客户门户网站的 Software Downloads 页面获取（需要登录）。

- 已安装 Apache Maven 3.6 或更高版本。Maven 可以从 Apache Maven Project 网站获得。

流程

1. 在命令终端中，输入以下命令验证 Maven 是否使用 JDK 11，并且 Maven 版本为 3.6 或更高版本：

```
mvn --version
```

2. 如果前面的命令没有返回 JDK 11，请将到 JDK 11 的路径添加到 PATH 环境变量，然后再次输入前面的命令。

3. 要生成 Quarkus OptaPlanner quickstart 项目，请输入以下命令：

```
mvn com.redhat.quarkus.platform:quarkus-maven-plugin:2.13.Final-redhat-00006:create \
    -DprojectGroupId=com.example \
    -DprojectArtifactId=optaplanner-quickstart  \
    -Dextensions="resteasy,resteasy-jackson,optaplanner-quarkus,optaplanner-quarkus-jackson" \
    -DplatformGroupId=com.redhat.quarkus.platform
    -DplatformVersion=2.13.Final-redhat-00006 \
    -DnoExamples
```

此命令在 ./optaplanner-quickstart 目录中创建以下元素：

- Maven 结构

- src/main/docker中的 Dockerfile 文件示例

- 应用程序配置文件

表 34.1. mvn io.quarkus:quarkus-maven-plugin:2.13.Final-redhat-00006:create 命令中使用的属性

属性	描述
projectGroupId	项目的组 ID。
projectArtifactId	项目的工件 ID。
extensions	用于此项目的以逗号分隔的 Quarkus 扩展列表。如需 Quarkus 扩展的完整列表，请在命令行中输入 **mvn quarkus:list-extensions**。
noExamples	使用项目结构创建项目，但不包括测试或类。

projectGroupID 和 projectArtifactID 属性的值用于生成项目版本。默认项目版本为 1 InventoryServiceSNAPSHOT。

4. ***要查看您的 OptaPlanner 项目，请将目录改为 OptaPlanner Quickstarts 目录：***

```
cd optaplanner-quickstart
```

5. ***检查 pom.xml 文件。内容应类似以下示例：***

```
<dependencyManagement>
 <dependencies>
  <dependency>
   <groupId>io.quarkus.platform</groupId>
   <artifactId>quarkus-bom</artifactId>
   <version>2.13.Final-redhat-00006</version>
   <type>pom</type>
   <scope>import</scope>
  </dependency>
  <dependency>
   <groupId>io.quarkus.platform</groupId>
   <artifactId>quarkus-optaplanner-bom</artifactId>
   <version>2.13.Final-redhat-00006</version>
   <type>pom</type>
   <scope>import</scope>
  </dependency>
 </dependencies>
</dependencyManagement>
<dependencies>
 <dependency>
  <groupId>io.quarkus</groupId>
  <artifactId>quarkus-resteasy</artifactId>
 </dependency>
 <dependency>
  <groupId>io.quarkus</groupId>
  <artifactId>quarkus-resteasy-jackson</artifactId>
```

```
    </dependency>
    <dependency>
      <groupId>org.optaplanner</groupId>
      <artifactId>optaplanner-quarkus</artifactId>
    </dependency>
    <dependency>
      <groupId>org.optaplanner</groupId>
      <artifactId>optaplanner-quarkus-jackson</artifactId>
    </dependency>
    <dependency>
      <groupId>io.quarkus</groupId>
      <artifactId>quarkus-junit5</artifactId>
      <scope>test</scope>
    </dependency>
  </dependencies>
```

34.3. 使用 CODE.QUARKUS.REDHAT.COM 创建 QUARKUS MAVEN 项目

您可以使用 code.quarkus.redhat.com 网站生成红帽构建的 OptaPlanner Quarkus Maven 项目，并自动添加并配置要在应用程序中使用的扩展。另外，code.quarkus.redhat.com 会自动管理将项目编译到原生可执行文件所需的配置参数。

本节介绍了如何生成 OptaPlanner Maven 项目并包含以下主题：

- 指定应用程序的基本详情。
- 选择您要包含在项目中的扩展。
- 使用您的项目文件生成可下载归档。
- 使用自定义命令编译和启动应用程序。

先决条件

- 您有一个 Web 浏览器。

流程

1. 在网页浏览器中打开 https://code.quarkus.redhat.com：

2. 指定项目详情：

3. 输入项目的组名称。名称的格式遵循 Java 软件包命名约定，如 com.example。

4. 输入您要用于从项目生成的 Maven 工件的名称，如 code-with-quarkus。

5. 选择 Build Tool > Maven 指定要创建 Maven 项目。您选择的构建工具决定了项目：

   - 生成的项目的目录结构
   - 生成的项目中使用的配置文件格式
   - 用于编译和启动应用程序的自定义构建脚本和命令，会在您生成项目后为您显示 code.quarkus.redhat.com

   注意

   红帽只支持使用 code.quarkus.redhat.com 创建 OptaPlanner Maven 项目。红帽不支持生成 Gradle 项目。

6. 输入要在项目生成的工件中使用的版本。此字段的默认值为 1.5.0.SNAPSHOT。建议使用语义版本，但若愿意，您可以使用不同类型的版本。

7. 输入构建工具在打包项目时生成的工件名称。

   根据 Java 软件包命名惯例，软件包名称应与软件包名称匹配您用于项目的组名称，但您可以指定不同的名称。

   注意

   code.quarkus.redhat.com 网站自动使用 OptaPlanner 的最新版本。您可以在生成项目后手动更改 pom.xml 文件中的 BOM 版本。

8. 选择以下扩展以作为依赖项包含：

- RESTEasy JAX-RS (quarkus-resteasy)
- resteasy Jackson(quarkus-resteasy-jackson)
- OptaPlanner AI constraint solver(optaplanner-quarkus)
- OptaPlanner Jackson(optaplanner-quarkus-jackson)

红帽为列表上的单个扩展提供不同级别的支持，这些扩展由每个扩展名称旁的标签来表示：

- 红帽完全支持 SUPPORTED 扩展用于生产环境中的企业应用程序。
- TECH-PREVIEW 扩展受红帽在生产环境中的支持，在 技术预览功能支持范围 下受到红帽的支持。
- 红帽在生产环境中不支持 DEV-SUPPORT 扩展，但红帽提供的核心功能则由红帽开发人员用于开发新的应用程序。
- 计划使用具有相同功能的较新的技术或实现替换 DEPRECATED 扩展。

红帽不支持在生产环境中使用的未标记扩展。

9. 选择 Generate your application 确认您的选择并显示覆盖屏幕，其中包含包含您生成的项目的存档的下载链接。overlay 屏幕还显示可用于编译和启动应用程序的自定义命令。

10. 选择 Download the ZIP，以使用生成的项目文件将存档保存到您的系统中。

11. 提取存档的内容。

12. 进入包含您提取的项目文件的目录：

```
cd <directory_name>
```

13. 以开发模式编译并启动应用程序：

```
./mvnw compile quarkus:dev
```

34.4. 使用 QUARKUS CLI 创建红帽 QUARKUS MAVEN 项目构建

您可以使用 Quarkus 命令行界面(CLI)创建 Quarkus OptaPlanner 项目。

先决条件

- 已安装 Quarkus CLI。如需更多信息，请参阅使用 Quarkus 命令行界面 构建 Quarkus 应用程序。

流程

1. 创建 Quarkus 应用程序：

```
quarkus create app -P io.quarkus:quarkus-bom:2.13.Final-redhat-00006
```

2. 要查看可用的扩展，请输入以下命令：

```
quarkus ext -i
```

这个命令返回以下扩展：

```
optaplanner-quarkus
optaplanner-quarkus-benchmark
optaplanner-quarkus-jackson
optaplanner-quarkus-jsonb
```

3. 输入以下命令在项目的 pom.xml 文件中添加扩展：

```
quarkus ext add resteasy-jackson
quarkus ext add optaplanner-quarkus
quarkus ext add optaplanner-quarkus-jackson
```

4. **在文本编辑器中打开 pom.xml 文件。该文件应该类似以下示例：**

```
<?xml version="1.0"?>
<project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd" xmlns="http://maven.apache.org/POM/4.0.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.acme</groupId>
  <artifactId>code-with-quarkus-optaplanner</artifactId>
  <version>1.0.0-SNAPSHOT</version>
  <properties>
 <compiler-plugin.version>3.8.1</compiler-plugin.version>
 <maven.compiler.parameters>true</maven.compiler.parameters>
 <maven.compiler.source>11</maven.compiler.source>
 <maven.compiler.target>11</maven.compiler.target>
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
 <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
 <quarkus.platform.artifact-id>quarkus-bom</quarkus.platform.artifact-id>
 <quarkus.platform.group-id>io.quarkus</quarkus.platform.group-id>
 <quarkus.platform.version>2.13.Final-redhat-00006</quarkus.platform.version>
 <surefire-plugin.version>3.0.0-M5</surefire-plugin.version>
  </properties>
  <dependencyManagement>
 <dependencies>
   <dependency>
     <groupId>${quarkus.platform.group-id}</groupId>
     <artifactId>${quarkus.platform.artifact-id}</artifactId>
     <version>${quarkus.platform.version}</version>
     <type>pom</type>
     <scope>import</scope>
   </dependency>
   <dependency>
     <groupId>io.quarkus.platform</groupId>
     <artifactId>optaplanner-quarkus</artifactId>
     <version>2.2.2.Final</version>
     <type>pom</type>
     <scope>import</scope>
   </dependency>
 </dependencies>
  </dependencyManagement>
  <dependencies>
 <dependency>
   <groupId>io.quarkus</groupId>
   <artifactId>quarkus-arc</artifactId>
 </dependency>
 <dependency>
   <groupId>io.quarkus</groupId>
   <artifactId>quarkus-resteasy</artifactId>
 </dependency>
```

```
<dependency>
  <groupId>org.optaplanner</groupId>
  <artifactId>optaplanner-quarkus</artifactId>
</dependency>
<dependency>
  <groupId>org.optaplanner</groupId>
  <artifactId>optaplanner-quarkus-jackson</artifactId>
</dependency>
<dependency>
  <groupId>io.quarkus</groupId>
  <artifactId>quarkus-resteasy-jackson</artifactId>
</dependency>
<dependency>
  <groupId>io.quarkus</groupId>
  <artifactId>quarkus-junit5</artifactId>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>io.rest-assured</groupId>
  <artifactId>rest-assured</artifactId>
  <scope>test</scope>
</dependency>
 </dependencies>
 <build>
<plugins>
  <plugin>
    <groupId>${quarkus.platform.group-id}</groupId>
    <artifactId>quarkus-maven-plugin</artifactId>
    <version>${quarkus.platform.version}</version>
    <extensions>true</extensions>
    <executions>
      <execution>
        <goals>
          <goal>build</goal>
          <goal>generate-code</goal>
          <goal>generate-code-tests</goal>
        </goals>
      </execution>
    </executions>
  </plugin>
  <plugin>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>${compiler-plugin.version}</version>
    <configuration>
      <parameters>${maven.compiler.parameters}</parameters>
    </configuration>
  </plugin>
  <plugin>
    <artifactId>maven-surefire-plugin</artifactId>
    <version>${surefire-plugin.version}</version>
    <configuration>
      <systemPropertyVariables>

<java.util.logging.manager>org.jboss.logmanager.LogManager</java.util.logging.manager>
        <maven.home>${maven.home}</maven.home>
      </systemPropertyVariables>
```

```
    </configuration>
   </plugin>
  </plugins>
  </build>
  <profiles>
 <profile>
   <id>native</id>
   <activation>
    <property>
     <name>native</name>
    </property>
   </activation>
   <build>
    <plugins>
     <plugin>
      <artifactId>maven-failsafe-plugin</artifactId>
      <version>${surefire-plugin.version}</version>
      <executions>
       <execution>
        <goals>
         <goal>integration-test</goal>
         <goal>verify</goal>
        </goals>
        <configuration>
         <systemPropertyVariables>
          <native.image.path>${project.build.directory}/${project.build.finalName}-
runner</native.image.path>

<java.util.logging.manager>org.jboss.logmanager.LogManager</java.util.logging.manager>
           <maven.home>${maven.home}</maven.home>
         </systemPropertyVariables>
        </configuration>
       </execution>
      </executions>
     </plugin>
    </plugins>
   </build>
   <properties>
    <quarkus.package.type>native</quarkus.package.type>
   </properties>
 </profile>
  </profiles>
</project>
```

附录 A. 版本信息

文档最新更新于 2024 年 3 月 14 日星期四。

附录 B. 联系信息

Red Hat Process Automation Manager 文档团队：brms-docs@redhat.com