



Red Hat Decision Manager 7.13

Red Hat Decision Manager のスタートガイド

法律上の通知

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

本書では、Red Hat Decision Manager でデシジョンサービスおよびプランニングソリューションを使い始める方法を説明します。

目次

はじめに	4
多様性を受け入れるオープンソースの強化	5
パート I. RED HAT PROCESS AUTOMATION MANAGER でのデシジョンサービスの使用	6
第1章 BUSINESS CENTRAL のプロジェクトおよびビジネスアセットの例	7
1.1. BUSINESS CENTRAL のプロジェクトおよびビジネスアセット例へのアクセス	7
第2章 RED HAT DECISION MANAGER BPMN および DMN モデラー	10
2.1. RED HAT DECISION MANAGER VS CODE 拡張機能バンドルのインストール	10
2.2. RED HAT DECISION MANAGER スタンドアロンのエディターの設定	11
第3章 MAVEN を使用した DMN モデルおよび BPMN モデルの作成および実行	15
第4章 BUSINESS CENTRAL での交通違反プロジェクトの作成	17
第5章 DMN (DECISION MODEL AND NOTATION)	18
5.1. 交通違反 DMN デシジョン要件ダイアグラム (DRD) の作成	18
5.2. 交通違反 DMN カスタムデータタイプの作成	20
5.3. DRD 入力およびデシジョンノードへのカスタムデータタイプの割り当て	24
5.4. 交通違反 DMN デシジョン論理の定義	25
第6章 テストシナリオ	28
6.1. テストシナリオを使用した交通違反のテスト	28
第7章 DMN モデルの実行	32
7.1. KIE SERVER REST API を使用した DMN サービスの実行	32
第8章 関連情報	47
パート II. RED HAT ビルドの OPTAPLANNER のスタートガイド	48
第9章 RED HAT ビルドの OPTAPLANNER の概要	49
9.1. 計画問題	49
9.2. 計画問題での NP 完全	49
9.3. 計画問題に対する解	50
9.4. 計画問題に対する制約	50
9.5. RED HAT ビルドの OPTAPLANNER で提供される例	51
9.6. N クィーン	54
9.7. クラウドバランシング	58
9.8. 巡回セールスマン (TSP - 巡回セールスマン問題)	58
9.9. テニスクラブのスケジュール	59
9.10. 会議のスケジュール	60
9.11. コースの時間割 (ITC 2007 TRACK 3 - カリキュラムのスケジュール)	61
9.12. マシンの再割当て (GOOGLE ROADEF 2012)	63
9.13. 配送経路	66
9.14. プロジェクトジョブのスケジュール	77
9.15. タスクの割り当て	79
9.16. 試験の時間割 (ITC 2007 TRACK 1 - 試験)	81
9.17. 看護師の勤務表 (INRC 2010)	84
9.18. 巡回トーナメント問題 (TTP)	89
9.19. コストを抑えるスケジュール	91
9.20. 投資資産クラスの割り当て (ポートフォリオの最適化)	94
9.21. 会議スケジュール	94

9.22. ロックツアー	97
9.23. 航空機乗組員のスケジューリング	97
第10章 RED HAT ビルドの OPTAPLANNER の例のダウンロード	99
10.1. OPTAPLANNER サンプルの実行	99
10.2. IDE (INTELLIJ、ECLIPSE、または NETBEANS) での RED HAT ビルドの OPTAPLANNER サンプルの実行	100
第11章 BUSINESS CENTRAL での OPTAPLANNER のスタートガイド: 従業員勤務表の例	101
11.1. BUSINESS CENTRAL への従業員勤務表サンプルプロジェクトのデプロイメント	101
11.2. 従業員の勤務表サンプルプロジェクトの再作成	101
11.3. REST API を使用した SOLVER へのアクセス	115
第12章 OPTAPLANNER および QUARKUS の使用ガイド	120
12.1. APACHE MAVEN および RED HAT ビルドの QUARKUS	120
12.2. MAVEN プラグインを使用した OPTAPLANNER RED HAT ビルドの QUARKUS MAVEN プロジェクトの作成	123
12.3. CODE.QUARKUS.REDHAT.COM を使用した RED HAT ビルドの QUARKUS MAVEN プロジェクトの作成	125
12.4. QUARKUS CLI を使用した RED HAT ビルドの QUARKUS MAVEN プロジェクトの作成	127
付録A バージョン情報	131
付録B お問い合わせ先	132

はじめに

ビジネスの意思決定の開発者として、Red Hat Decision Manager を使用してさまざまな利用可能なアセットを使用してデシジョンサービスを開発することができます。Red Hat ビルドの OptaPlanner を使用して、制限されたリソースのセットと特定の制約に基づいて問題の計画に最適なソリューションを見つけることもできます。

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みにより、これらの変更は今後の複数のリリースに対して段階的に実施されます。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

パート I. RED HAT PROCESS AUTOMATION MANAGER でのデシジョンサービスの使用

ビジネスルール開発者は、Red Hat Decision Manager の Business Central または VS Code の Red Hat Decision Manager DMN モデラーを使用して、さまざまなデシジョンサービスを設計できます。Red Hat Decision Manager は、Business Central に直接参考用として、ビジネスアセット例を含むプロジェクトサンプルを提供しています。本書は、Business Central に含まれる **Traffic_Violation** サンプルプロジェクトをもとに、交通違反例を作成してテストする方法を説明します。このサンプルプロジェクトは、Decision Model and Notation (DMN) モデルを使用して、交通違反のデシジョンサービスで運転手の罰則と免許停止のルールを定義します。本書の手順に従い、プロジェクトとプロジェクトに含まれるアセットを作成するか、既存の **Traffic_Violation** サンプルプロジェクトを開き、レビューします。

Red Hat Decision Manager における DMN コンポーネントと実装についての情報は、[DMN モデルを使用したデシジョンサービスの作成](#) を参照してください。

前提条件

- Red Hat JBoss Enterprise Application Platform 7.4 がインストールされている。詳細は、[Red Hat JBoss EAP 7.4 インストールガイド](#) を参照してください。
- Red Hat Process Automation Manager がインストールされ、KIE Server で設定されている。詳細は、[Red Hat JBoss EAP 7.4 への Red Hat Decision Manager のインストールおよび設定](#) を参照してください。
- Red Hat Decision Manager が稼働し、**developer** ロールで Business Central にログインできる。詳細は、[Planning a Red Hat Decision Manager installation](#) を参照してください。

第1章 BUSINESS CENTRAL のプロジェクトおよびビジネスアセットの例

Business Central には、プロジェクト例がビジネスアセット例と合わせて同梱されており、ルールや他のアセットを自分の Red Hat Decision Manager プロジェクトに作成するときに参考として使用できます。各プロジェクトは、Red Hat Decision Manager の意思決定管理やビジネス最適化アセットおよび論理を異なる方法で説明するように設計されています。



注記

Red Hat は、Red Hat Decision Manager ディストリビューションに含まれるコードサンプルのサポートはしていません。

以下のプロジェクト例が、Business Central で利用できます。

- **Course_Scheduling**: (ビジネス最適化) コースのスケジュールとカリキュラム決定プロセス。授業を部屋に割り当て、当然の競合やクラス部屋のキャパシティーなどの要因に基づいて学生のラボを決定します。
- **Dinner_Party**: (ビジネス最適化) ガイド付きデシジョンテーブルを使用したゲストの座席割り当ての最適化。各ゲストの職種、政治的信条、既知の関係を基にしてゲストに座席を割り当てます。
- **Employee_Rostering (従業員勤務表)**: (ビジネス最適化) デシジョンおよびソルバーアセットを使用した従業員勤務表の最適化。スキルに基づいて従業員をシフトに割り当てます。
- **Evaluation_Process**: (プロセス自動化) ビジネスプロセスアセットを使用したプロセス評価。実績に基づいて従業員を評価します。
- **IT_Orders**: (プロセス自動化およびケース管理) ビジネスプロセスとケース管理アセットを使用したケース注文。ニーズと承認に基づいて IT ハードウェアを注文します。
- **Mortgages (住宅ローン)**: (ルールでのデシジョン管理) ルールベースのデシジョンアセットを使用した住宅ローン審査プロセス。申し込み者のデータと資格を基にローンの申し込み資格を判定します。
- **Mortgage_Process (住宅ローン)**: (プロセス自動化) ビジネスプロセスとデシジョンアセットを使用した住宅ローン審査プロセス。申し込み者のデータと資格を基にローンの申し込み資格を判定します。
- **OptaCloud**: (ビジネス最適化) デシジョンおよびソルバーアセットを使用したリソース割り当ての最適化。リソースが制限されるなかでプロセスをコンピューターに割り当てます。
- **Traffic_Violation (交通違反)**: (DMN でのデシジョン管理) Decision Model and Notation (DMN) モデルを使用した交通違反のデシジョンサービス。交通違反をもとに運転手の罰則および免許停止を判断します。

1.1. BUSINESS CENTRAL のプロジェクトおよびビジネスアセット例へのアクセス

Business Central のプロジェクト例を使用すると、自分の Red Hat Decision Manager プロジェクトにルールや他のアセットを作成するときに、参考としてビジネスアセットを確認できます。

前提条件

- Business Central をインストールし、実行している。インストールオプションについては、[Planning a Red Hat Decision Manager installation](#) を参照してください。

手順

1. Business Central で、**Menu** → **Design** → **Projects** に移動します。既存のプロジェクトがある場合は、**MySpace** のデフォルトのスペースをクリックして、**Add Project** ドロップダウンメニューから **Try Samples** を選択して、サンプルにアクセスできます。既存のプロジェクトがない場合には、**Try samples** をクリックします。
2. 各サンプルプロジェクトの説明を読んで、どのプロジェクトが最適か確認します。各プロジェクトは、Red Hat Decision Manager の意思決定管理やビジネス最適化アセットおよび論理を異なる方法で説明するように設計されています。
3. サンプルプロジェクトを選択し、**Ok** をクリックして自分のスペースにプロジェクトを追加します。
4. 自分のスペースの **Projects** ページで、サンプルプロジェクトの1つを選択して、そのプロジェクトのアセットを表示します。
5. 各アセットを選択して、指定のゴールまたはワークフローに到達するためにプロジェクトがどのように設計されているのかを確認します。サンプルのプロジェクトには、アセットが複数ページ含まれているものもあります。右上隅の左向きまたは右向き矢印をクリックして、全アセットリストを表示します。

図1.1 アセットページの選択



6. プロジェクトの **Assets** ページの右上隅にある **Build** をクリックしてサンプルプロジェクトをビルドするか、**Deploy** をクリックしてプロジェクトをビルドしてから、KIE Server にデプロイします。



注記

Build & Install オプションを選択してプロジェクトをビルドし、KJAR ファイルを KIE Server にデプロイせずに設定済みの Maven リポジトリに公開することもできます。開発環境では、**Deploy** をクリックすると、ビルドされた KJAR ファイルを KIE Server に、実行中のインスタンス (がある場合はそれ) を停止せずにデプロイできます。または **Redeploy** をクリックして、ビルドされた KJAR ファイルをデプロイしてすべてのインスタンスを置き換えることもできます。次回、ビルドされた KJAR ファイルをデプロイまたは再デプロイすると、以前のデプロイメントユニット (KIE コンテナ) が同じターゲット KIE Server で自動的に更新されます。実稼働環境では **Redeploy** オプションは無効になっており、**Deploy** をクリックして、ビルドされた KJAR ファイルを KIE Server 上の新規デプロイメントユニット (KIE コンテナ) にデプロイすることのみが可能です。

KIE Server の環境モードを設定するには、**org.kie.server.mode** システムプロパティを **org.kie.server.mode=development** または **org.kie.server.mode=production** に設定します。Business Central でそれぞれのプロジェクトのデプロイメント動作を設定するには、プロジェクトの **Settings** → **General Settings** → **Version** に移動し、**Development Mode** オプションを選択して、**Save** をクリックします。デフォルトでは、KIE Server および Business Central のすべての新規プロジェクトは開発モードになっています。**Development Mode** をオンにしたプロジェクトをデプロイしたり、実稼働モードになっている KIE Server に手動で **SNAPSHOT** バージョンの接尾辞を追加したプロジェクトをデプロイしたりすることはできません。

プロジェクトのデプロイメントに関する詳細を確認するには、画面の上部にあるデプロイメントバナーの **View deployment details** か、**Deploy** のドロップダウンメニューをクリックします。このオプションを使用すると、**Menu** → **Deploy** → **Execution Servers** ページに移動します。

第2章 RED HAT DECISION MANAGER BPMN および DMN モデラー

Red Hat Decision Manager は、グラフィカルモデラーを使用して Business Process Model and Notation (BPMN) プロセスモデルと、Decision Model and Notation (DMN) デシジョンモデルを設計するのに使用できる次の拡張機能またはアプリケーションを提供します。

- Business Central:** 関連する埋め込みデザイナーで、BPMN モデル、DMN モデル、およびテストシナリオファイルを表示および設計できます。
 Business Central を使用するには、Business Central を含む開発環境を設定してビジネスルールおよびプロセスを作成し、KIE Server を作成して、作成したビジネスルールとプロセスを実行およびテストします。
- Red Hat Decision Manager VS Code 拡張機能** Visual Studio Code (VS Code) で BPMN モデル、DMN モデル、およびテストシナリオファイルを表示して、作成できるようにします。VS Code 拡張機能には VS Code 1.46.0 以降が必要です。
 Red Hat Decision Manager VS Code 拡張機能をインストールするには、VS Code で **Extensions** メニューオプションを選択して、**Red Hat Business Automation Bundle** 拡張機能を検索し、インストールします。
- スタンドアロン BPMN および DMN エディター:** Web アプリケーションに組み込まれた BPMN モデルおよび DMN モデルを表示して、作成できます。必要なファイルをダウンロードするには、[NPM レジストリー](#) から NPM アーティファクトを使用するか、https://<YOUR_PAGE>/dmn/index.js (DMN スタンドアロンのエディターライブラリーの場合)、または https://<YOUR_PAGE>/bpmn/index.js (BPMN スタンドアロンエディターライブラリーの場合) で JavaScript ファイルを直接ダウンロードします。

2.1. RED HAT DECISION MANAGER VS CODE 拡張機能バンドルのインストール

Red Hat Decision Manager は、**Red Hat Business Automation Bundle** VS Code 拡張機能を提供します。これにより、Decision Model and Notation (DMN) デシジョンモデル、Business Process Model and Notation (BPMN) 2.0 ビジネスプロセス、およびテストシナリオを VS Code で直接作成できます。VS Code は、新しいビジネスアプリケーションを開発するために推奨される統合開発環境 (IDE) です。Red Hat Decision Manager は、必要に応じて DMN サポートまたは BPMN サポートに VS Code 拡張機能である **DMN Editor** および **BPMN Editor** をそれぞれ提供します。



重要

VS Code のエディターは、Business Central のエディターと部分的に互換性があり、VS Code では複数の Business Central 機能がサポートされていません。

前提条件

- VS Code** の最新の安定版がインストールされている。

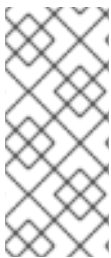
手順

- VS Code IDE で **Extensions** メニューオプションを選択し、DMN、BPMN、およびテストシナリオファイルのサポートに対して **Red Hat Business Automation Bundle** を検索します。
 DMN ファイルまたは BPMN ファイルだけをサポートする場合は、**DMN Editor** または **BPMN Editor** 拡張機能をそれぞれ検索することもできます。

2. Red Hat Business Automation Bundle 拡張機能が VS Code に表示される際に、これを選択し、Install をクリックします。
3. VS Code エディターの動作を最適化するには、拡張機能のインストールが完了した後に、VS Code のインスタンスを再度読み込み、閉じるか、再起動します。

VS Code 拡張バンドルをインストールした後、VS Code で開くか作成するすべての **.dmn** ファイル、**.bpmn** ファイル、または **.bpmn2** ファイルがグラフィカルモデルとして自動的に表示されます。さらに、開くまたは作成する **.scsim** ファイルが、ビジネスデシジョンの機能をテストするテーブルテストシナリオモデルとして自動的に表示されます。

DMN、BPMN、またはテストシナリオモデラーが DMN、BPMN、またはテストシナリオファイルの XML ソースのみを開き、エラーメッセージが表示される場合は、報告されたエラーおよびモデルファイルを確認して、すべての要素が正しく定義されていることを確認します。



注記

新しい DMN モデルまたは BPMN モデルの場合は、Web ブラウザーで **dmn.new** または **bpmn.new** を入力して、オンラインモデラーで DMN モデルまたは BPMN モデルを設計することもできます。モデルの作成が終了したら、オンラインモデラーページで **Download** をクリックして、DMN ファイルまたは BPMN ファイルを VSCode の Red Hat Decision Manager プロジェクトにインポートできます。

2.2. RED HAT DECISION MANAGER スタンドアロンのエディターの設定

Red Hat Decision Manager は、自己完結型のライブラリーに分散されたスタンドアロンのエディターを提供し、エディターごとにオールインワンの JavaScript ファイルを提供します。JavaScript ファイルは、包括的な API を使用してエディターを設定および制御します。

以下の方法を使用して、スタンドアロンのエディターをインストールします。

- 各 JavaScript ファイルを手動でダウンロード
- NPM パッケージの使用

手順

1. 以下の方法のいずれかを使用して、スタンドアロンのエディターをインストールします。
各 JavaScript ファイルを手動でダウンロード: この方法の場合は、以下の手順に従います。
 - a. JavaScript ファイルをダウンロードします。
 - b. ダウンロードした Javascript ファイルをホスト型アプリケーションに追加します。
 - c. 以下の **<script>** タグを HTML ページに追加します。

DMN エディターの HTML ページのスクリプトタグ

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

BPMN エディターの HTML ページのスクリプトタグ

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

NPM パッケージの使用: この方法の場合は、以下の手順に従います。

- a. NPM パッケージを **package.json** ファイルに追加します。

NPM パッケージの追加

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

- b. 各エディターライブラリーを **TypeScript** ファイルにインポートします。

各エディターのインポート

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

2. スタンドアロンのエディターをインストールしたら、以下の例のように提供されたエディター API を使用して必要なエディターを開き、DMN エディターを開きます。API は、各エディターで同じものになります。

DMN スタンドアロンのエディターを開く

```
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("")
      }
    ]
  ])
});
```

エディター API で以下のパラメーターを使用します。

表2.1 パラメーターの例

パラメーター	説明
container	エディターが追加される HTML 要素。

パラメーター	説明
initialContent	DMN モデルのコンテンツへの Promise。以下の例のように、このパラメーターは空にすることができます。 <ul style="list-style-type: none"> ● <code>Promise.resolve("")</code> ● <code>Promise.resolve("<DIAGRAM_CONTENT_DIRECTLY_HERE>")</code> ● <code>fetch("MyDmnModel.dmn").then(content => content.text())</code>
readonly (任意)	エディターでの変更を許可します。コンテンツの編集を許可する場合は false (デフォルト)、エディターで読み取り専用モードの場合は true に設定します。
origin (任意)	リポジトリの起点。デフォルト値は window.location.origin です。
resources (任意)	エディターのリソースのマッピング。たとえば、このパラメーターを使用して、BPMN エディターの DMN エディターまたは作業アイテム定義に含まれるモデルを提供します。マップの各エントリには、リソース名と、 content-type (text または binary) および content (initialContent パラメーターと同様) で設定されるオブジェクトが含まれています。

返されるオブジェクトには、エディターの操作に必要なメソッドが含まれます。

表2.2 返されたオブジェクトメソッド

メソッド	説明
getContent(): Promise<string>	エディターのコンテンツを含む promise を返します。
setContent(path: string, content: string): void	エディターの内容を設定します。
getPreview(): Promise<string>	現在のダイアグラムの SVG 文字列が含まれる promise を返します。
subscribeToContentChanges(callback: (isDirty: boolean) => void): (isDirty: boolean) => void	エディターでコンテンツを変更し、サブスクリプション解除に使用されるのと同じコールバックを返す際に呼び出されるコールバックを設定します。
unsubscribeToContentChanges(callback: (isDirty: boolean) => void): void	エディターでコンテンツが変更される際に渡されたコールバックのサブスクリプションを解除します。

メソッド	説明
markAsSaved(): void	エディターの内容が保存されることを示すエディターの状態をリセットします。また、コンテンツの変更に関連するサブスクライブされたコールバックをアクティベートします。
undo(): void	エディターの最後の変更を元に戻します。また、コンテンツの変更に関連するサブスクライブされたコールバックをアクティベートします。
redo(): void	エディターで、最後に元に戻した変更をやり直します。また、コンテンツの変更に関連するサブスクライブされたコールバックをアクティベートします。
close(): void	エディターを終了します。
getElementPosition(selector: string): Promise<Rect>	要素をキャンバスまたはビデオコンポーネント内に置いた場合に、標準のクエリーセクターを拡張する方法を提供します。 selector パラメーターは、 Canvas:::MySquare 、 Video:::PresenterHand などの <PROVIDER>:::<SELECT> 形式に従う必要があります。このメソッドは、要素の位置を表す Rect を返します。
envelopeApi: MessageBusClientApi<KogitoEditorEnvelopeApi>	これは高度なエディター API です。高度なエディター API の詳細は、 MessageBusClientApi および KogitoEditorEnvelopeApi を参照してください。

第3章 MAVEN を使用した DMN モデルおよび BPMN モデルの作成および実行

Maven アーキタイプを使用して、Business Central ではなく Red Hat Decision Manager VS Code 拡張機能を使用して、VS Code で DMN モデルおよび BPMN モデルを開発できます。その後、必要に応じて、Business Central で、アーキタイプを Red Hat Decision Manager のデシジョンサービスおよびプロセスサービスに統合できます。DMN モデルおよび BPMN モデルを開発する方法は、Red Hat Decision Manager VS Code 拡張機能を使用して新規ビジネスアプリケーションを構築する場合に便利です。

手順

1. コマンドターミナルで、新しい Red Hat Decision Manager プロジェクトを保存するローカルディレクトリーに移動します。
2. Maven アーキタイプを使用して、定義されたフォルダー内のプロジェクトを生成するには、次のコマンドを入力します。

Maven アーキタイプを使用したプロジェクトの生成

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

このコマンドにより、必要な依存関係で Maven プロジェクトが生成され、ビジネスアプリケーションを構築するのに必要なディレクトリーとファイルが生成されます。プロジェクトの開発には、バージョン管理システム Git) 推奨を使用することができます。

同じディレクトリーに複数のプロジェクトを生成する場合は、直前のコマンドに **-DgroupId=<groupid> -DartifactId=<artifactId>** を追加して、生成されたビジネスアプリケーションの **artifactId** および **groupId** を指定できます。

3. VS Code IDE で **File** をクリックし、**Open Folder** を選択し、直前のコマンドを使用して生成されたディレクトリーに移動します。
4. 最初のアセットを作成する前に、ビジネスアプリケーションのパッケージ (例: **org.kie.businessapp**) を設定し、以下のパスにそれぞれのディレクトリーを作成します。

- **PROJECT_HOME/src/main/java**
- **PROJECT_HOME/src/main/resources**
- **PROJECT_HOME/src/test/resources**

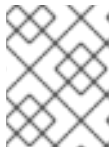
たとえば、**org.kie.businessapp** パッケージの **PROJECT_HOME/src/main/java/org/kie/businessapp** を作成できます。

5. VS Code を使用して、ビジネスアプリケーションにアセットを作成します。以下の方法で、Red Hat Decision Manager VS Code 拡張機能がサポートするアセットを作成できます。
 - ビジネスプロセスを作成するには、**PROJECT_HOME/src/main/resources/org/kie/businessapp** ディレクトリーに、**.bpmn** または **.bpmn2** の新規ファイルを作成します (例: **Process.bpmn**)。

- DMN モデルを作成するには、**PROJECT_HOME/src/main/resources/org/kie/businessapp** ディレクトリーに、**.dmn** の新規ファイルを作成します (例: **AgeDecision.dmn**)。
 - テストシナリオシミュレーションモデルを作成するには、**PROJECT_HOME/src/test/resources/org/kie/businessapp** ディレクトリーに、**.scesim** の新規ファイルを作成します (例: **TestAgeScenario.scesim**)。
6. Maven アーキタイプでアセットを作成したら、コマンドラインで (**pom.xml** がある) プロジェクトのルートディレクトリーに移動し、以下のコマンドを実行してプロジェクトのナレッジ JAR (KJAR) を構築します。

```
mvn clean install
```

ビルドに失敗したら、コマンドラインのエラーメッセージに記載されている問題に対応し、ビルドに成功するまでプロジェクトの妥当性確認を行います。ただし、ビルドに成功すると、**PROJECT_HOME/target** ディレクトリーでビジネスアプリケーションのアーティファクトを確認できます。



注記

mvn clean install コマンドを使用して、開発中の主要な変更ごとにプロジェクトを検証します。

REST API を使用して実行中の KIE Server に、ビジネスアプリケーションの生成されたナレッジ JAR (KJAR) をデプロイできます。REST API の使用方法は、[KIE API を使用した Red Hat Decision Manager の操作](#) を参照してください。

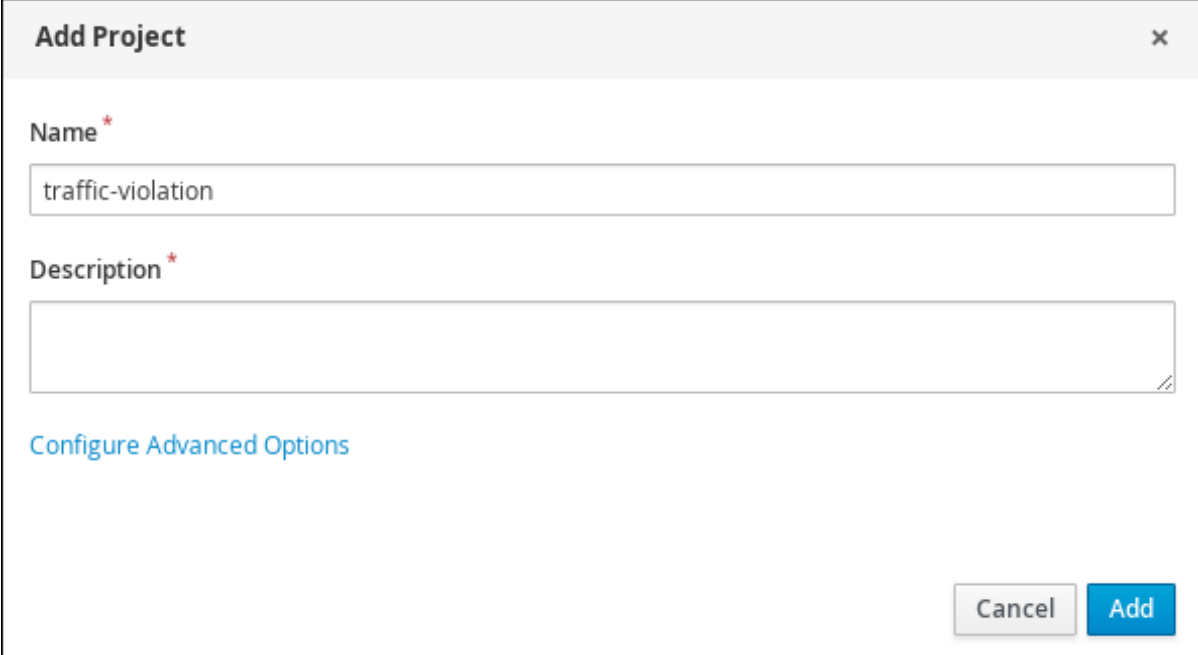
第4章 BUSINESS CENTRAL での交通違反プロジェクトの作成

この例では、**traffic-violation** という名前の新規プロジェクトを作成します。プロジェクトは、データオブジェクト、DMN アセット、およびテストシナリオなどのアセットのコンテナです。作成中のプロジェクト例は、Business Central に含まれる既存の **Traffic_Violation** サンプルプロジェクトに似ています。

手順

1. Business Central で、**Menu** → **Design** → **Projects** に移動します。
Red Hat Decision Manager は **MySpace** と呼ばれるデフォルトスペースを提供します。このデフォルトスペースを使用してサンプルプロジェクトを作成およびテストできます。
2. **Add Project** をクリックします。
3. **Name** フィールドに **traffic-violation** と入力します。
4. **Add** をクリックします。

図4.1 Add Project ウィンドウ



The screenshot shows a dialog box titled "Add Project" with a close button (X) in the top right corner. Below the title bar, there are two input fields: "Name" and "Description". The "Name" field contains the text "traffic-violation". The "Description" field is empty. Below the "Description" field, there is a link labeled "Configure Advanced Options". At the bottom right of the dialog, there are two buttons: "Cancel" and "Add".

プロジェクトの **Assets** ビューを開きます。

第5章 DMN (DECISION MODEL AND NOTATION)

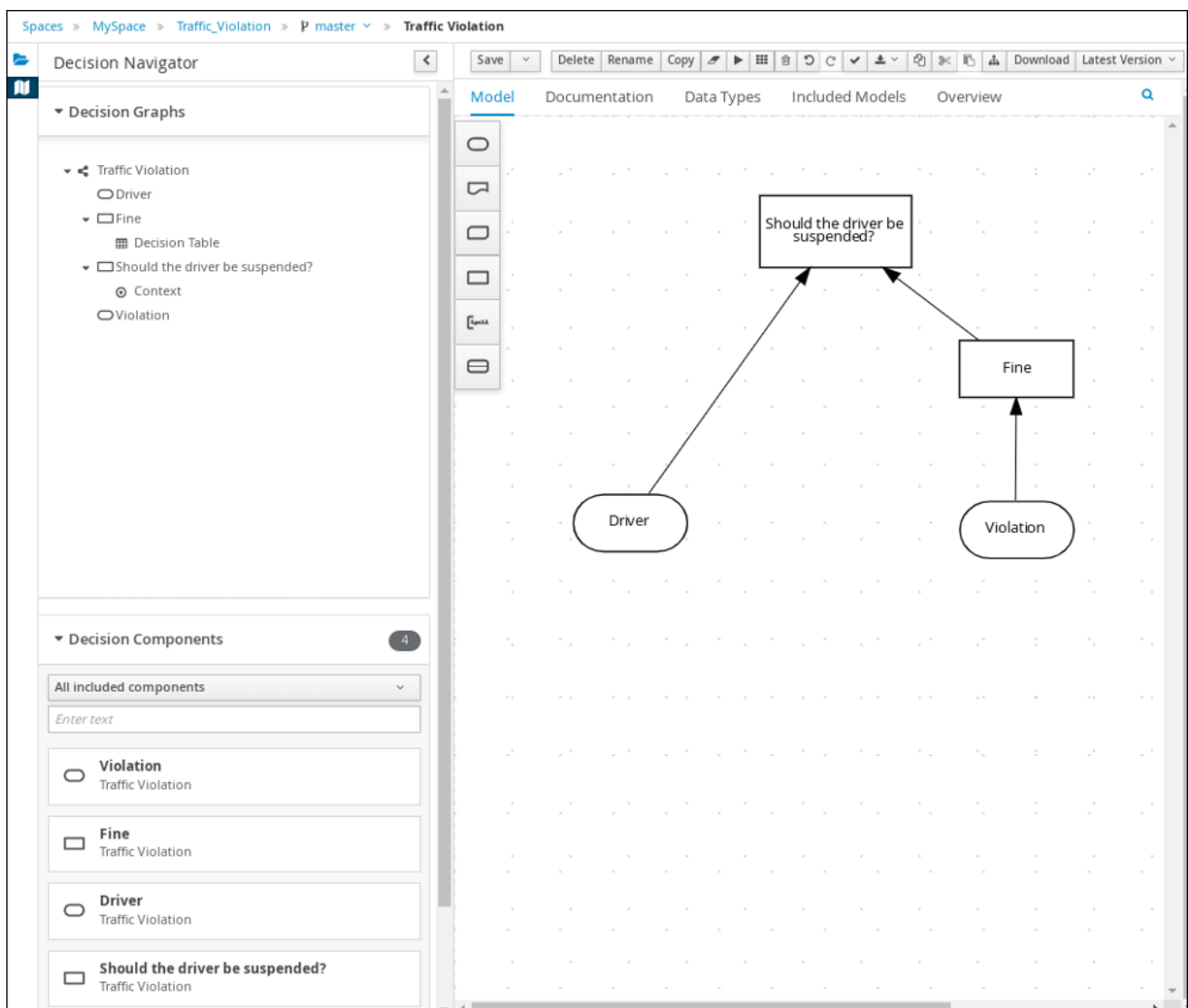
DMN (Decision Model and Notation) は、業務的意思決定を説明してモデル化するために、OMG (Object Management Group) が確立している規格です。DMN は XML スキーマを定義して、DMN モデルを DMN 準拠のプラットフォーム間や組織間で共有し、ビジネスアナリストやビジネスルール開発者が DMN デシジョンサービスの設計と実装で協力できるようにするものです。DMN 規格は、ビジネスプロセスを開発してモデル化する BPMN (Business Process Model and Notation) 規格と類似しており、一緒に使用できます。

DMN の背景およびアプリケーションの詳細は、OMG の [Decision Model and Notation specification](#) を参照してください。

5.1. 交通違反 DMN デシジョン要件ダイアグラム (DRD) の作成

デシジョン要件ダイアグラム (DRD) は、DMN モデルを視覚的にしたものです。Business Central の DMN デザイナーを使用して交通違反プロジェクトの DRD を設計し、DRD コンポーネントのデシジョン論理を定義します。

図5.1 交通違反の例の DRD



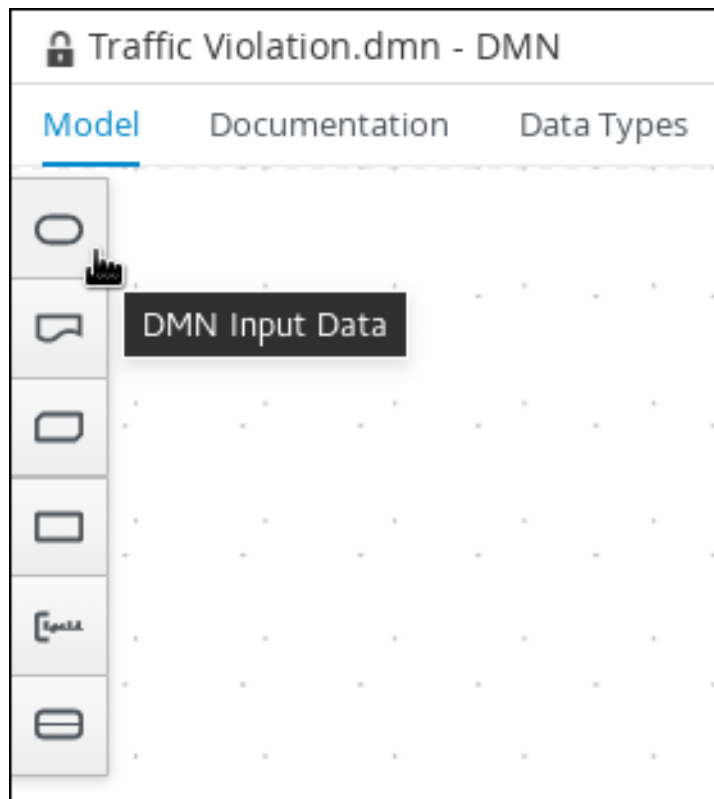
前提条件

- Business Central に交通違反プロジェクトを作成している。

手順

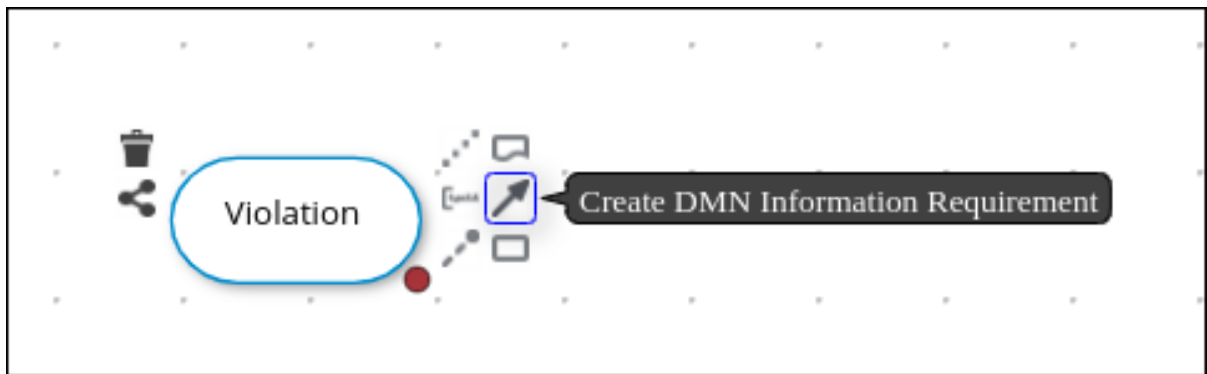
1. traffic-violation プロジェクトのホーム画面で **Add Asset** をクリックします。
2. **Add Asset** ページで **DMN** をクリックします。 **Create new DMN** ウィンドウが開きます。
 - a. **Create new DMN** ウィンドウの **DMN 名** フィールドで **Traffic Violation** を入力します。
 - b. **Package** リストから **com.myspace.traffic_violation** を選択します。
 - c. **OK** をクリックします。DMN デザイナーでDMN アセットが開きます。
3. DMN デザイナーキャンバスで、**DMN Input Data** の入力ノード2つをキャンバスにドラッグします。

図5.2 DMN 入力データノード



4. 右上隅の  アイコンをクリックします。
5. 入力ノードをダブルクリックして、名前の1つを **Driver** に、もう1つを **Violation** に変更します。
6. **DMN Decision** デシジョンノードをキャンバスにドラッグします。
7. デシジョンノードをダブルクリックして、**Fine** に名前を変更します。
8. **Violation** 入力ノードをクリックして **Create DMN Information Requirement** アイコンを選択し、2つのノードを接続する **Fine** デシジョンノードをクリックします。

図5.3 DMN 情報要件アイコンの作成



9. DMN Decision デシジョンノードをキャンバスにドラッグします。
10. デシジョンノードをダブルクリックして、**Should the driver be suspended?** に名前を変更します。
11. Driver 入力ノードをクリックして **Create DMN Information Requirement** アイコンを選択し、2つのノードを接続する **Should the driver be suspended?** デシジョンノードをクリックします。
12. Fine デシジョンノードをクリックして **Create DMN Information Requirement** アイコンを選択し、**Should the driver be suspended?** デシジョンノードを選択します。
13. **Save** をクリックします。



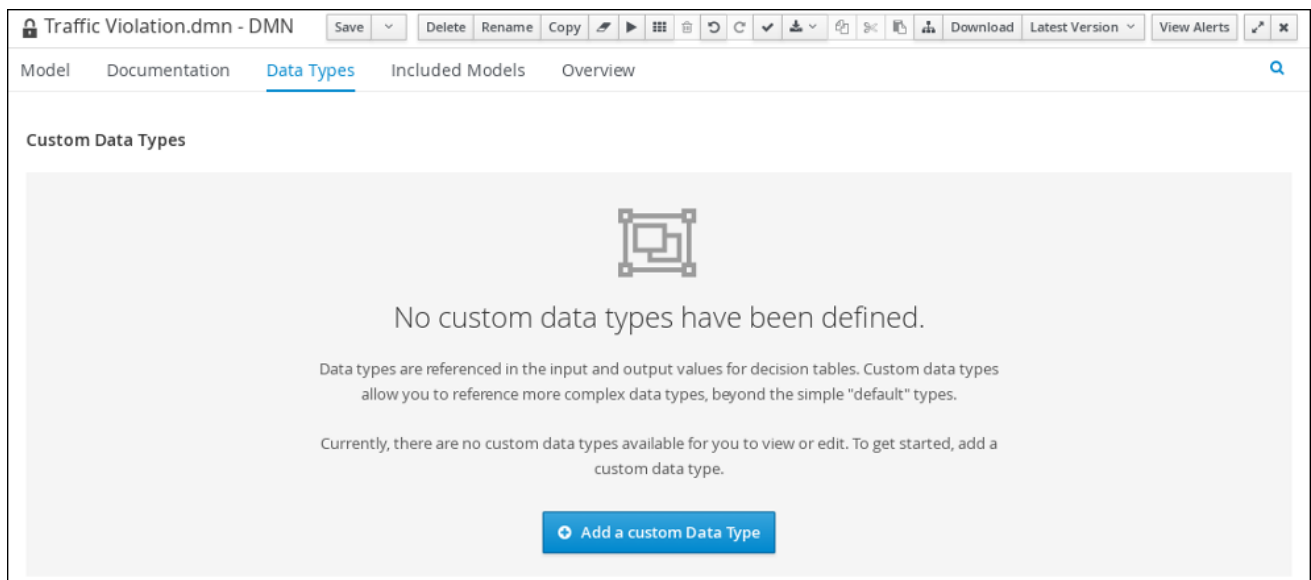
注記

DRD を定期的に保存すると、DMN デザイナーは DMN モデルを静的に検証し、モデルが完全に定義されるまでエラーメッセージを出力する可能性があります。DMN モデルをすべて定義し終えてもエラーが発生する場合は、特定の問題を随時トラブルシューティングしてください。

5.2. 交通違反 DMN カスタムデータタイプの作成

DMN データタイプは、デシジョン論理の定義向けの DMN ボックス式のテーブル、列、フィールドで使用するデータ構造を決定します。デフォルトの DMN データタイプ (文字列、数字、ブール値など) を使用するか、独自のデータタイプを作成して、ボックス式の値に実装する新たなフィールドや制限を指定することもできます。Business Central の DMN デザイナーの **Data Types** タブを使用して交通違反プロジェクトのカスタムデータタイプを定義します。

図5.4 カスタムデータタイプのタブ



以下のテーブルでは、このプロジェクト用に作成する **tDriver**、**tViolation**、および **tFine** のカスタムデータタイプをリスト表示しています。

表5.1 tDriver カスタムデータタイプ

名前	タイプ
tDriver	Structure
名前	string
Age	number
State	string
City	string
Points	number

表5.2 tViolation カスタムデータタイプ

名前	タイプ
tViolation	Structure
Code	string
Date	date
タイプ	string
Speed Limit	number

名前	タイプ
Actual Speed	number

表5.3 tFine カスタムデータタイプ

名前	タイプ
tFine	Structure
Amount	number
Points	number

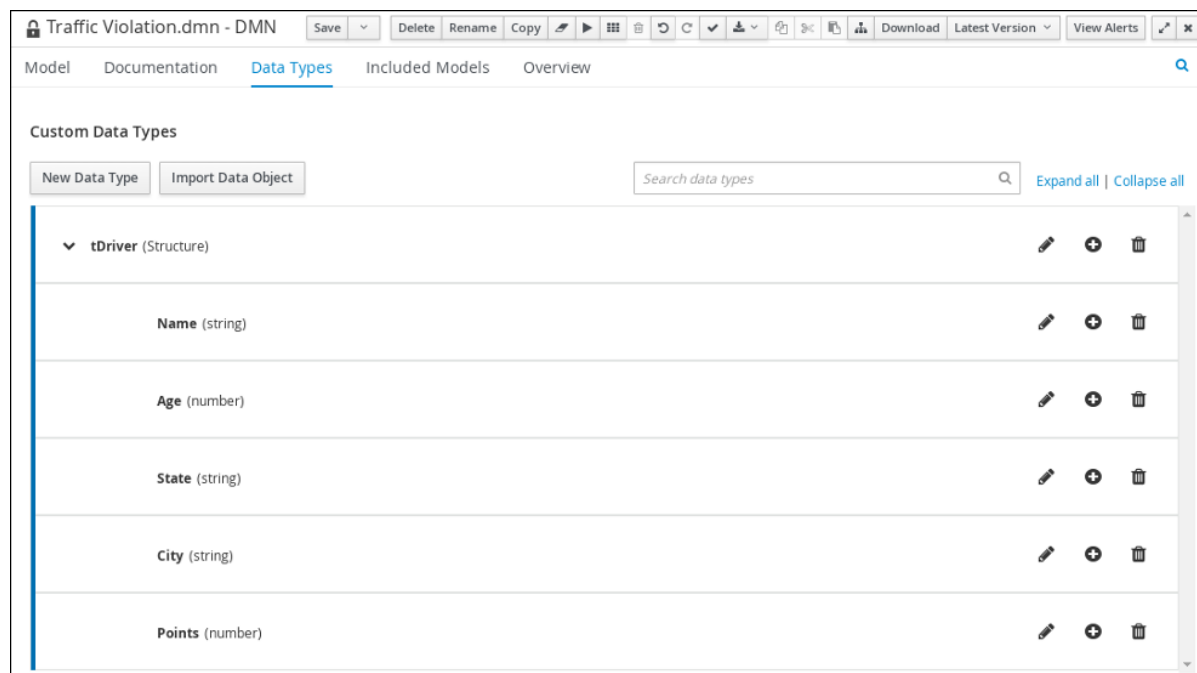
前提条件

- 交通違反 DMN デシジョン要件ダイアグラム (DRD) を Business Central で作成している。

手順

1. **tDriver** カスタムデータタイプを作成するには、**Data Types** タブの **Add a custom Data Type** をクリックし、**Name** フィールドに **tDriver** と入力して、**Type** リストから **Structure** を選択します。
2. 新しいデータタイプの右側にあるチェックマークをクリックして、変更を保存します。

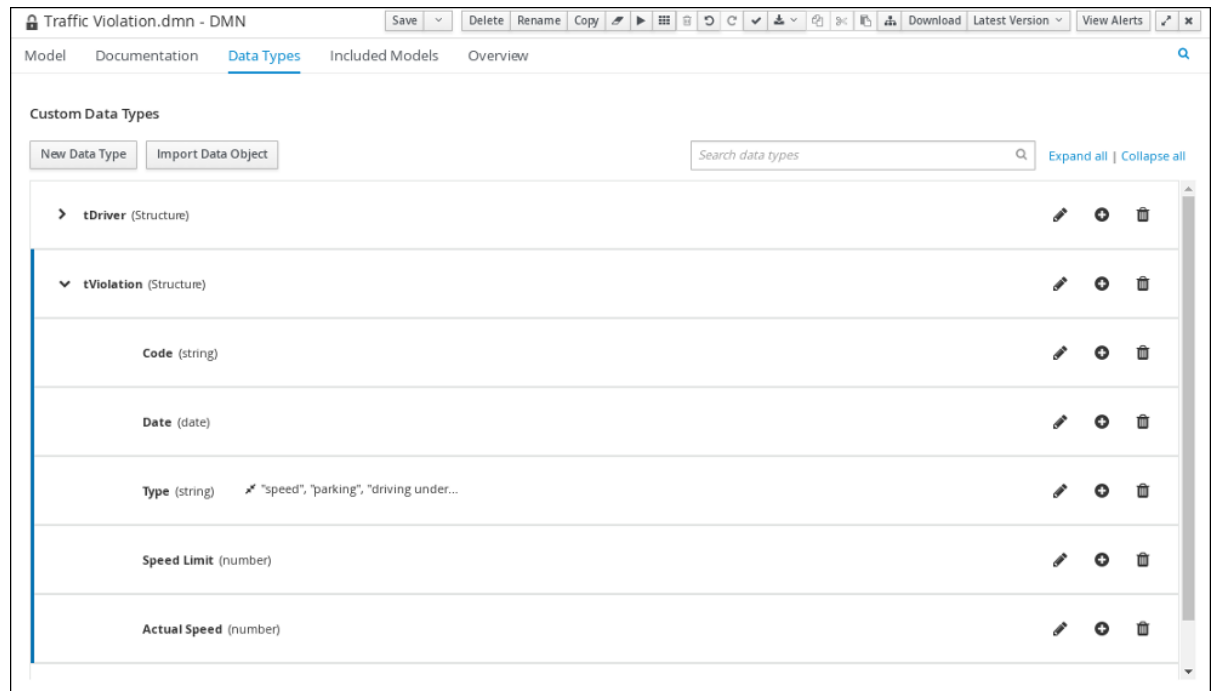
図5.5 tDriver のカスタムデータタイプ



3. 新しくネスト化されたデータタイプごとに、**tDriver** の横にあるプラス記号をクリックして、**tDriver** の構造化データタイプに、以下のネスト化されたデータタイプを追加します。新規データタイプの右側にあるチェックマークをクリックして、変更を保存します。
 - **Name** (文字列)

- **Age** (数字)
 - **State** (文字列)
 - **City** (文字列)
 - **Points** (数字)
4. **tViolation** カスタムデータタイプを作成するには、**New Data Types** をクリックし、**Name** フィールドに **tViolation** と入力して、**Type** リストから **Structure** を選択します。
 5. 新しいデータタイプの右側にあるチェックマークをクリックして、変更を保存します。

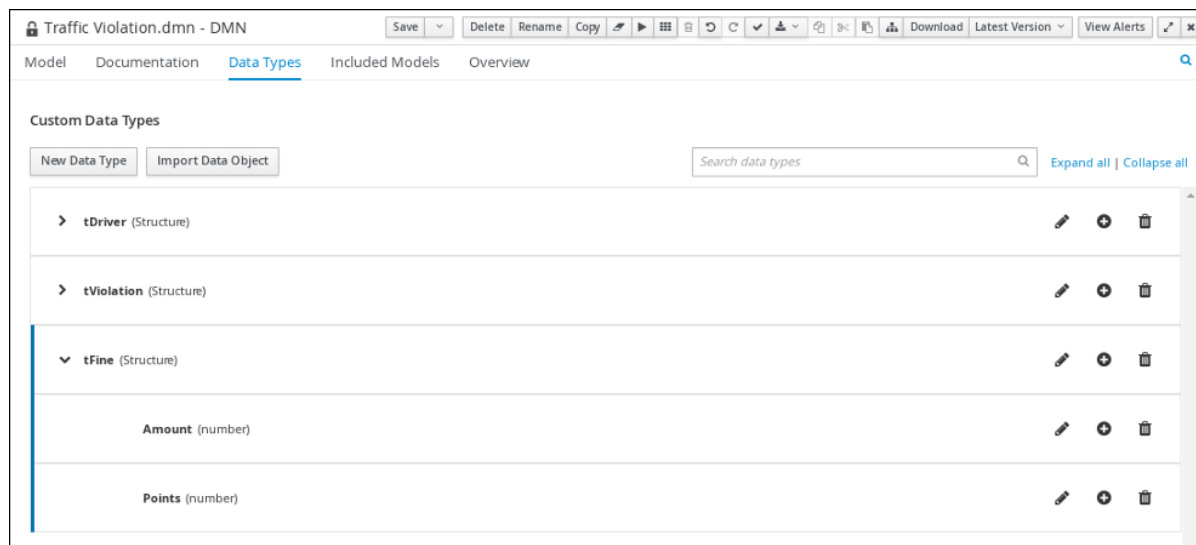
図5.6 tViolation のカスタムデータタイプ



6. 新しくネスト化されたデータタイプごとに、**tViolation** の横にあるプラス記号をクリックして、**tViolation** の構造化データタイプに、以下のネスト化されたデータタイプを追加します。新規データタイプの右側にあるチェックマークをクリックして、変更を保存します。
 - **Code** (文字列)
 - **Date** (日付)
 - **Type** (文字列)
 - **Speed Limit** (数字)
 - **Actual Speed** (数字)
7. 以下の制約を **Type** のネスト化されたデータタイプに追加するには、編集アイコンをクリックして **Add Constraints** をクリックし、**Select constraint type** ドロップダウンメニューから **Enumeration** を選択します。
 - **speed**
 - **parking**
 - **driving under the influence**

8. **OK** をクリックしてから、**type** データタイプの右側にあるチェックマークをクリックし、変更を保存します。
9. **tFine** カスタムデータタイプを作成するには、**New Data Types** をクリックし、**Name** フィールドに **tFine** と入力して、**Type** リストから **Structure** を選択し、**Save** をクリックします。

図5.7 tFine のカスタムデータタイプ



10. 新しくネスト化されたデータタイプごとに、**tFine** の横にあるプラス記号をクリックして、**tFine** の構造化データタイプに、以下のネスト化されたデータタイプを追加します。新規データタイプの右側にあるチェックマークをクリックして、変更を保存します。
 - **Amount** (数字)
 - **Points** (数字)
11. **Save** をクリックします。

5.3. DRD 入力およびデシジョンノードへのカスタムデータタイプの割り当て

DMN カスタムデータタイプを作成したら、それらを交通違反 DRD 内で適切な **DMN Input Data** と **DMN Decision** のノードに割り当てます。

前提条件

- Business Central で交通違反 DMN カスタムデータタイプを作成している。

手順

1. DMN デザイナーで **Model** タブをクリックし、右上の **Properties** アイコンをクリックして DRD プロパティを開きます。
2. DRD で **Driver** 入力データノードを選択し、**Properties** パネルで **Data type** ドロップダウンメニューから **tDriver** を選択します。
3. **Violation** 入力データノードを選択し、**Data type** ドロップダウンメニューから **tViolation** を選択します。

4. Fine デシジョンノードを選択し、Data type ドロップダウンメニューから **tFine** を選択します。
5. Should the driver be suspended? デシジョンノードを選択し、以下のプロパティを設定します。
 - Data type: **string**
 - Question: **Should the driver be suspended due to points on his driver license?**
 - Allowed Answers: **Yes,No**
6. Save をクリックします。

これでカスタムデータタイプが DRD の入力およびデシジョンノードに割り当てられました。

5.4. 交通違反 DMN デシジョン論理の定義

罰金を計算し、ドライバーが免許停止になるかどうかを判定するために、DMN デシジョンテーブルとコンテキストボックス式を使用した交通違反 DMN デシジョン論理を定義します。

図5.8 正規表現

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	

図5.9 ドライバーを免許停止にする式にする必要があります。

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

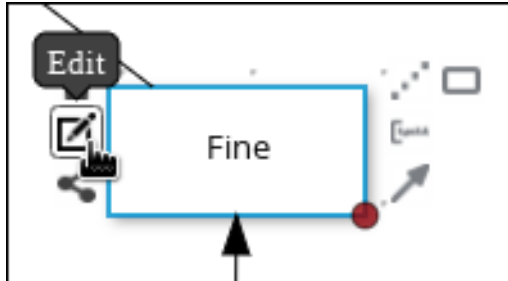
前提条件

- Business Central の交通違反 DRD で、DMN カスタムデータタイプが適切なデシジョンおよび入力ノードに割り当てられている。

手順

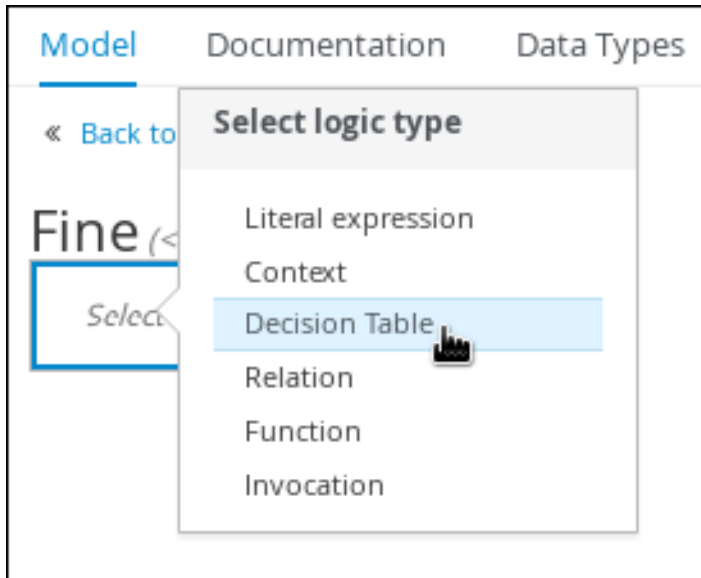
1. 罰金を計算するには、DMN デザイナーキャンバスで **Fine** デシジョンノードを選択し、**Edit** をクリックして DMN ボックス式デザイナーを開きます。

図5.10 デシジョンノードの編集アイコン



2. **Select expression** → **Decision Table** をクリックします。

図5.11 デシジョンテーブルロジックタイプの選択



3. **Violation.Date**、**Violation.Code**、および **Violation.Speed Limit** の列で右クリックして、各フィールドで **Delete** を選択します。
4. **Violation.Actual Speed** 列ヘッダーをクリックし、**Expression** フィールドに **Violation.Actual Speed - Violation.Speed Limit** 式を入力します。
5. デシジョンテーブルの一行目に以下の値を入力します。
 - **Violation.Type: "speed"**
 - **Violation.Actual Speed - Violation.Speed Limit [10..30)**
 - **Amount: 500**
 - **Points: 3**
1行目を右クリックし、**Insert below** を選択して新たな行を追加します。
6. デシジョンテーブルの2行目に以下の値を入力します。

- Violation.Type: "speed"
 - Violation.Actual Speed - Violation.Speed Limit ≥ 30
 - Amount: 1000
 - Points: 7
2行目を右クリックし、**Insert below** を選択して新たな行を追加します。
7. デシジョンテーブルの3行目に以下の値を入力します。
- Violation.Type: "parking"
 - Violation.Actual Speed - Violation.Speed Limit -
 - Amount: 100
 - Points: 1
3行目を右クリックし、**Insert below** を選択して新たな行を追加します。
8. デシジョンテーブルの4行目に以下の値を入力します。
- Violation.Type: "driving under the influence"
 - Violation.Actual Speed - Violation.Speed Limit -
 - Amount: 1000
 - Points: 5
9. **Save** をクリックします。
10. ドライバーの免許停止ルールを定義するにはDMN デザイナーキャンバスに戻って **Should the driver be suspended?** デシジョンノードを選択し、**Edit** をクリックしてDMN ボックス式デザイナーを開きます。
11. **Select expression** → **Context** をクリックします。
12. **ContextEntry-1** をクリックして **Name** に **Total Points** と入力し、**Data Type** ドロップダウンメニューから **number** を選択します。
13. **Total Points** の横のセルをクリックしてコンテキストメニューから **Literal Expression** を選択し、**Driver.Points + Fine.Points** 式を入力します。
14. **Driver.Points + Fine.Points** の下のセルのコンテキストメニューから **Literal Expression** を選択し、**if Total Points ≥ 20 then "Yes" else "No"** と入力します。
15. **Save** をクリックします。
これで罰金の計算方法とドライバーをいつ免許停止にするかを決定するコンテキストが定義されました。**traffic-violation** プロジェクトページに移動して **Build** をクリックすると、用例のプロジェクトがビルドされ、**Alerts** パネルのエラー (ある場合) が対処されます。

第6章 テストシナリオ

Red Hat Decision Manager のテストシナリオでは、ビジネスルールを実稼働環境にデプロイする前に、ビジネスルールとビジネスルールデータの機能 (ルールベースのテストシナリオの場合)、および DMN モデルの機能 (DMN ベースのテストシナリオの場合) を検証できます。このテストシナリオでは、プロジェクトのデータを使用して、指定した条件と、定義した1つ以上のビジネスルールで想定される結果を設定できます。シナリオを実行する際は、想定した結果と、ルールのインスタンスから実際に得られた結果を比較します。想定される結果が実際の結果と一致すると、テストは成功します。想定された結果が実際の結果と一致しないと、テストは失敗します。

Red Hat Decision Manager は現在、新規の **テストシナリオ デザイナー** と以前の **テストシナリオ (レガシー) デザイナー** の両方を含みます。デフォルトのデザイナー、新規のテストシナリオデザイナーで、ルールと DMN モデルのテストをサポートし、テストシナリオの全体的な使用感が改善されています。必要に応じて、レガシーのテストシナリオをそのまま使用することができますが、ルールベースのテストシナリオしかサポートされません。



重要

レガシーのテストシナリオデザイナーは、Red Hat Decision Manager バージョン 7.3.0 で非推奨になりました。今後の Red Hat Decision Manager リリースで削除される予定です。代わりに、新しいテストシナリオデザイナーを使用してください。

プロジェクトレベルや、特定のシナリオアセット内で利用可能なテストシナリオを実行するなど、複数の方法で定義済みのテストシナリオを実行できます。テストシナリオは独立しており、他のテストシナリオに影響を与えたり、テストシナリオを変更したりできません。テストシナリオは、Business Central のプロジェクト開発時にいつでも実行できます。テストシナリオを実行するために、デシジョンサービスをコンパイルまたはデプロイする必要はありません。

別のパッケージからのデータオブジェクトは、テストシナリオと同じプロジェクトパッケージにインポートできます。同じパッケージに含まれるアセットはデフォルトでインポートされます。必要なデータオブジェクトとテストシナリオを作成したら、テストシナリオデザイナーの **Data Objects** タブを使用して、必要なデータオブジェクトがすべてリストされていることを検証するか、**アイテムを追加** して既存のデータオブジェクトをインポートします。



重要

テストシナリオのドキュメント全体で、**テストシナリオ** および **テストシナリオデザイナー** に関する言及はすべて、レガシーバージョンと明示的に記載がない限り、新規バージョンを対象としています。

6.1. テストシナリオを使用した交通違反のテスト

Business Central のテストシナリオデザイナーを使用して DMN 意思決定要件図 (DRD) をテストし、交通違反プロジェクトのデシジョン論理を定義します。

図6.1 交通違反の例のテストシナリオ

Violation Scenarios.scesim - Test Scenarios								
#	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"	<i>Insert value</i>	<i>Insert value</i>	1	100	"No"
4	DUI violation	10	"driving under the influence"	<i>Insert value</i>	<i>Insert value</i>	5	1000	"No"
5	Driver suspended	15	"speed"	100	140	7	1000	"Yes"

前提条件

- Business Central で交通違反プロジェクトを作成している。

手順

1. **traffic-violation** プロジェクトのホーム画面で **Add Asset** をクリックして **Add Asset** 画面を開きます。
2. **Test Scenario** をクリックして **Create new Test Scenario** ダイアログを開きます。
 - a. **Test Scenario** フィールドに **Violation Scenarios** と入力します。
 - b. **Package** リストから **com.myspace.traffic_violation** を選択します。
 - c. **Source type** で **DMN** を選択します。
 - d. **Choose DMN asset** リストから DMN アセットへのパスを選択します。
 - e. **Ok** をクリックして、**Test Scenarios** デザイナーで **Violation Scenarios** テストシナリオを開きます。
3. **Driver** 列サブヘッダー下で、**State**、**City**、**Age**、および **Name** の値のセルを右クリックし、コンテキストメニューから **Delete column** を選択してそれらを削除します。
4. **Violation** 列サブヘッダー下で **Date** と **Code** の値のセルを右クリックし、**Delete column** を選択してそれらを削除します。
5. 以下の情報をテストシナリオの1行目に入力します。
 - **Scenario description: Above speed limit: 10km/h and 30 km/h**
 - **Points (Given 列ヘッダー下): 10**
 - **Type: "speed"**
 - **Speed Limit: 100**
 - **Actual Speed: 120**
 - **Points: 3**
 - **Amount: 500**
 - **Should the driver be suspended?: "No"**
1行目を右クリックし、**Insert row below** を選択して新たな行を追加します。
6. 以下の情報をテストシナリオの2行目に入力します。
 - **Scenario description: Above speed limit: more than 30 km/h**
 - **Points (Given 列ヘッダー下): 10**
 - **Type: "speed"**
 - **Speed Limit: 100**
 - **Actual Speed: 150**

- **Points: 7**
 - **Amount: 1000**
 - **Should the driver be suspended?: "No"**
2行目を右クリックし、**Insert row below**を選択して新たな行を追加します。
7. 以下の情報をテストシナリオの3行目に入力します。
- **Scenario description: Parking violation**
 - **Points (Given 列ヘッダー下): 10**
 - **Type: "parking"**
 - **Speed Limit: 空白のまま**
 - **Actual Speed: 空白のまま**
 - **Points: 1**
 - **Amount: 100**
 - **Should the driver be suspended?: "No"**
3行目を右クリックし、**Insert row below**を選択して新たな行を追加します。
8. 以下の情報をテストシナリオの4行目に入力します。
- **Scenario description: DUI violation**
 - **Points (Given 列ヘッダー下): 10**
 - **Type: "driving under the influence"**
 - **Speed Limit: 空白のまま**
 - **Actual Speed: 空白のまま**
 - **Points: 5**
 - **Amount: 1000**
 - **Should the driver be suspended?: "No"**
4行目を右クリックし、**Insert row below**を選択して新たな行を追加します。
9. 以下の情報をテストシナリオの5行目に入力します。
- **Scenario description: Driver suspended**
 - **Points (Given 列ヘッダー下): 15**
 - **Type: "speed"**
 - **Speed Limit: 100**
 - **Actual Speed: 140**
 - **Points: 7**

- Amount: 1000
- Should the driver be suspended?: "Yes"

10. Save をクリックします。


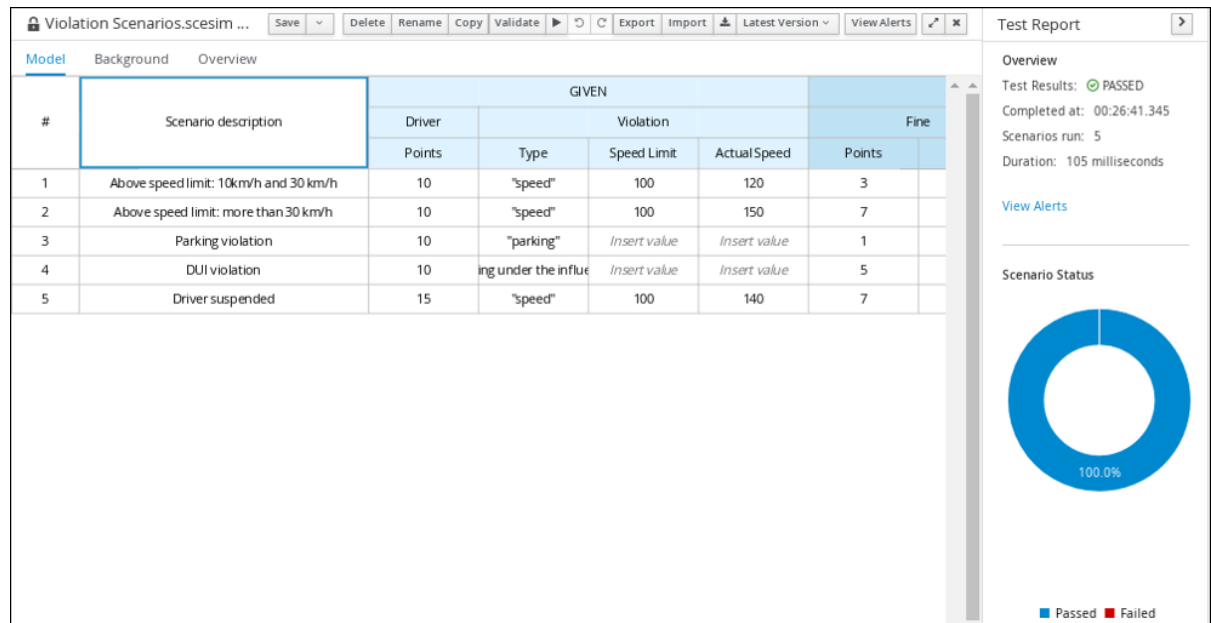
11. Play ボタン  をクリックして、テストシナリオが合格するかどうかを確認します。

図6.2 交通違反の例のテストシナリオ実行結果



失敗した場合は、エラーを修正してサイドテストシナリオを実行します。

第7章 DMN モデルの実行

Business Central を使用して Red Hat Decision Manager のプロジェクトに DMN ファイルをインポートまたは作成するか、Business Central を使用しないプロジェクトのナレッジ JAR (KJAR) ファイルの一部として DMN ファイルをパッケージ化できます。Red Hat Decision Manager プロジェクトに DMN ファイルを実装したら、リモートアクセスの KIE Server にこのファイルを含む KIE コンテナをデプロイして、KIE Server REST API を使用するコンテナと対話することで、DMN デシジョンサービスを実行できます。

プロジェクトのパッケージングおよびデプロイメントの方法に外部 DMN アセットを含める方法は、[Red Hat Decision Manager プロジェクトのパッケージ化およびデプロイ](#) を参照してください。

7.1. KIE SERVER REST API を使用した DMN サービスの実行

KIE Server の REST エンドポイントで直接対話することで、呼び出しコードと、意思決定ロジックの定義の分離が最大になります。呼び出しコードに直接の依存関係がないため、**Node.js**、**.NET** など、完全に異なる開発プラットフォームに実装できます。このセクションの例では、Nix スタイルの curl コマンドを示しますが、REST クライアントに適用するための関連情報を提供します。

KIE Server の REST エンドポイントを使用する場合、標準の KIE Server マーシャリングアノテーションが付けられたドメインオブジェクト POJO Java クラスを定義することが推奨されます。たとえば、以下のコードは、適切にアノテーションが付けられたドメインオブジェクトの **Person** クラスを使用しています。

POJO Java クラスの例

```
@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;
    }
}
}

```

KIE Server REST API の詳細は、[KIE API を使用した Red Hat Decision Manager の操作](#)を参照してください。

前提条件

- KIE Server がインストールされ、設定されている (**kie-server** ロールが割り当てられているユーザーの既知のユーザー名と認証情報を含む)。インストールオプションについては、[Planning a Red Hat Decision Manager installation](#) を参照してください。
- KJAR アーティファクトとして DMN プロジェクトをビルドして、KIE Server にデプロイしておく。

```
mvn clean install
```

プロジェクトのパッケージ化およびデプロイメント、ならびに実行可能モデルに関する詳細は、[Red Hat Decision Manager プロジェクトのパッケージ化およびデプロイ](#)を参照してください。

- KIE コンテナの ID に DMN モデルを含んでいる。1つ以上のモデルが存在する場合は、そのモデルの名前空間およびモデル名が必要です。

手順

1. KIE Server REST API エンドポイントにアクセスするためのベース URL を決定します。これには、以下の値が必要です (例ではローカルデプロイメントのデフォルト値を使用しています)。
 - ホスト (**localhost**)
 - ポート (**8080**)
 - ルートコンテキスト (**kie-server**)
 - ベース REST パス (**services/rest/**)

交通違反プロジェクトでのローカルデプロイメントにおけるベース URL の例:

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

2. ユーザー認証要件を決定します。

ユーザーを KIE Server 設定に直接定義すると、ユーザー名およびパスワードを要求する HTTP Basic 認証が使用されます。要求を成功させるには、ユーザーに **kie-server** ルールが必要です。

以下の例は、curl 要求に認証情報を追加する方法を示します。

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

Red Hat Single Sign-On を使用して KIE Server を設定している場合は、要求にベアラートークンが必要です。

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

3. 要求と応答の形式を指定します。REST API エンドポイントには JSON と XML の両方の書式が利用でき、要求ヘッダーを使用して設定されます。

JSON

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

XML

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

4. 必要に応じて、デプロイしたデシジョンモデルのリストに対するコンテナのクエリーです。
[GET] server/containers/{containerId}/dmn

curl 要求例:

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

サンプルの XML 出力:

```
<?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-typepref>
    <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-typepref>
        <namespace-uri>https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8</namespace-uri>
        <local-part>number</local-part>
        <prefix></prefix>
    </itemdefinition-typepref>
    <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-typepref>
        <namespace-uri>https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8</namespace-uri>
        <local-part>string</local-part>
        <prefix></prefix>
    </itemdefinition-typepref>
    <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-typepref>
        <namespace-uri>https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8</namespace-uri>
        <local-part>string</local-part>
        <prefix></prefix>
    </itemdefinition-typepref>
    <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-typepref>
        <namespace-uri>https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8</namespace-uri>
        <local-part>number</local-part>
        <prefix></prefix>
    </itemdefinition-typepref>
    <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-typerref>
          <namespace-uri>https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8</namespace-uri>
          <local-part>number</local-part>
          <prefix></prefix>
        </itemdefinition-typerref>
        <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-typerref>
      <namespace-uri>https://kiegroup.org/dmn/_60B01F4D-E407-43F7-848E-
258723B5FAC8</namespace-uri>
      <local-part>number</local-part>
      <prefix></prefix>
    </itemdefinition-typerref>
    <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>

```

サンプルの JSON 出力:

```

{
  "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-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-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. モデルを実行します。

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



注記

model-namespace の属性は自動生成され、各ユーザーで異なります。**model-namespace** と **model-name** の属性がデプロイされているモデルのものに合致することを確認してください。

curl 要求例:

```

curl -u wbadm:wbadm -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 要求例:

```

{
  "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 要求例 (JAXB 形式):

```

<?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>

```

注記

要求には、その形式にかかわらず、以下の要素が必要です。

- モデルの名前空間
- モデル名
- 入力値を含むコンテキストオブジェクト

JSON 応答例:

```

{
  "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 形式) 応答例:

```

<?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 関連情報

- [DMN モデルを使用したデシジョンサービスの作成](#)
- [テストシナリオを使用したデシジョンサービスのテスト](#)
- [Business Central でのプロジェクトの管理](#)
- [KIE API を使用した Red Hat Decision Manager の操作](#)

パート II. RED HAT ビルドの OPTAPLANNER のスタートガイド

ビジネスルールの開発者は、Red Hat ビルドの OptaPlanner を使用して、限られたリソースや個別の制約の中で計画問題に対する最適解を見つけ出すことができます。

本書を使用して、Red Hat ビルドの OptaPlanner で Solver の開発を開始していきます。

第9章 RED HAT ビルドの OPTAPLANNER の概要

OptaPlanner は組み込み可能な軽量プランニングエンジンで、プランニングの問題を最適化します。最適化のためのヒューリスティック法およびメタヒューリスティック法を、非常に効率的なスコア計算と組み合わせ、一般的な Java プログラマーが計画問題を効率的に解決できるようにします。

たとえば、OptaPlanner は、さまざまなユースケースの解決に役立ちます。

- **従業員勤務表/患者リスト:** 看護師の勤務シフト作成を容易にし、病床管理を追跡します。
- **教育機関の時間割:** 授業、コース、試験、および会議の計画を容易にします。
- **工場の計画:** 自動車の組み立てライン、機械の操業計画、および作業員のタスク計画を追跡します。
- **Inventoryの削減:** 紙や金属などの資源の消費を減らし、無駄を最小限に抑えます。

どの組織も、制約のある限定されたリソース (従業員、資産、時間、および資金) を使用して製品やサービスを提供するといった計画問題に直面しています。

OptaPlanner は、Apache Software License 2.0 を使用するオープンソースソフトウェアです。100% Pure Java に認定されており、ほとんどの Java 仮想マシン (JVM) で稼働します。

9.1. 計画問題

計画問題 では、限られたリソースや個別の制約の中で最適なゴールを見つけ出します。最適なゴールは、次のようなさまざまなものです。

- **最大の利益:** 最適なゴールにより、可能な限り高い利益が得られます。
- **経済活動の最小フットプリント:** 最適なゴールでは、環境負荷が最小となります。
- **スタッフ/顧客の最大満足:** 最適なゴールでは、スタッフ/顧客のニーズが優先されます。

これらのゴールに到達できるかどうかは、利用できるリソースの数に依存します。たとえば、以下のようリソースには制限があります。

- ユーザー数
- 時間
- 予算
- 装置、車両、コンピューター、施設などの物理資産

これらのリソースに関連する個別の制約についても考慮する必要があります。たとえば、要員が働くことのできる時間数、特定の装置を使用することのできる技能、または機器同士の互換性などです。

Red Hat ビルドの OptaPlanner は、Java プログラマーが制約の飽和性の問題を効率的に解決するのに役立ちます。最適化ヒューリスティックとメタヒューリスティックを効率的なスコア計算と組み合わせます。

9.2. 計画問題での NP 完全

例に挙げたユースケースは **通常 NP 完全**または **NP 困難** であり、以下のことが言えます。

- 問題に対する解を実用的な時間内に検証することが容易です。
- 問題に対する最適解を実用的な時間内に見つけ出す確実な方法がない。

この場合、一般的な2つの手法では不十分であるため、問題を解くのが予想より困難だと考えられます。

- かまかせアルゴリズムでは(より高度な類似アルゴリズムであっても)、時間がかかり過ぎる。
- たとえば [ビンパッキング問題](#) のような迅速なアルゴリズムでは、**容量の大きい順でアイテムを入力すると、最適とはほど遠い解が返されます。**

高度な最適化アルゴリズムを用いる OptaPlanner であれば、このような計画問題に対する適切な解を、妥当な時間内に見つけ出すことができます。

9.3. 計画問題に対する解

計画問題には、多数の解が存在します。

以下に示すように、解は複数のカテゴリーに分類されます。

可能解

可能解とは、制約に違反するかどうかは問わず、あらゆる解を指します。通常、計画問題には膨大な数の可能解が存在します。ただし、このような解の多くは、役に立ちません。

実行可能解

実行可能解とは、いずれの(負の)ハード制約にも違反しない解を指します。実行可能解の数は、可能解の数に比例します。実行可能解が存在しないケースもあります。実行可能解は、可能解の部分集合です。

最適解

最適解とは、最高スコアの解を指します。通常、計画問題には数個の最適解が存在します。実行可能解が存在せず、最適解が現実的ではない場合でも、計画問題には少なくとも1つの最適解が必ず存在します。

見つかった最善解

最善解とは、指定された時間内に実施した検索で見つかった最高スコアの解を指します。通常、見つかった最善解は現実的で、十分な時間があれば最適解を見つけることができます。

直観には反していますが、小規模なデータセットの場合であっても、(正しく計算された場合は)膨大な数の可能解が存在します。

planner-engine ディストリビューションディレクトリーのサンプルでも、ほとんどのインスタンスには多数の可能解が存在します。最適解を確実に見つけることができる方法は存在しないため、いかなる実行方法も、これらすべての可能解の部分集合を評価することしかできません。

膨大な数の可能解全体を効率的に網羅するために、OptaPlanner はさまざまな最適化アルゴリズムをサポートしています。

ユースケースによっては、ある最適化アルゴリズムが他のアルゴリズムより勝ることがありますが、それを事前に予測することは不可能です。OptaPlanner では、XML またはコード中の Solver 設定を数行変更するだけで、最適化アルゴリズムを切り替えることができます。

9.4. 計画問題に対する制約

通常、プランニングの問題には、少なくとも2つの制約レベルがあります。

- **(負の)ハード制約** は、絶対に違反してはならない。
例: 1人の教師は同時に2つの講義を受け持つことはできない。
- **(負の)ソフト制約** は、避けることが可能であれば違反してはならない。
例: 教師 A は金曜日の午後に講義を受け持ちたくない。

正の制約を持つ問題もあります。

- **正のソフト制約 (ボーナス)** は、可能であれば満たす必要がある。
例: 教師 B は月曜日の午前中に講義を受け持つことを希望している。

一部の基本的な問題にはハード制約のみがあります。問題によっては、3つ以上の制約があります (例: ハード制約、中程度の制約、ソフト制約)。

これらの制約により、計画問題における **スコア計算方法** (または **適合度関数**) が定義されます。プランニングの問題の解は、それぞれスコアで等級付けすることができます。OptaPlanner のスコア制約は、Java などのオブジェクト指向言語または Drools ルールで記述されます。

このタイプのコードは柔軟で、スケーラビリティに優れます。

9.5. RED HAT ビルドの OPTAPLANNER で提供される例

Red Hat Decision Manager には、Red Hat ビルドの OptaPlanner のサンプルが複数同梱されています。たとえばコードなどを確認して、ニーズに合ったものに変更できます。



注記

Red Hat は、Red Hat Decision Manager ディストリビューションに含まれるコードサンプルのサポートはしていません。

OptaPlanner サンプルには、教育関連のコンテストで出題された問題を解決するものもあります。以下の表の **Contest** 列には、このようなコンテストが掲載されています。また、コンテストの目的として、**現実的** か、**非現実的** かの識別をしています。**現実的なコンテスト** とは、以下の基準を満たす、独立した公式コンテストを指します。

- 明確に定義された実際のユースケースであること
- 実際に制約があること
- 実際のデータセットが複数あること
- 特定のハードウェアで特定の時間内に結果を再現できること
- 教育機関および/または企業の運用研究コミュニティが真剣に参加していること

現実的なコンテストでは、競合のソフトウェアや教育研究と OptaPlanner を客観的に比較できます。

表9.1 サンプルの概要

例	ドメイン	サイズ	コンテスト	ディレクトリー名
			ト	

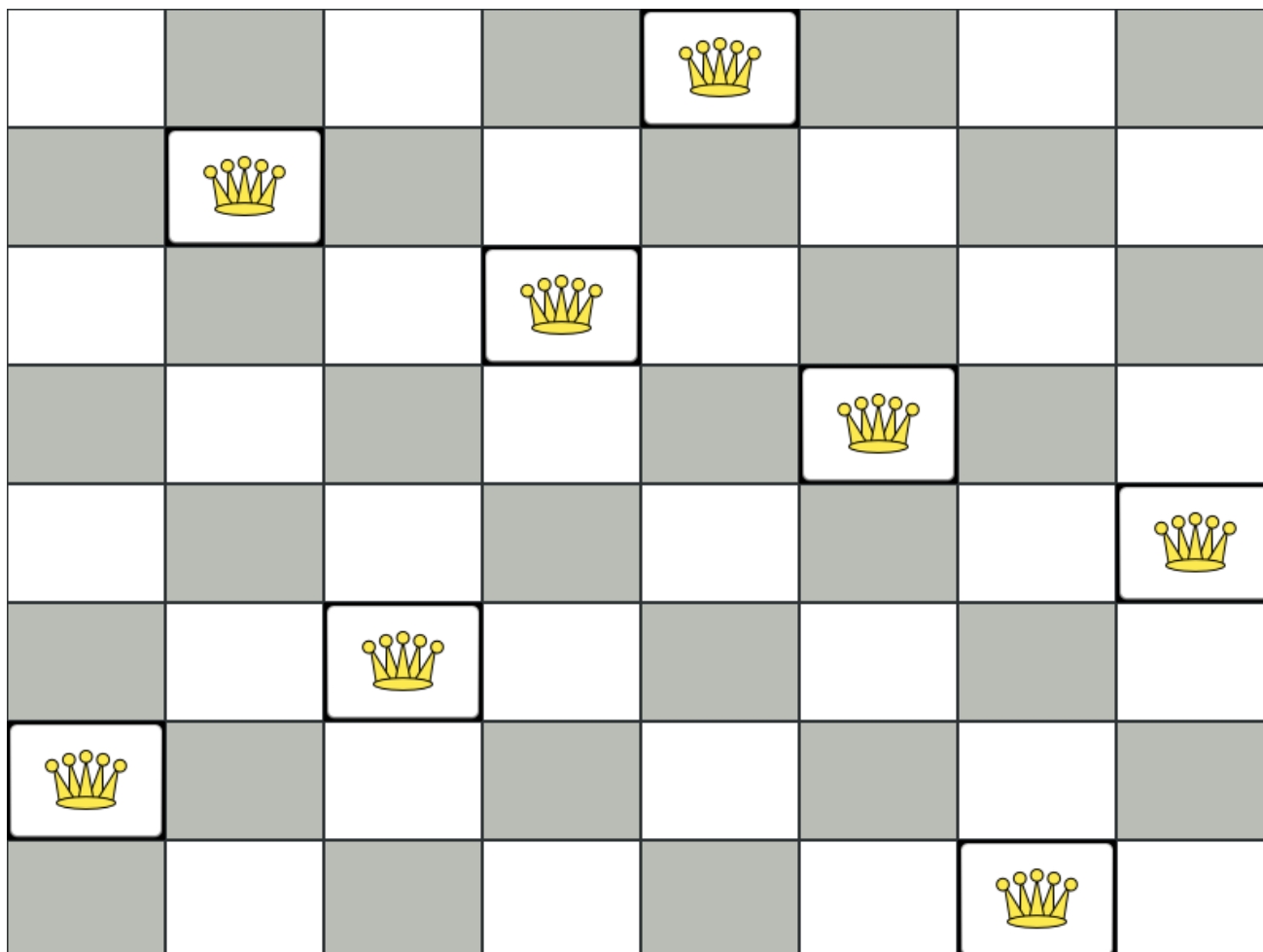
例	ドメイン	サイズ	コンテ スト	ディレクトリー名
Nクイーン	エンティティークラス1つ (変数1つ)	エンティティークラス1つ 値 \leftarrow 256 探索空間 \leftarrow 10^{616}	無意味 (不正が可能)	nqueens
クラウド バラン シング	エンティティークラス1つ (変数1つ)	エンティティークラス1つ 値 \leftarrow 2400 探索空間 \leftarrow 10^{6967}	いいえ (弊社が定義)	cloudbalancing
巡回セ ールスマ ン	エンティティークラス1つ (連鎖変数1つ)	エンティティークラス1つ 値 \leftarrow 980 探索空間 \leftarrow 10^{2504}	現実的で ない TSP Web	tsp
テニス クラブの スケジ ュー ル	エンティティークラス1つ (変数1つ)	エンティティークラス1つ 値 \leftarrow 72 探索空間 \leftarrow 10^{60}	いいえ (弊社が定義)	tennis
会議の スケジ ュー ル	エンティティークラス1つ (変数2つ)	エンティティークラス1つ 値 \leftarrow 10 値 \leftarrow 320 および \leftarrow 5 探索空間 \leftarrow 10^{320}	いいえ (弊社が定義)	meetingscheduling
コースの 時間割	エンティティークラス1つ (変数2つ)	エンティティークラス1つ 値 \leftarrow 434 値 \leftarrow 25 および \leftarrow 20 探索空間 \leftarrow 10^{1171}	現実的 ITC 2007 track 3	curriculumCourse
マシンの 再割当	エンティティークラス1つ (変数1つ)	エンティティークラス1つ 値 \leftarrow 50000 値 \leftarrow 5000 探索空間 \leftarrow 10^{184948}	ほぼ現実 的 ROADEF 2012	machineReassignment

例	ドメイン	サイズ	コンテ スト	ディレクトリー名
配送経路	エンティティークラス1つ (連鎖変数1つ) シャドウエンティティークラス1つ (自動シャドウ変数1つ)	エンティティークラス1つ 値 \leftarrow 2740 値 \leftarrow 2795 探索空間 \leftarrow 10^{8380}	現実的でない VRP Web	vehiclerouting
時間枠がある中で の 配送経路	配送経路すべて (シャドウ変数1つ)	エンティティークラス1つ 値 \leftarrow 2740 値 \leftarrow 2795 探索空間 \leftarrow 10^{8380}	現実的でない VRP Web	vehiclerouting
プロジェクトジョブのスケジュール	エンティティークラス1つ (変数2つ) (シャドウ変数1つ)	エンティティークラス1つ 値 \leftarrow 640 値 \leftarrow ? および \leftarrow ? 探索空間 \leftarrow ?	ほぼ現実的 MISTA 2013	projectjobscheduling
タスクの割り当て	エンティティークラス1つ (連鎖変数1つ) (シャドウ変数1つ) シャドウエンティティークラス1つ (自動シャドウ変数1つ)	エンティティークラス1つ 値 \leftarrow 500 値 \leftarrow 520 探索空間 \leftarrow 10^{1168}	いいえ (弊社が定義)	taskassigning
試験の時間割	エンティティークラス2つ (同じ階層) (変数2つ)	エンティティークラス2つ 値 \leftarrow 1096 値 \leftarrow 80 および \leftarrow 49 探索空間 \leftarrow 10^{3374}	現実的 ITC 2007 track 1	examination
看護師の勤務表	エンティティークラス1つ (変数1つ)	エンティティークラス1つ 値 \leftarrow 752 値 \leftarrow 50 探索空間 \leftarrow 10^{1277}	現実的 INRC 2010	nurserostering

例	ドメイン	サイズ	コンテ スト	ディレクトリー名
巡回トー ナメント	エンティティークラス1 つ (変数1つ)	エンティティークラス1つ 値 \Leftarrow 1560 値 \Leftarrow 78 探索空間 \Leftarrow 10²³⁰¹	現実的で ない TTP	travelingtournament
コストを 抑えるス ケジュー ル	エンティティークラス1 つ (変数2つ)	エンティティークラス1つ 値 \Leftarrow 500 値 \Leftarrow 100 および \Leftarrow 288 探索空間 \Leftarrow 10²⁰⁰⁷⁸	ほぼ現実 的 ICON Energy	cheaptimeschedulin g
投資	エンティティークラス1 つ (変数1つ)	エンティティークラス1つ 値 = 1000 探索空間 \Leftarrow 10⁴	いいえ (弊 社が定義)	investment
会議スケ ジュール	エンティティークラス1 つ (変数2つ)	エンティティークラス1つ 値 \Leftarrow 216 値 \Leftarrow 18 および \Leftarrow 20 探索空間 \Leftarrow 10⁵⁵²	いいえ (弊 社が定義)	conferencescheduli ng
ロックツ アー	エンティティークラス1 つ (連鎖変数1つ) (シャドウ変数4つ) シャドウエンティ ティークラス1つ (自動シャドウ変数1つ)	エンティティークラス1つ 値 \Leftarrow 47 値 \Leftarrow 48 探索空間 \Leftarrow 10⁵⁹	いいえ (弊 社が定義)	rocktour
航空機乗 組員のス ケジュー リング	エンティティークラス1 つ (変数1つ) シャドウエンティ ティークラス1つ (自動シャドウ変数1つ)	エンティティークラス1つ 値 \Leftarrow 4375 値 \Leftarrow 750 探索空間 \Leftarrow 10¹²⁵⁷⁸	いいえ (弊 社が定義)	flightcrewschedulin g

9.6. N キーン

n サイズのチェスボードに、他のキーンに取られないキーンに n 個のキーンを置きます。最も一般的な n キーンパズルは、 $n = 8$ の 8 個のキーンパズルです。



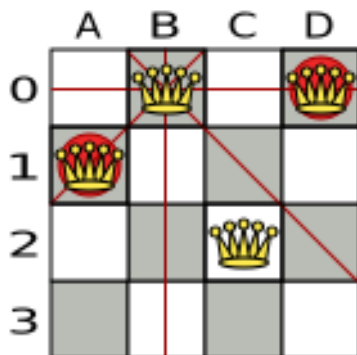
制約:

- n 列および n 行のチェスボードを使用します。
- チェスボードに n 個のクィーンを置きます。
- クィーンが他のクィーンに取られないように配置します。クィーンは、同じ水平線上、垂直線上、対角線上にある他のクィーンを取ることができます。

本書では、4つのクィーンパズルを主な例として多用しています。

以下が提案された解です。

図9.14 個のクィーンパズルの誤った解



上記の解は、**A1** と **B0** (および **B0** と **D0**) のクイーンがお互いに駒を取れるので間違っています。**B0** のクイーンをどこに置けば他のクイーンに取られないようにするという制約は順守できますが、**n** 個のクイーンを置くという制約に違反します。

以下は正しい解です。

図9.2 クイーン 4 個のパズルの正しい解

	A	B	C	D
0			♔	
1	♔			
2				♔
3		♔		

すべての制約が満たされているので、これが正解です。

n クイーンパズルでは、正解が複数存在する場合があります。特定の **n** に対して考えられる解を見つけるのではなく、特定の **n** に対する正しい解を1つ導き出すことにフォーカスします。

問題の規模

```
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 クイーンは、初心者用のサンプルとして実装されているため、最適化はされていません。それにもかかわらず、クイーンが64個になっても簡単に処理できます。何点か変更を加えると、クイーンが5000個以上になっても簡単に対応できることが立証されています。

9.6.1. N クイーンのドメインモデル

この例では、4つのクイーンの問題を解決するドメインモデルを使用します。

- **ドメインモデルの作成**

適切なドメインモデルを使用すると、プランニングの問題をより簡単に理解し、解決することができます。

以下は、**n** クイーンの例のドメインモデルです。

```
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
}
```

- 探索空間の計算

Queen インスタンスには **Column** (例: 0 は列 A、1 は列 B) および **Row** (例: 0 は行 0、1 は行 1) が含まれます。

列と行をもとに、昇順の対角線、および降順の対角線を計算することができます。

列と行のインデックスは、チェスボードの左上隅から数えています。

```
public class NQueens {
    private int n;
    private List<Column> columnList;
    private List<Row> rowList;

    private List<Queen> queenList;

    private SimpleScore score;

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

- 解の求め方

1つの **NQueens** インスタンスには **Queen** インスタンスのリストが含まれています。これが **Solution** 実装として提供され、Solver が解決して読み出します。

たとえば、4 クイーンのサンプルでは、NQueens の **getN()** メソッドが常に 4 を返します。

図9.3 キーン 4 個の解

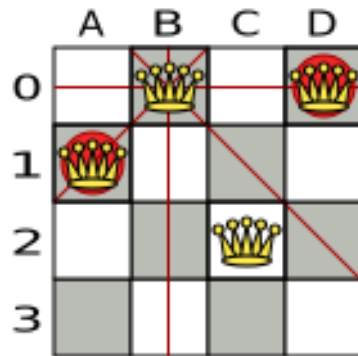


表9.2 ドメインモデルの解の詳細

	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

(*) や (**) のように、キーン 2 つが同じ行、列、対角線を共有する場合は、2 つの駒が互いを取ることができます。

9.7. クラウドバランシング

この例に関する詳細は、[Red Hat ビルドの OptaPlanner クイックスタートガイド](#) を参照してください。

9.8. 巡回セールスマン (TSP - 巡回セールスマン問題)

都市のリストをもとに、セールスマンが最短距離で、各都市を 1 度だけ訪問するルートを探します。

この問題は [ウィキペディア](#) に定義されています。これは、計算数学で [最も熱心に研究された問題の 1 つ](#) です。大概は、従業員のシフト勤務など、その他の制約と一緒に計画の問題の一部として使用されます。

問題の規模

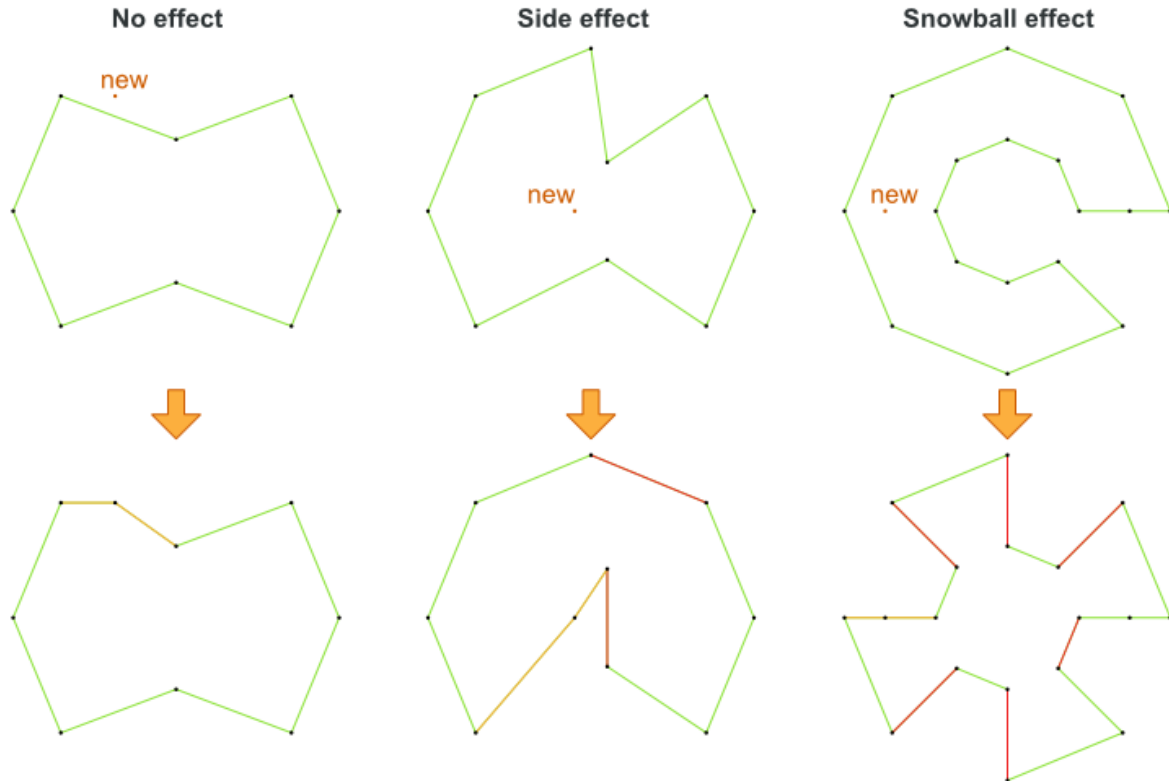
dj38 has 38 cities with a search space of 10^{43} .
 europe40 has 40 cities with a search space of 10^{46} .
 st70 has 70 cities with a search space of 10^{98} .
 pcb442 has 442 cities with a search space of 10^{976} .
 lu980 has 980 cities with a search space of 10^{2504} .

問題の難易度

TSP の定義は単純ですが、問題の解決は驚くほど難しくなります。これは NP 困難問題と呼ばれ、多くの計画の問題と同様、特定の問題のデータセットに対する最適な解は、その問題のデータセットが少しでも変更すると、大幅に変化する可能性があります。

TSP optimal solution volatility

How much does the optimal solution change if we add 1 new location?



9.9. テニスクラブのスケジュール

テニスクラブでは、毎週 4 チームが総あたりで試合をします。4 つの対戦枠を公平にチームに割り当てます。

ハード制約:

- 競合: チームは 1 日に 1 回だけ試合ができる。
- 参加不可: 日程によって参加できないチームがある。

中程度の制約:

- 公平な割り当て: 各チームが試合をする回数を (ほぼ) 同じにする。

ソフト制約:

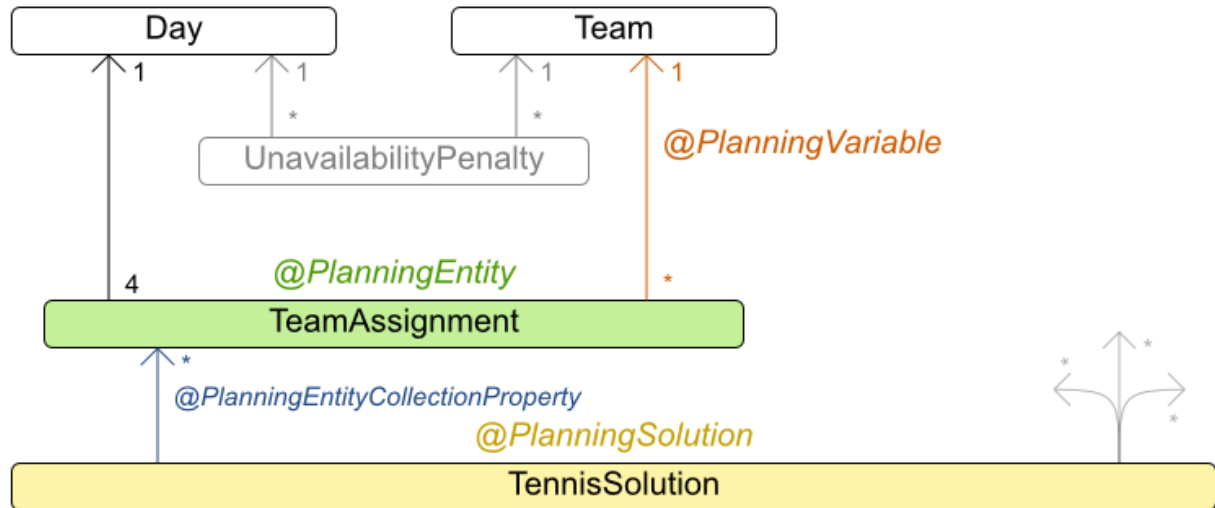
- 均等に対戦: 各チームが、各対戦相手と対戦する回数を同じにする。

問題の規模

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

図9.4 ドメインモデル

Tennis class diagram



9.10. 会議のスケジュール

各会議に、開始時間と会議室を割り当てます。会議の長さは異なります。

ハード制約:

- 部屋の制約: 2つの会議が、同じ時間に同じ会議室を使用することはできない。
- 必須の出席者: 同じ時間に開催される必須の会議を2つ割り当てることはできない。
- 必要とされる部屋の収容人数: 会議の出席者全員を収容できない部屋では会議を行ってはいけません。
- 同日中に開始して終了: 会議は複数の日にわたってスケジュールされないようにする。

中程度の制約:

- 任意の出席者: 同じ時間に開催される任意の会議を2つ割り当てることはできない。また、任意の会議と必須の会議を同じ時間に割り当てることはできない。

ソフト制約:

- 早い段階でスケジュール: すべての会議をできるだけ早くスケジュールする。
- 会議と会議の間の休憩時間: 会議と会議の間には、最低でも時間枠1つ分、休憩を入れる必要がある。

- 会議の重複: 並行して行われる会議の数を最小限に抑えて、どちらかの会議を選択しなければならない状況をなくす。
- 先に大きい部屋から割り当てる: 参加者が登録していない場合でも、できるだけ多数の参加者を収容するために、大きい部屋が空いている場合にはその部屋から割り当てていく必要がある。
- 部屋の不変性: 会議が連続して行われ、休憩の時間枠が2つ分より少ない場合には、会議は同じ部屋で行う方が良い。

問題の規模

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} .

9.11. コースの時間割 (ITC 2007 TRACK 3 - カリキュラムのスケジュール)

各授業を、時間枠および講義室に割り当ててスケジュールを組みます。

ハード制約:

- 講師の制約: 各講師は、同じ時間に授業を2つ受け持つことはできない。
- カリキュラムの制約: カリキュラムには、2つの授業を同じ時間に設定することはできない。
- 部屋の占有: 同じ時間の同じ講義室に、2つの授業を割り当てることはできない。
- 利用不可の時間 (データセットごとに指定): 授業には割り当てられない時間がある。

ソフト制約:

- 講義室の収容人数: 講義室の収容人数は、その授業を受ける学生の数よりも多くなければならない。
- 最小限の就業日数: 同じコースの授業の開講期間は、最短になるようにする。
- カリキュラムの緊密さ: 同じカリキュラムに含まれる授業は、時間帯を近く (連続した時間に) 設定する。
- 講義室の不変性: 同じコースの授業は同じ講義室を割り当てる必要がある。

この問題は、[International Timetabling Competition 2007 track 3](#) で定義されています。

問題の規模

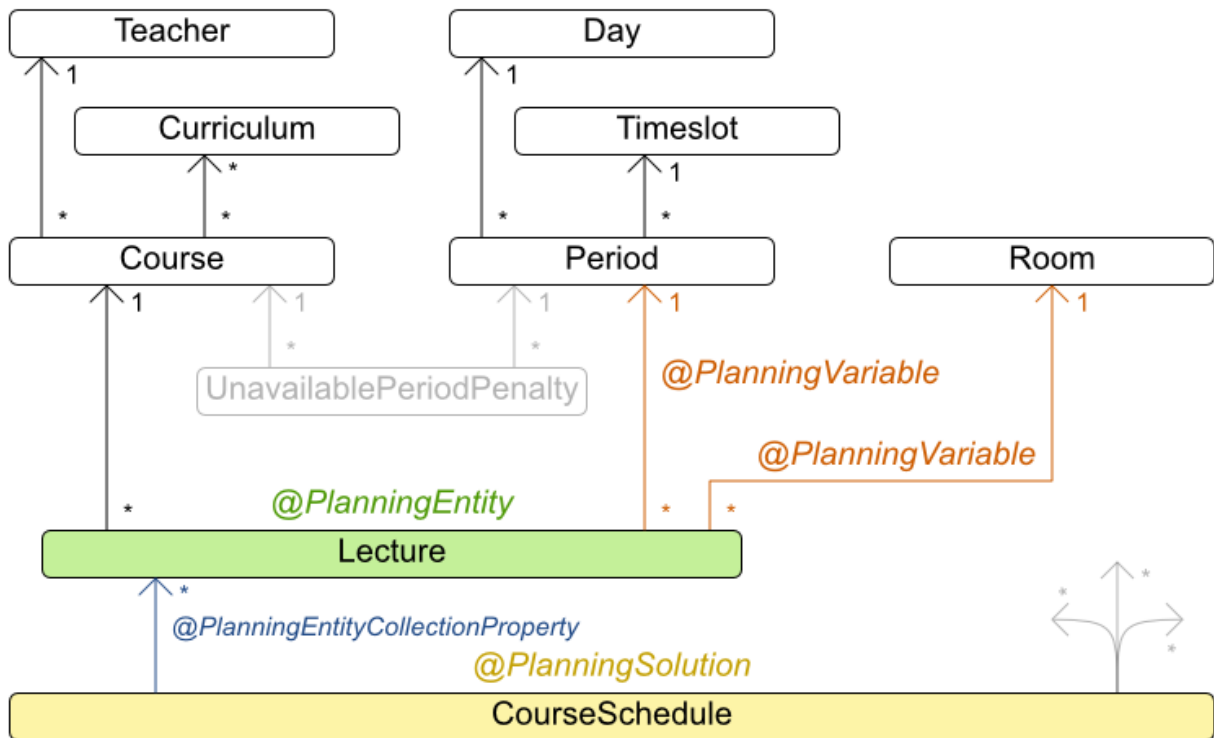
comp01 has 24 teachers, 14 curricula, 30 courses, 160 lectures, 30 periods, 6 rooms and 53 unavailable period constraints with a search space of 10^{360} .

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} .

図9.5 ドメインモデル

Curriculum course class diagram



9.12. マシンの再割当て (GOOGLE ROADEF 2012)

各プロセスをマシンに割り当てます。全プロセスには、すでに元の (最適化されていない) 割り当てがあります。プロセスにはそれぞれ、各リソース (CPU、メモリーなど) が一定量必要です。これは、クラウドのバランスの例の応用です。

ハード制約:

- 最大容量: マシンに割り当てる各リソースはこの量を超えてはいけません。
- 競合: 同じサービスのプロセスは別のマシンで実行する必要があります。
- 分散: 同じサービスのプロセスは複数の場所に分散させる必要があります。
- 依存関係: 他のサービスに依存するサービスのプロセスは、そのサービスの近くで実行する必要があります。
- 一時的な使用: リソースによっては一時的なものがあり、元のマシンと、新たに割り当てられたマシンの両方の最大容量にカウントされる。

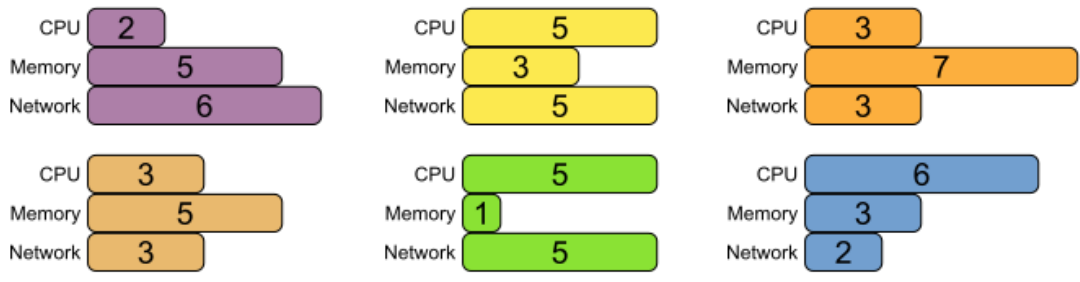
ソフト制約:

- 負荷: 各マシンの各リソースの安全容量を超えてはいけません。
- 負荷分散: 各マシンで利用可能なリソースを分散させて、今後の割り当てに対応できるように容量を空ける。
- プロセスの移動コスト: プロセスには移動コストが発生する。
- サービスの移動コスト: サービスには移動コストが発生する。
- 機械の移動コスト: マシン A からマシン B にプロセスを移動すると、A から B に固有の移動コストが別途発生する。

この問題は [the Google ROADEF/EURO Challenge 2012](#) で定義されています。

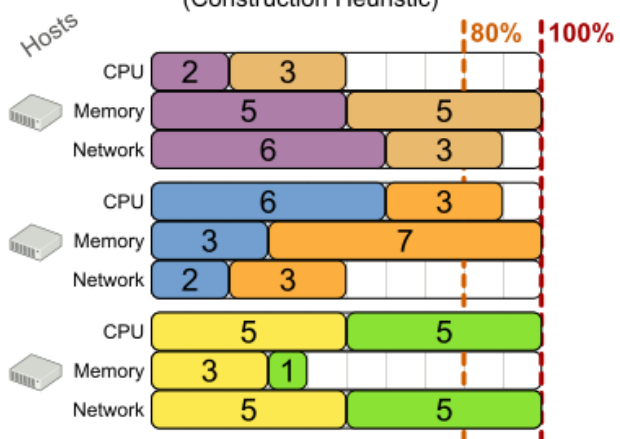
Cloud optimization is like Tetris

Processes



Traditional algorithm

(Construction Heuristic)



OptaPlanner

(Construction Heuristic + Local Search)

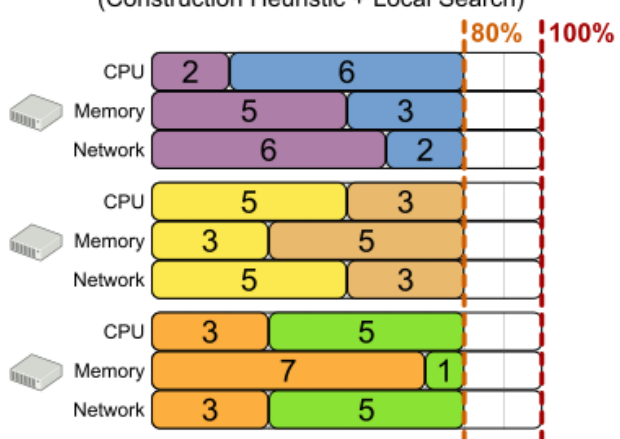
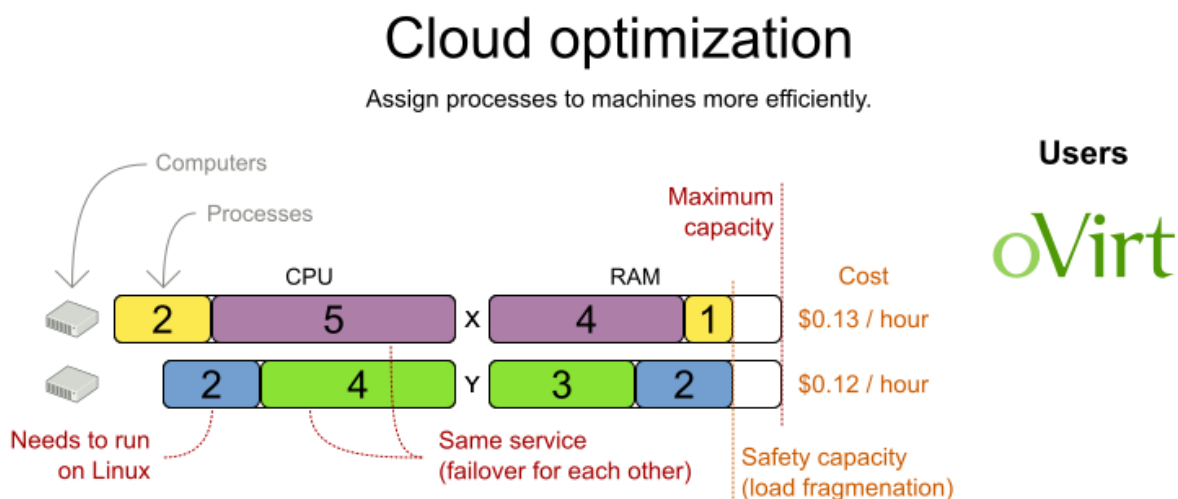


図9.6 価値提案



CloudBalancing benchmark

Cloud hosting cost

OptaPlanner versus traditional algorithm with domain knowledge

Average

-18%

Min/Max

-16%
-21%

datasets

5

Biggest dataset

1600 computers
4800 processes

5 mins Simulated Annealing vs First Fit Decreasing

MachineReassignment benchmark

Hardware congestion

OptaPlanner versus arbitrary feasible assignments

Average

-63%

Min/Max

-25%
-97%

datasets

20

Biggest dataset

50k machines
5k processes

5 mins Tabu Search vs First Feasible Fit

Don't believe us? Run our open benchmarks yourself: <http://www.optaplanner.org/code/benchmarks.html>

問題の規模

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} .

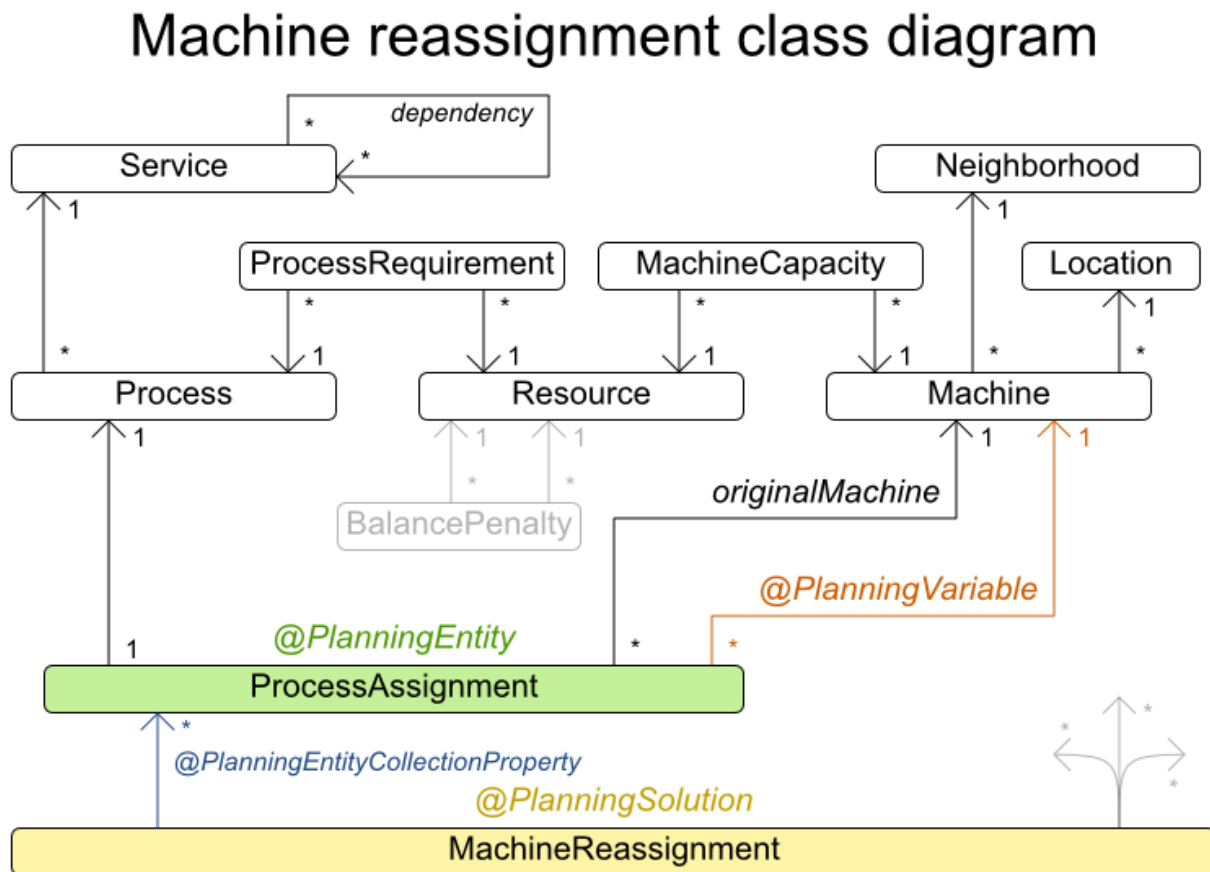
model_a2_5 has 12 resources, 5 neighborhoods, 25 locations, 50 machines, 153 services, 1000 processes and 0 balancePenalties with a search space of 10^{1698} .

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} .

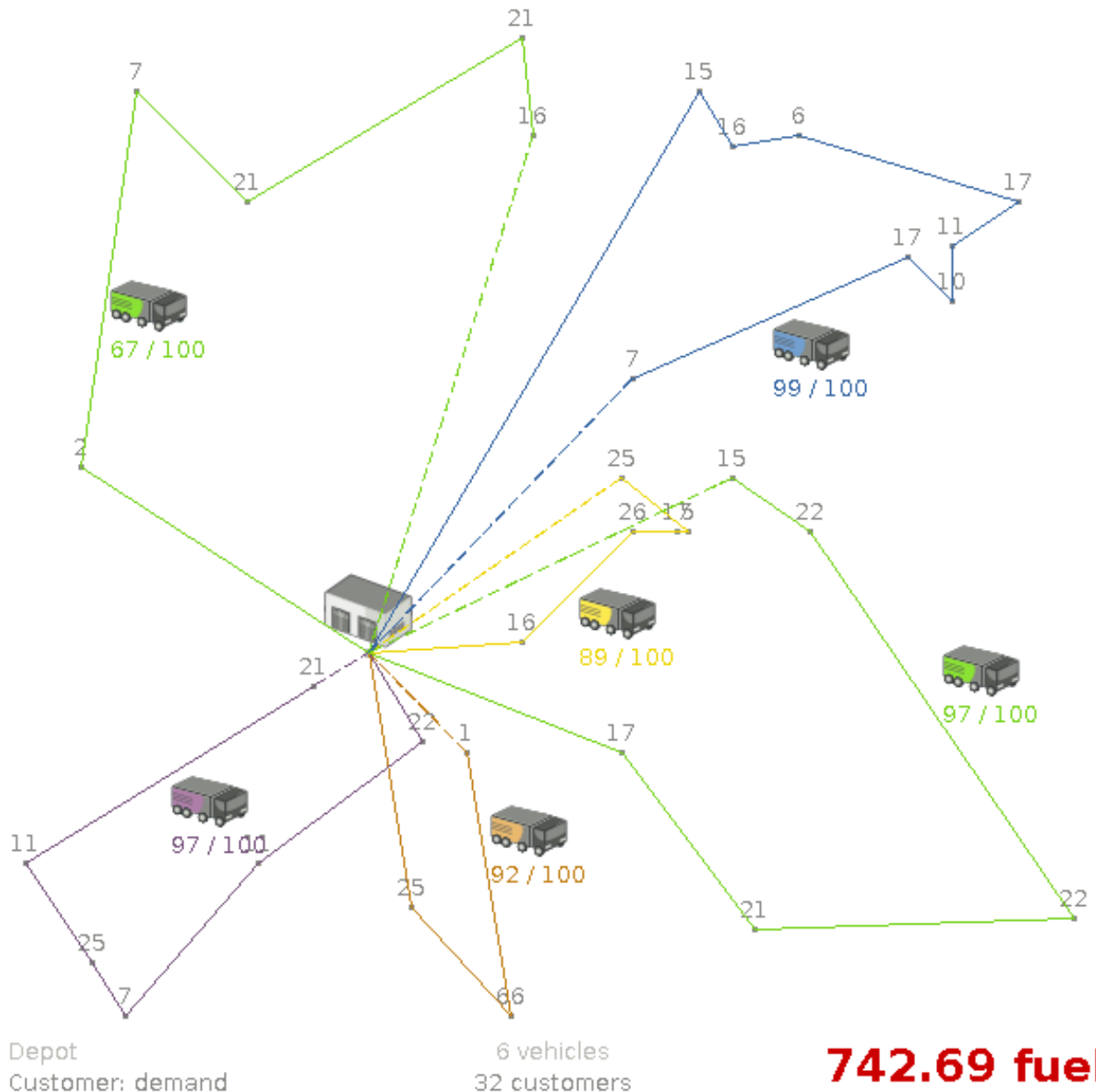
model_b_3 has 6 resources, 5 neighborhoods, 10 locations, 100 machines, 15025 services, 20000 processes and 0 balancePenalties with a search space of 10^{40000} .
 model_b_4 has 6 resources, 5 neighborhoods, 50 locations, 500 machines, 1732 services, 20000 processes and 1 balancePenalties with a search 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} .

図9.7 ドメインモデル



9.13. 配送経路

複数の車両を使用して、各顧客の品物を回収し、倉庫まで運びます。1つの車両で複数の顧客から品物を回収することはできませんが、収容できる容量には限りがあります。



基本例 (CVRP) のほかに、時間枠の設定が加わった例 (CVRPTW) もあります。

ハード制約:

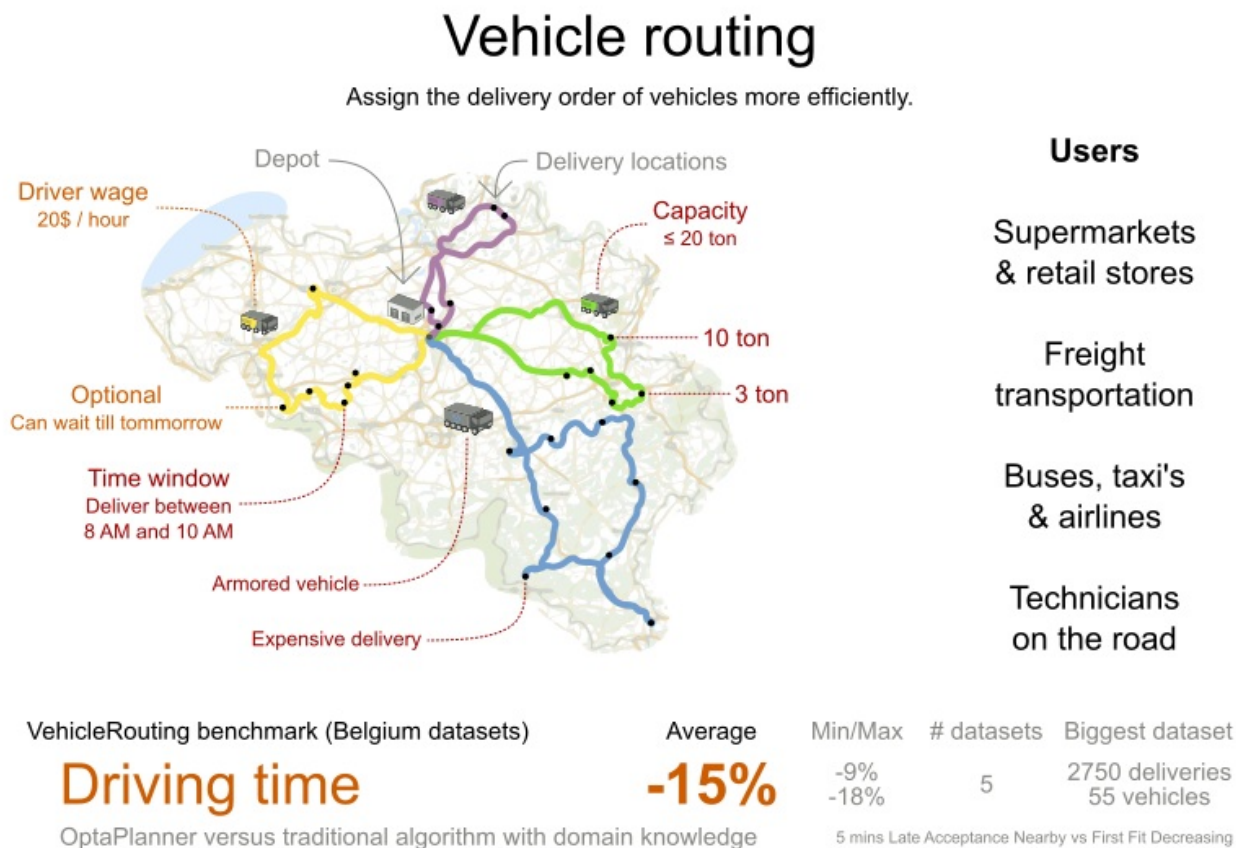
- 車両の容量: 車両は、車載容量を超えて品物を運ぶことができない。
- 時間枠 (CVRPTW のみ):
 - 移動時間: 別の場所に移動する場合には時間がかかる。
 - 顧客対応の時間: 車両は顧客に対応している時間、顧客先にとどまる必要がある。
 - 顧客の準備が整う時間: 顧客の準備が整う前に車両が到着する可能性があるが、準備ができるまで待機してから顧客に対応する必要がある。
 - 顧客が設定した締め切り時間: 車両は、顧客が設定した締め切り時間までに到着する必要がある。

ソフト制約:

- 合計距離: 車両が移動する合計距離 (ガソリンの消費量) を最小限に抑える。

CVRP (Capacitated Vehicle Routing Problem) と CVRPTW (Capacitated Vehicle Routing Problem with Time Window) は、VRP Web で定義されています。

図9.8 価値提案



Don't believe us? Run our open benchmarks yourself: <http://www.optaplanner.org/code/benchmarks.html>

問題の規模

CVRP インスタンス (時間枠なし):

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} .

10⁸³⁸⁰.

belgium-road-time-n50-k10 has 1 depots, 10 vehicles and 49 customers with a search space of 10⁷⁴.

belgium-road-time-n100-k10 has 1 depots, 10 vehicles and 99 customers with a search space of 10¹⁷⁰.

belgium-road-time-n500-k20 has 1 depots, 20 vehicles and 499 customers with a search space of 10¹¹⁶⁸.

belgium-road-time-n1000-k20 has 1 depots, 20 vehicles and 999 customers with a search space of 10²⁶⁰⁷.

belgium-road-time-n2750-k55 has 1 depots, 55 vehicles and 2749 customers with a search space of 10⁸³⁸⁰.

belgium-d2-n50-k10 has 2 depots, 10 vehicles and 48 customers with a search space of 10⁷⁴.

belgium-d3-n100-k10 has 3 depots, 10 vehicles and 97 customers with a search space of 10¹⁷⁰.

belgium-d5-n500-k20 has 5 depots, 20 vehicles and 495 customers with a search space of 10¹¹⁶⁸.

belgium-d8-n1000-k20 has 8 depots, 20 vehicles and 992 customers with a search space of 10²⁶⁰⁷.

belgium-d10-n2750-k55 has 10 depots, 55 vehicles and 2740 customers with a search space of 10⁸³⁸⁰.

A-n32-k5 has 1 depots, 5 vehicles and 31 customers with a search space of 10⁴⁰.

A-n33-k5 has 1 depots, 5 vehicles and 32 customers with a search space of 10⁴¹.

A-n33-k6 has 1 depots, 6 vehicles and 32 customers with a search space of 10⁴².

A-n34-k5 has 1 depots, 5 vehicles and 33 customers with a search space of 10⁴³.

A-n36-k5 has 1 depots, 5 vehicles and 35 customers with a search space of 10⁴⁶.

A-n37-k5 has 1 depots, 5 vehicles and 36 customers with a search space of 10⁴⁸.

A-n37-k6 has 1 depots, 6 vehicles and 36 customers with a search space of 10⁴⁹.

A-n38-k5 has 1 depots, 5 vehicles and 37 customers with a search space of 10⁴⁹.

A-n39-k5 has 1 depots, 5 vehicles and 38 customers with a search space of 10⁵¹.

A-n39-k6 has 1 depots, 6 vehicles and 38 customers with a search space of 10⁵².

A-n44-k7 has 1 depots, 7 vehicles and 43 customers with a search space of 10⁶¹.

A-n45-k6 has 1 depots, 6 vehicles and 44 customers with a search space of 10⁶².

A-n45-k7 has 1 depots, 7 vehicles and 44 customers with a search space of 10⁶³.

A-n46-k7 has 1 depots, 7 vehicles and 45 customers with a search space of 10⁶⁵.

A-n48-k7 has 1 depots, 7 vehicles and 47 customers with a search space of 10⁶⁸.

A-n53-k7 has 1 depots, 7 vehicles and 52 customers with a search space of 10⁷⁷.

A-n54-k7 has 1 depots, 7 vehicles and 53 customers with a search space of 10⁷⁹.

A-n55-k9 has 1 depots, 9 vehicles and 54 customers with a search space of 10⁸².

A-n60-k9 has 1 depots, 9 vehicles and 59 customers with a search space of 10⁹¹.

A-n61-k9 has 1 depots, 9 vehicles and 60 customers with a search space of 10⁹³.

A-n62-k8 has 1 depots, 8 vehicles and 61 customers with a search space of 10⁹⁴.

A-n63-k9 has 1 depots, 9 vehicles and 62 customers with a search space of 10⁹⁷.

A-n63-k10 has 1 depots, 10 vehicles and 62 customers with a search space of 10⁹⁸.

A-n64-k9 has 1 depots, 9 vehicles and 63 customers with a search space of 10⁹⁹.

A-n65-k9 has 1 depots, 9 vehicles and 64 customers with a search space of 10¹⁰¹.

A-n69-k9 has 1 depots, 9 vehicles and 68 customers with a search space of 10¹⁰⁸.

A-n80-k10 has 1 depots, 10 vehicles and 79 customers with a search space of 10¹³⁰.

F-n45-k4 has 1 depots, 4 vehicles and 44 customers with a search space of 10⁶⁰.

F-n72-k4 has 1 depots, 4 vehicles and 71 customers with a search space of 10¹⁰⁸.

F-n135-k7 has 1 depots, 7 vehicles and 134 customers with a search space of 10²⁴⁰.

CVRPTW インスタンス (時間枠あり):

belgium-tw-d2-n50-k10 has 2 depots, 10 vehicles and 48 customers with a search space of

10⁷⁴.

belgium-tw-d3-n100-k10 has 3 depots, 10 vehicles and 97 customers with a search space of 10¹⁷⁰.

belgium-tw-d5-n500-k20 has 5 depots, 20 vehicles and 495 customers with a search space of 10¹¹⁶⁸.

belgium-tw-d8-n1000-k20 has 8 depots, 20 vehicles and 992 customers with a search space of 10²⁶⁰⁷.

belgium-tw-d10-n2750-k55 has 10 depots, 55 vehicles and 2740 customers with a search space of 10⁸³⁸⁰.

belgium-tw-n50-k10 has 1 depots, 10 vehicles and 49 customers with a search space of 10⁷⁴.

belgium-tw-n100-k10 has 1 depots, 10 vehicles and 99 customers with a search space of 10¹⁷⁰.

belgium-tw-n500-k20 has 1 depots, 20 vehicles and 499 customers with a search space of 10¹¹⁶⁸.

belgium-tw-n1000-k20 has 1 depots, 20 vehicles and 999 customers with a search space of 10²⁶⁰⁷.

belgium-tw-n2750-k55 has 1 depots, 55 vehicles and 2749 customers with a search space of 10⁸³⁸⁰.

Solomon_025_C101 has 1 depots, 25 vehicles and 25 customers with a search space of 10⁴⁰.

Solomon_025_C201 has 1 depots, 25 vehicles and 25 customers with a search space of 10⁴⁰.

Solomon_025_R101 has 1 depots, 25 vehicles and 25 customers with a search space of 10⁴⁰.

Solomon_025_R201 has 1 depots, 25 vehicles and 25 customers with a search space of 10⁴⁰.

Solomon_025_RC101 has 1 depots, 25 vehicles and 25 customers with a search space of 10⁴⁰.

Solomon_025_RC201 has 1 depots, 25 vehicles and 25 customers with a search space of 10⁴⁰.

Solomon_100_C101 has 1 depots, 25 vehicles and 100 customers with a search space of 10¹⁸⁵.

Solomon_100_C201 has 1 depots, 25 vehicles and 100 customers with a search space of 10¹⁸⁵.

Solomon_100_R101 has 1 depots, 25 vehicles and 100 customers with a search space of 10¹⁸⁵.

Solomon_100_R201 has 1 depots, 25 vehicles and 100 customers with a search space of 10¹⁸⁵.

Solomon_100_RC101 has 1 depots, 25 vehicles and 100 customers with a search space of 10¹⁸⁵.

Solomon_100_RC201 has 1 depots, 25 vehicles and 100 customers with a search space of 10¹⁸⁵.

Homberger_0200_C1_2_1 has 1 depots, 50 vehicles and 200 customers with a search space of 10⁴²⁹.

Homberger_0200_C2_2_1 has 1 depots, 50 vehicles and 200 customers with a search space of 10⁴²⁹.

Homberger_0200_R1_2_1 has 1 depots, 50 vehicles and 200 customers with a search space of 10⁴²⁹.

Homberger_0200_R2_2_1 has 1 depots, 50 vehicles and 200 customers with a search space of 10⁴²⁹.

Homberger_0200_RC1_2_1 has 1 depots, 50 vehicles and 200 customers with a search space of 10⁴²⁹.

Homberger_0200_RC2_2_1 has 1 depots, 50 vehicles and 200 customers with a search space of 10⁴²⁹.

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} .

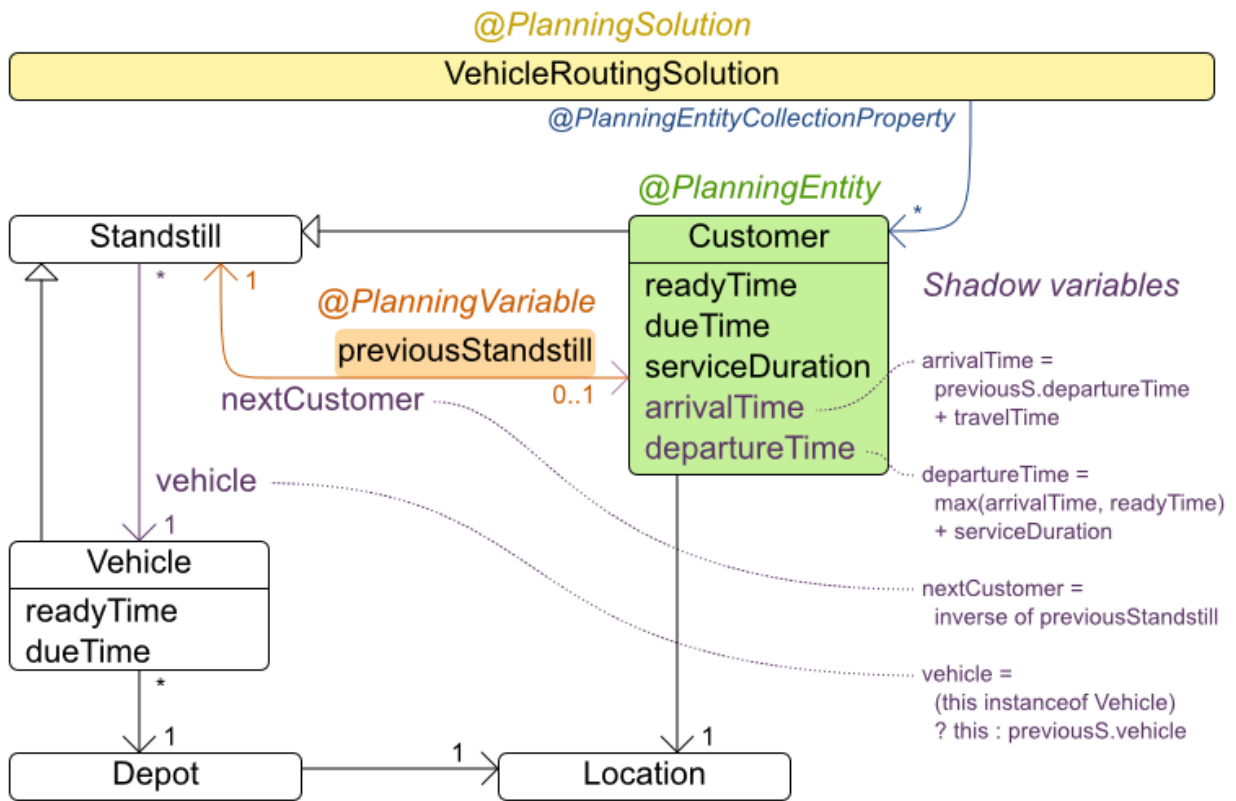
Homberger_1000_R210_1 has 1 depots, 250 vehicles and 1000 customers with a search space of 10^{2840} .

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} .

9.13.1. 配送経路のドメインモデル

Vehicle routing class diagram



時間枠ありの配送経路のドメインモデルでは、シャドウ変数の機能を多用します。こうすることで、**arrivalTime** や **departureTime** などのプロパティーがドメインモデルで直接利用できるため、制約をより自然に表現できます。

直線距離ではなく道路の距離

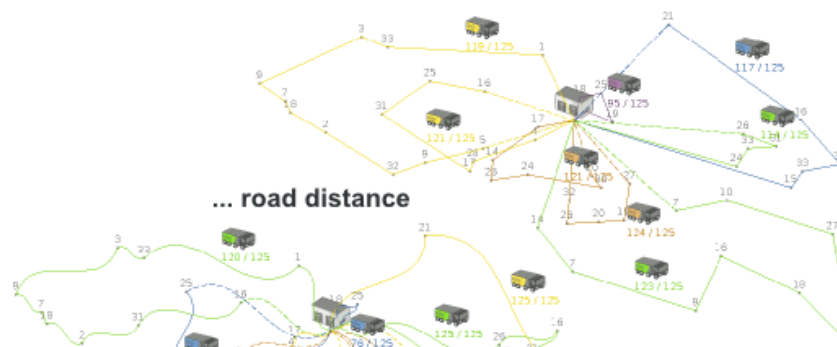
車は、直線距離を移動するのではなく、道路や高速道路を使用する必要があります。ビジネスの観点からすると、これは非常に重要です。

Vehicle routing distance type

Can we optimize for air distances, when we need road distances or driving times?

Optimized for ...

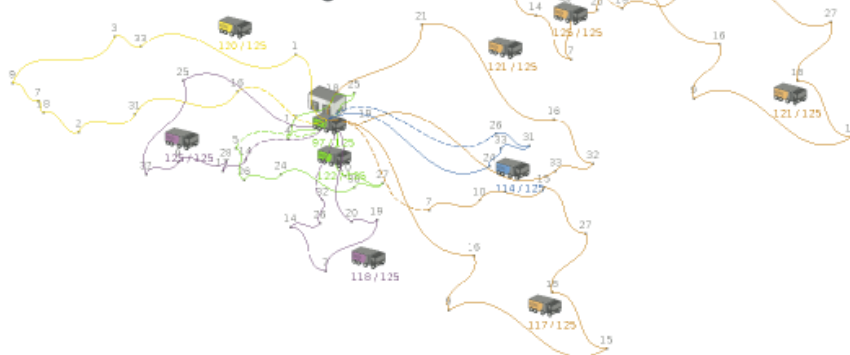
... air distance



... road distance



... driving time



2 327.32 km 118 632 sec

3.8% worse 4.0% worse

2 243.15 km 115 516 sec

best 1.2% worse

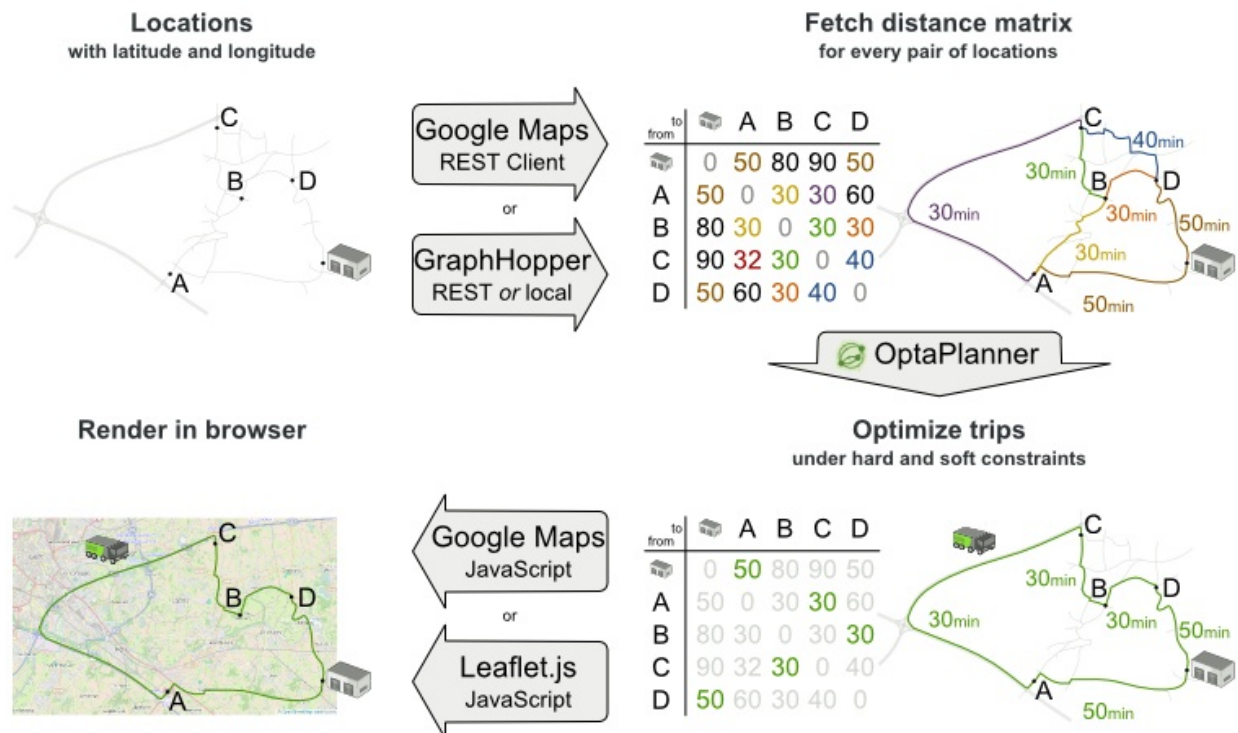
2 300.32 km 114 105 sec

2.5% worse best

最適化アルゴリズムでは、2点の距離を検索できている(できれば、事前に計算されている)場合には、これは特に重要ではありません。道路費は距離である必要はありません。また、移動時間、フラッシュコスト、または加重関数を使用することもできます。[GraphHopper](#) (埋め込み可能なオフライン Java エンジン)、[Open MapQuest](#) (web サービス)、[Google Maps Client API](#) (web サービス) など、移動コストを事前に計算する技術があります。

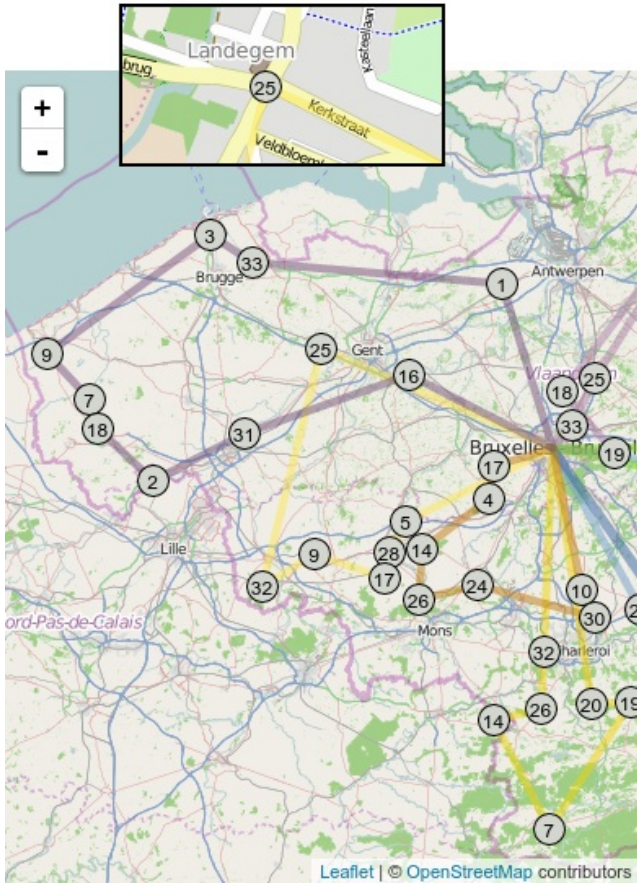
Integration with real maps

Google Maps or GraphHopper (OpenStreetMap) calculate distances, OptaPlanner optimizes the trips.

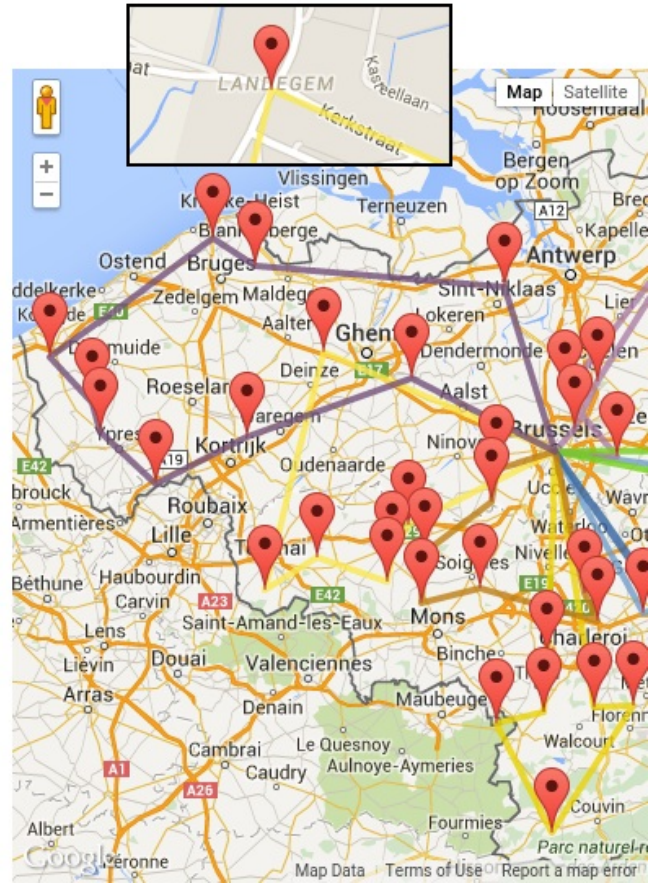


また、Leaflet や Google Maps for developers など、レンダリングする技術も複数あります。

Leaflet.js



Google Maps



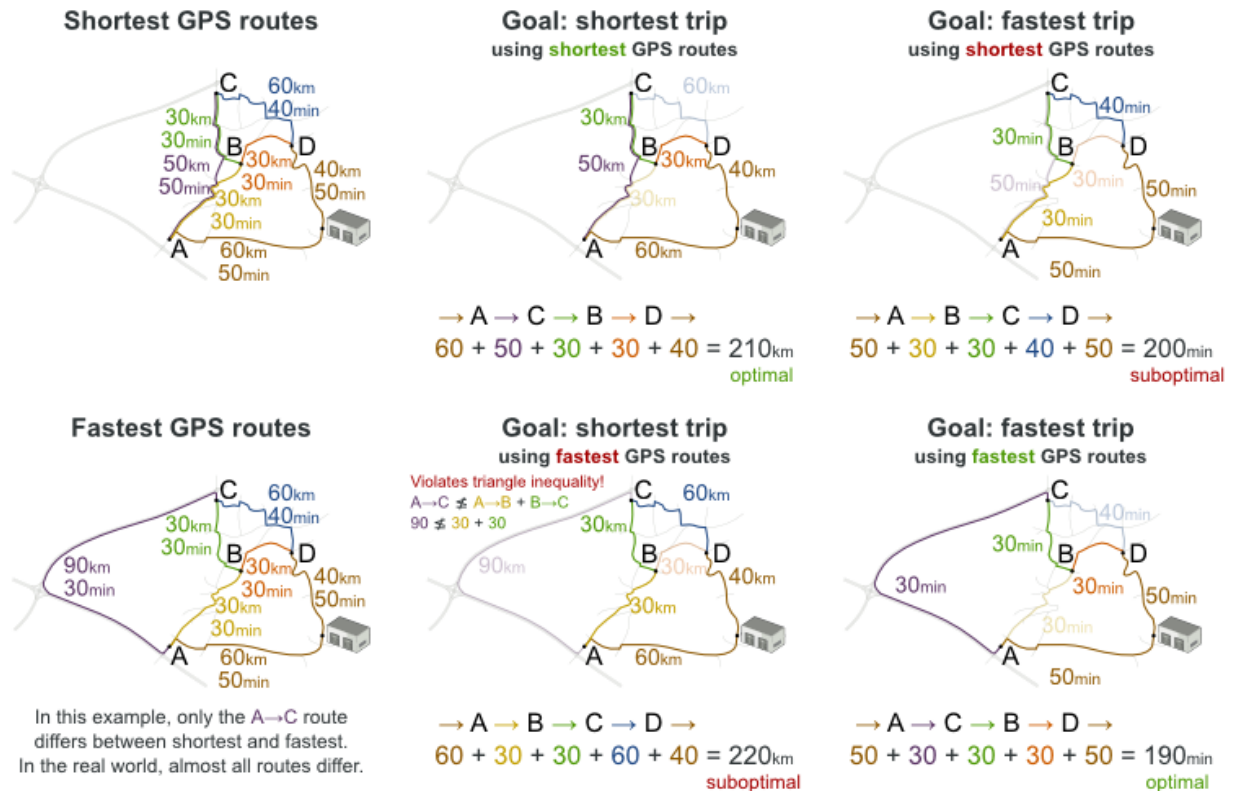
GraphHopper または Google Map Directions を使用して実際の経路をレンダリングすることも可能ですが、高速道路で経路が重なるため、停止する順番を確認するのが困難になります。



2点間の移動費は、OptaPlannerで使用されるのと同じ最適化条件を使用する点に注意してください。たとえば、GraphHopperはデフォルトで、最短ではなく、最速の経路を返します。最速のGPS経路のkm(またはマイル)の距離を使用して、OptaPlannerで最短の移動を最適化しないようにしてください。以下のように、準最適解が導き出される可能性があります。

Road distance triangle inequality

Routes and trips must optimize the same property to avoid suboptimal solutions.



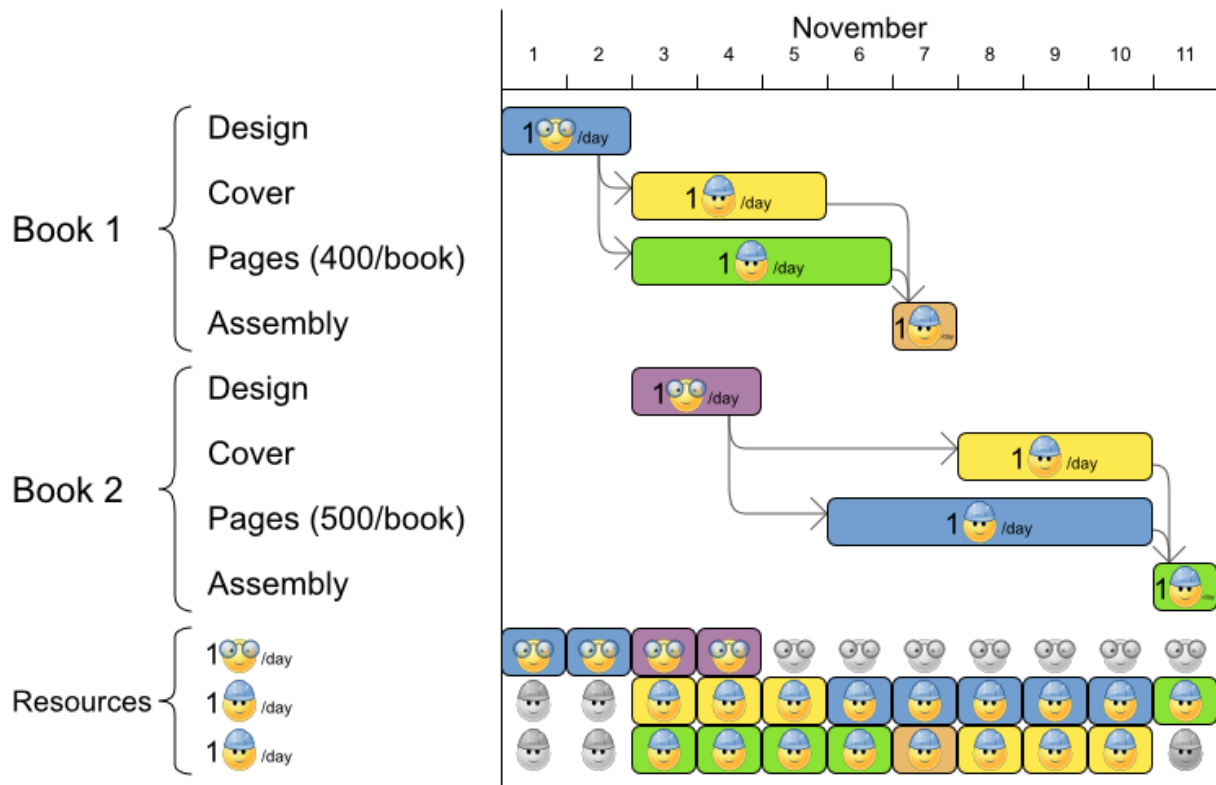
一般的な考え方とは異なり、多くのユーザーは最短の経路ではなく、最速の経路を使用したいと考えます。通常の道路よりも高速道路の使用を好みます。舗装されていない道よりも舗装されている道路を好みます。実際には、最速の経路と、最短の経路が同じであることはほとんどありません。

9.14. プロジェクトジョブのスケジュール

プロジェクトの遅延を最小限に抑えるために、すべてのジョブを時間内に実行できるようにスケジュールを設定します。各ジョブは、プロジェクトに含まれます。ジョブは、異なる方法で実行できます。方法ごとに期間や使用するリソースが異なります。これは、柔軟な **ジョブショップスケジューリング (JSP)** の応用です。

Project job scheduling

For each job, choose an execution mode and a start time.



ハード制約:

- ジョブの優先順位: ジョブは、先行のジョブがすべて完了するまで開始しない。
- リソースの容量: 利用可能な量を超えるリソースを使用しない。
 - リソースはローカル (同じプロジェクトのジョブ間で共有)、またはグローバル (全ジョブ間で共有) とする。
 - リソースは更新可能 (1日に利用可能な容量) または更新不可 (全日で利用可能な容量) とする。

中程度の制約:

- プロジェクトの合計遅延時間: 各プロジェクトの所要時間 (メイクスパン) を最短にする。

ソフト制約:

- メイクスパン合計: 複数のプロジェクトスケジュールの合計所要時間を最短にする。

この問題は、[the MISTA 2013 challenge](#) で定義されています。

問題の規模

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.

9.15. タスクの割り当て

従業員のキューのスポットに各タスクを割り当てます。タスクごとに、従業員のアフィニティーレベルから影響を受ける期間と、タスクの顧客が含まれます。

ハード制約:

- スキル: タスクごとに1つ以上のスキルが必要である。従業員には、このようなスキルがすべて必要です。

ソフトレベル0の制約:

- 極めて重要なタスク: 主要なタスクやマイナーなタスクの前に、極めて重要なタスクを完了する。

ソフトレベル1の制約:

- メークスパンの最小化: 全タスクを完了するまでの時間を短縮する。

〜 熟練度の高い従業員から順番に始めていき、公平性を保つ。レベル0、1、2のタスクを作成する。

- 勤務歴の長い従業員から順番に遅めし、公平性やロードバランシングを作成する。

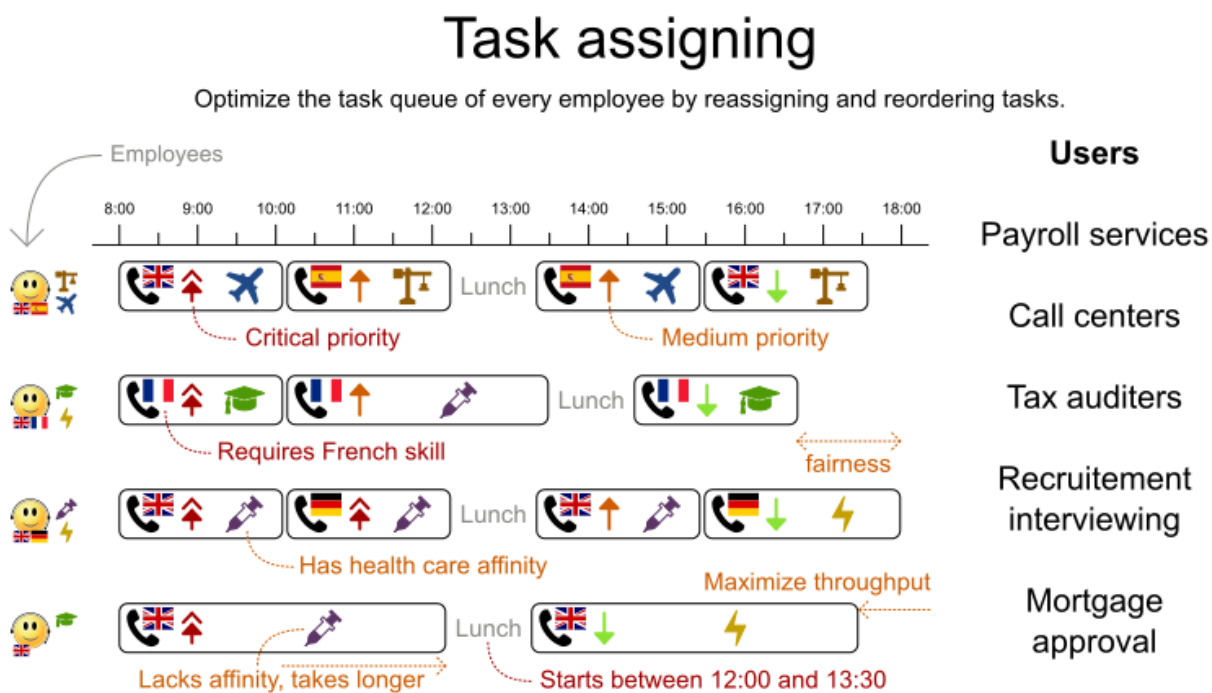
ソフトレベル2の制約:

- 主要なタスク: マイナーなタスクの前に、主要なタスクをできるだけ早く完了する。

ソフトレベル3の制約:

- マイナーなタスク: できるだけ早くマイナーなタスクを完了する。

図9.9 価値提案



問題の規模

24tasks-8employees has 24 tasks, 6 skills, 8 employees, 4 task types and 4 customers with a search space of 10^{30} .

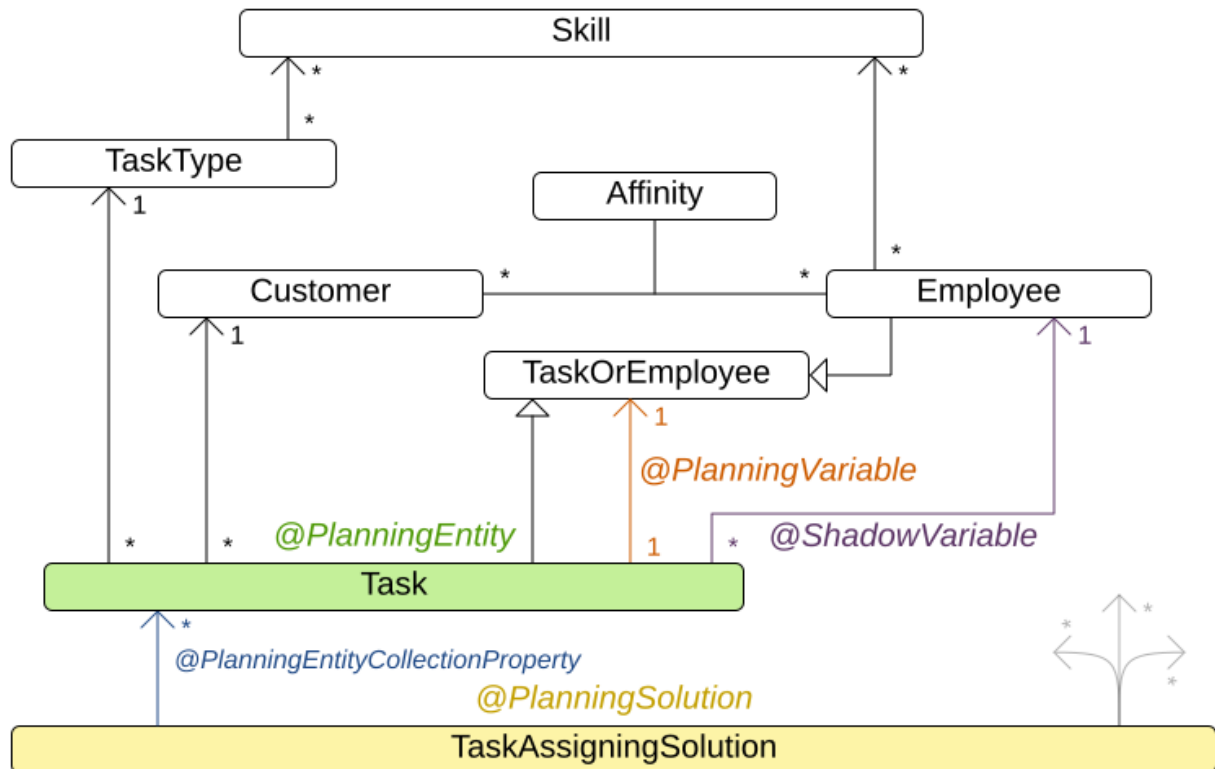
50tasks-5employees has 50 tasks, 5 skills, 5 employees, 10 task types and 10 customers with a search space of 10^{69} .

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} .

図9.10 ドメインモデル

Task assigning class diagram

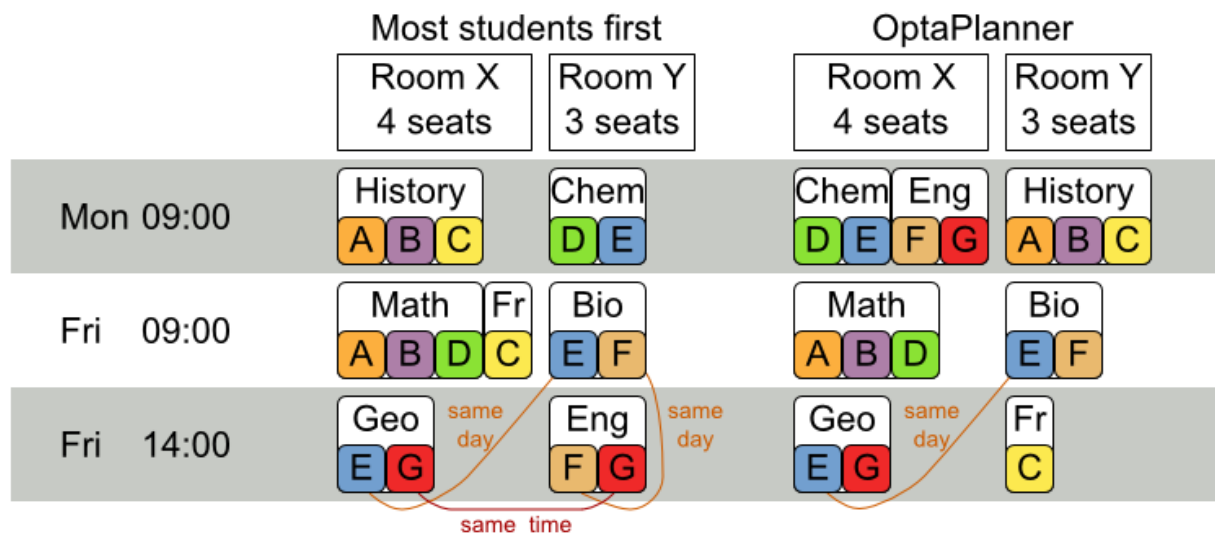
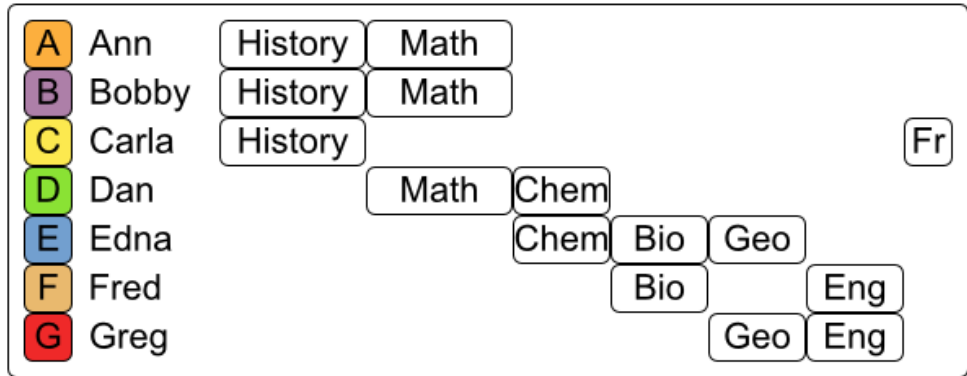


9.16. 試験の時間割 (ITC 2007 TRACK 1 - 試験)

すべての試験に、時間と部屋を割り当てます。同じ時間帯に同じ部屋で、複数の試験を行うことができるものとします。

Examination timetabling

Assign each exam a period and a room.



ハード制約:

- 試験の制約: 同じ学生が受ける 2 つの試験は、同じ時間帯に実施できないものとする。
- 教室の収容人数: 教室の座席数は、常に受験者数よりも多くなければならない。
- 期間: 期間は、すべての試験に対応できる長さでなければならない。
- 期間関連のハード制約 (データセットごとに指定):
 - 一致: 特定の 2 つの試験を同じ時間帯に設定する必要がある (別の教室を使用することも可能)。
 - 除外: 特定の 2 つの試験を同じ時間帯に設定できない。
 - 以降: 特定の試験を、別の特定の試験の後に行う必要がある。
- 教室関連の制約 (データセットごとに指定):
 - 排他的: 特定の試験を、他の試験と同じ教室で行うことはできない。

ソフト制約 (パラメーター化されたペナルティーがそれぞれ設定されている):

- 同じ学生が、続けて試験を 2 つ受けてはいけない。
- 同じ学生が、同じ日に試験を 2 つ受けてはいけない。
- 時間帯の分散: 同じ学生が受ける 2 つの試験は、時間をある程度あける。

- 異なる試験の長さ: 教室を共有する 2 つの試験の長さは、同じにする。
- 前倒し: 規模の大きい試験は、スケジュールを早めに決定する。
- 期間のペナルティー (データセットごとに指定): 期間によっては、使用されるとペナルティーが発生する。
- 部屋のペナルティー (データセットごとに指定): 部屋によっては、使用されるとペナルティーが発生する。

実際に大学から取得した大規模な試験データセットを使用します。

この問題は、[International Timetabling Competition 2007 track 1](#) で定義されています。Geoffrey De Smet は、非常に初期バージョンの OptaPlanner で 4 位を終了しました。このコンペティション以降、多くの改良点が加えられています。

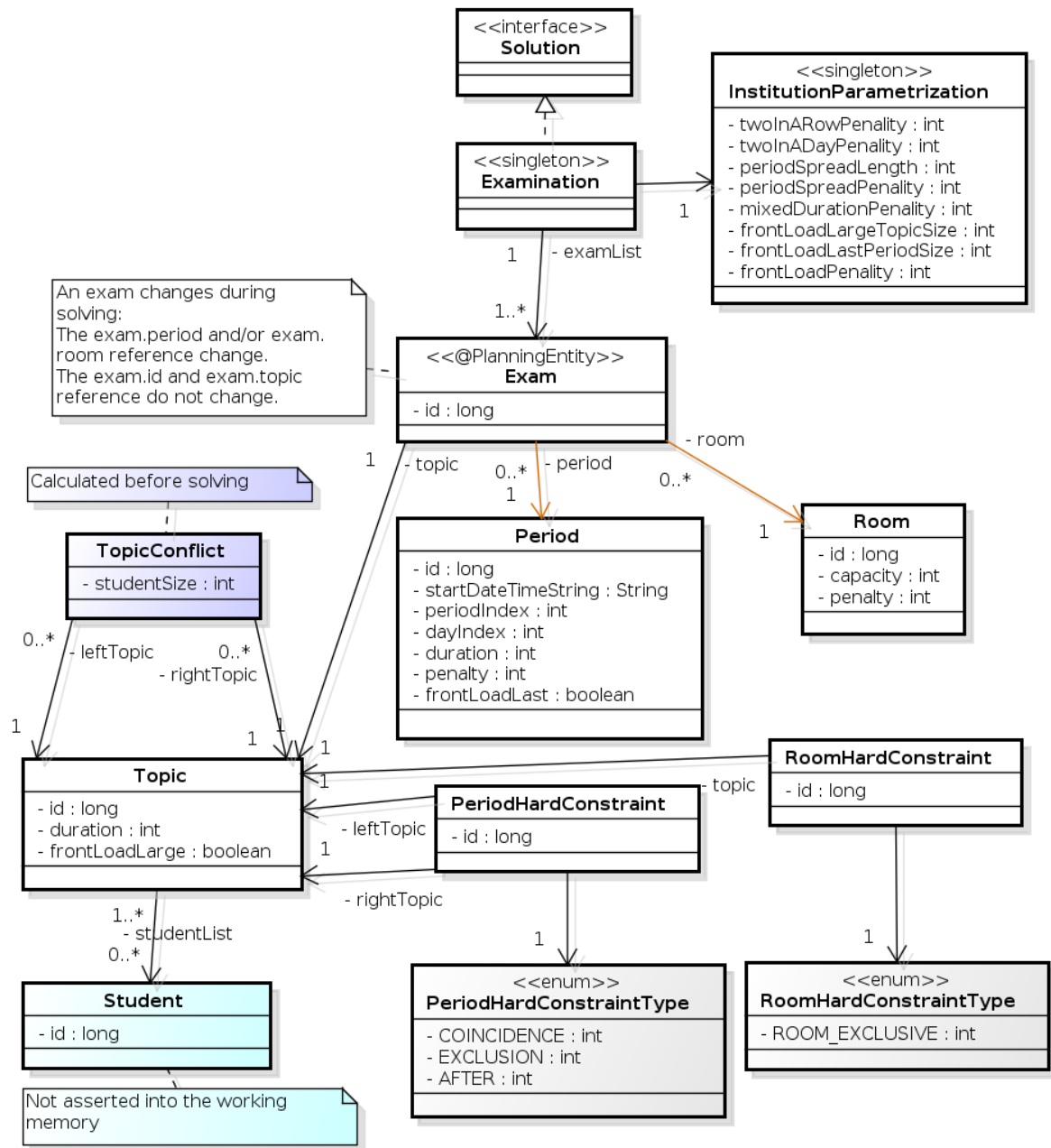
問題の規模

```
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.
```

9.16.1. テストの時間割のドメインモデル

以下の図では、主な試験のドメインクラスを紹介しています。

図9.11 試験のドメインクラスの図



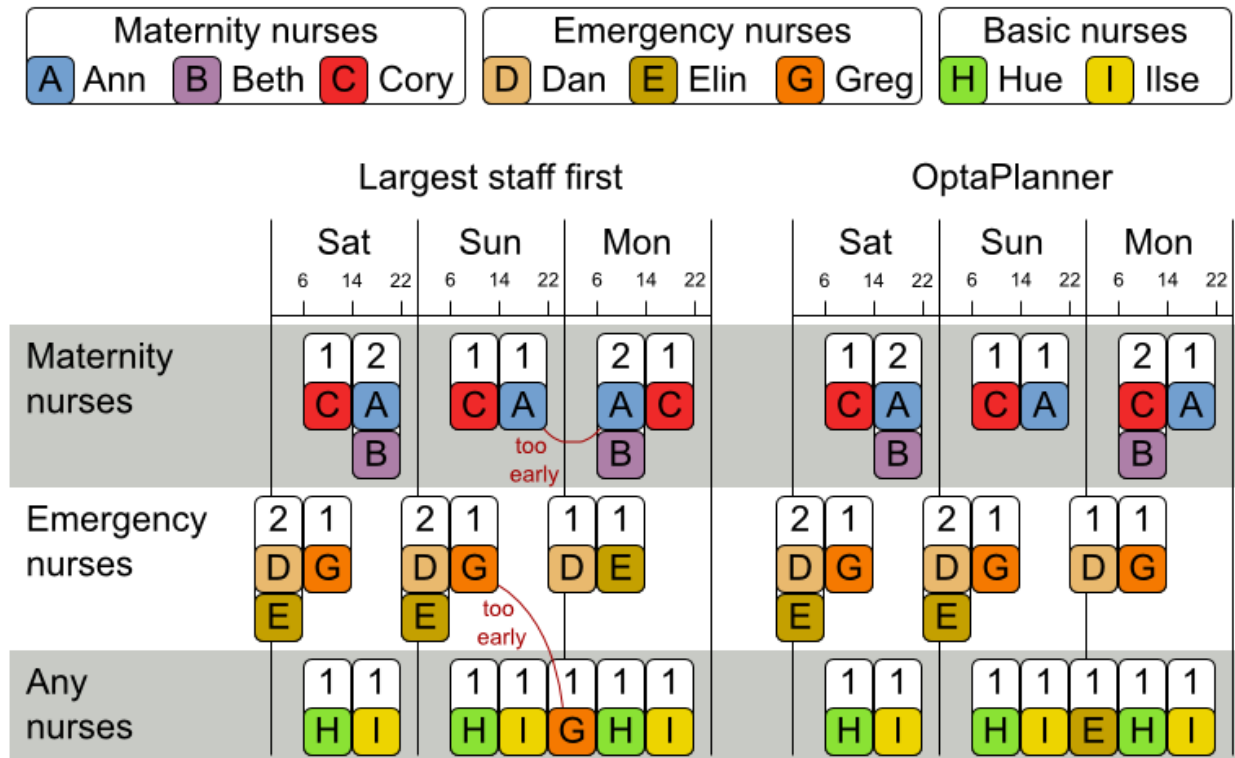
試験のコンセプトを、**Exam** クラスと **Topic** クラスに分けた点に注意してください。期間または教室のプロパティを変更し、解 (プランニングエンティティークラス) を求めると、**Exam** インスタンスが変化します。このとき、**Topic** インスタンス、**Period** インスタンス、および **Room** インスタンスは変化しません (他のクラスと同様、これらも問題ファクトです)。

9.17. 看護師の勤務表 (INRC 2010)

各シフトに看護師を割り当てます。

Employee shift rostering

Populate each work shift with a nurse.



ハード制約:

- 未割り当てのシフトなし (組み込み): すべてのシフトを従業員に割り当てる必要がある。
- シフトの制約: 従業員には1日に1シフトだけ割り当てることができる。

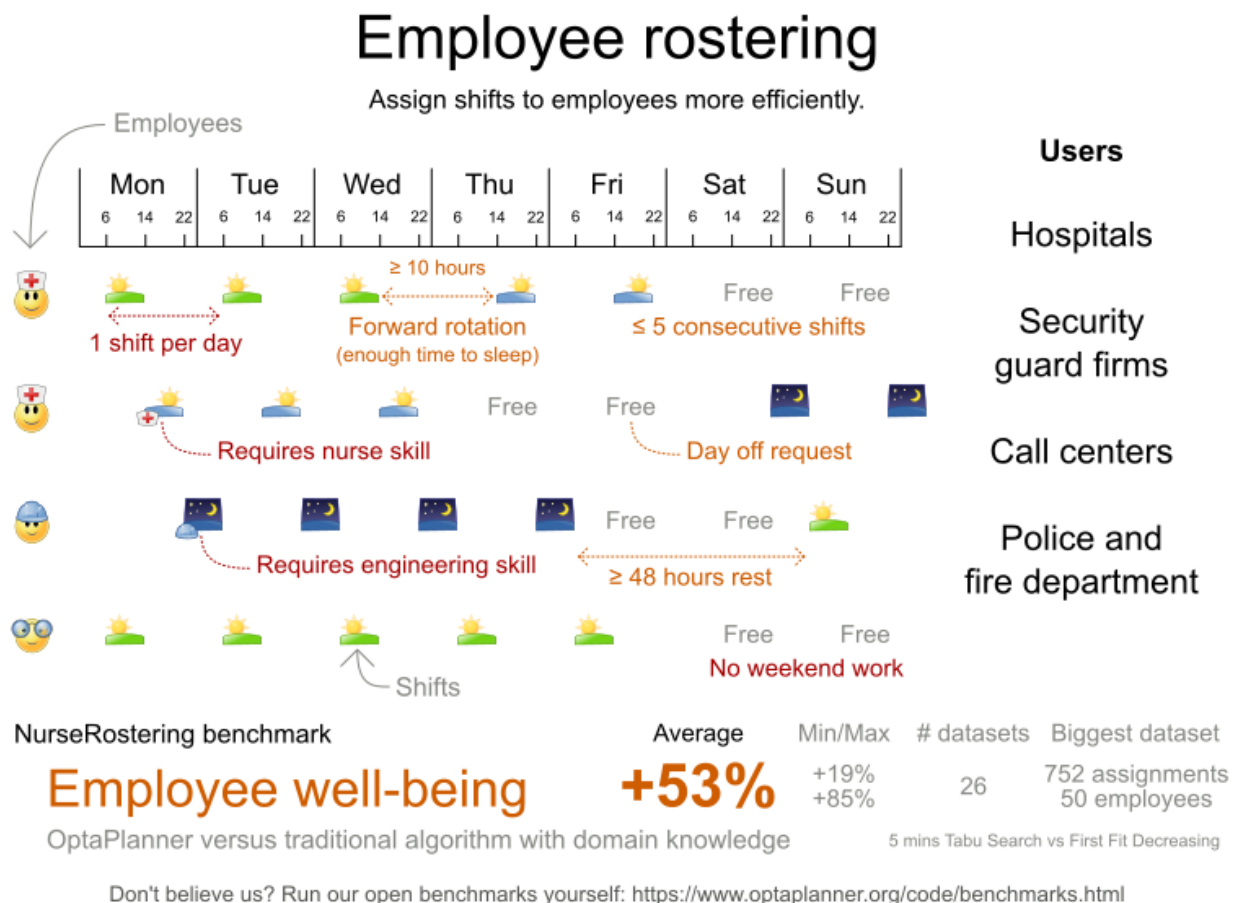
ソフト制約:

- 契約上の義務。この業界では、頻繁に契約上の義務に違反するため、ハード制約ではなく、ソフト制約として定義することに決定しました。
 - 割り当ての下限および上限: 各従業員は、(それぞれの契約に合わせて) x より多く、 y よりも少ないシフト数を勤務する必要がある。
 - 連続勤務日数の下限および上限: 各従業員は、(それぞれの契約に合わせて) 連続で x 日から y 日間、勤務する必要がある。
 - 連続公休日数の下限および上限: 各従業員は、(それぞれの契約に合わせて) 連続で x 日から y 日間、休む必要がある。
 - 週末に連続勤務する回数下限および上限: 各従業員は、(それぞれの契約に合わせて) 連続で x 回から y 回、週末勤務する必要がある。
 - 週末の勤務有無を同じにする: 各従業員は、週末の両日を勤務する、または休む必要がある。
 - 週末のシフトタイプを同じにする: 各従業員で、同じ週末のシフトタイプは、同じにする必要がある。

- 好ましくないシフトパターン: 遅番+早番+遅番など、好ましくないシフトタイプを連続で組み合わせさせたパターン。
- 従業員の希望:
 - 勤務日のリクエスト: 従業員は、特定の勤務希望日を申請できる。
 - 公休日のリクエスト: 従業員は、特定の公休希望日を申請できる。
 - 勤務するシフトのリクエスト: 従業員は特定のシフトへの割り当てを希望できる。
 - 勤務しないシフトのリクエスト: 従業員は特定のシフトに割り当てられないように希望できる。
- 他のスキル: スキルに割り当てられた従業員は、そのシフトに必要な全スキルに堪能である必要がある。

この問題は [International Nurse Rostering Competition 2010](#) で定義されています。

図9.12 価値提案



問題の規模

以下のように、データセットの種類は3つあります。

- sprint: 数秒で問題を解決する必要があります。
- medium: 数分で問題を解決する必要があります。
- long: 時間で解決する必要があります。

medium03 has 1 skills, 4 shiftTypes, 0 patterns, 4 contracts, 31 employees, 28 shiftDates, 608 shiftAssignments and 403 requests with a search space of 10^9 .

medium04 has 1 skills, 4 shiftTypes, 0 patterns, 4 contracts, 31 employees, 28 shiftDates, 608 shiftAssignments and 403 requests with a search space of 10^9 .

medium05 has 1 skills, 4 shiftTypes, 0 patterns, 4 contracts, 31 employees, 28 shiftDates, 608 shiftAssignments and 403 requests with a search space of 10^9 .

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 10^6 .

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 10^6 .

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 10^6 .

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 10^6 .

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 10^6 .

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 10^6 .

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 10^6 .

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 10^6 .

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^{12} .

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^{12} .

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^{12} .

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^{12} .

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^{12} .

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^{12} .

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^{12} .

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^{12} .

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^{12} .

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^{12} .

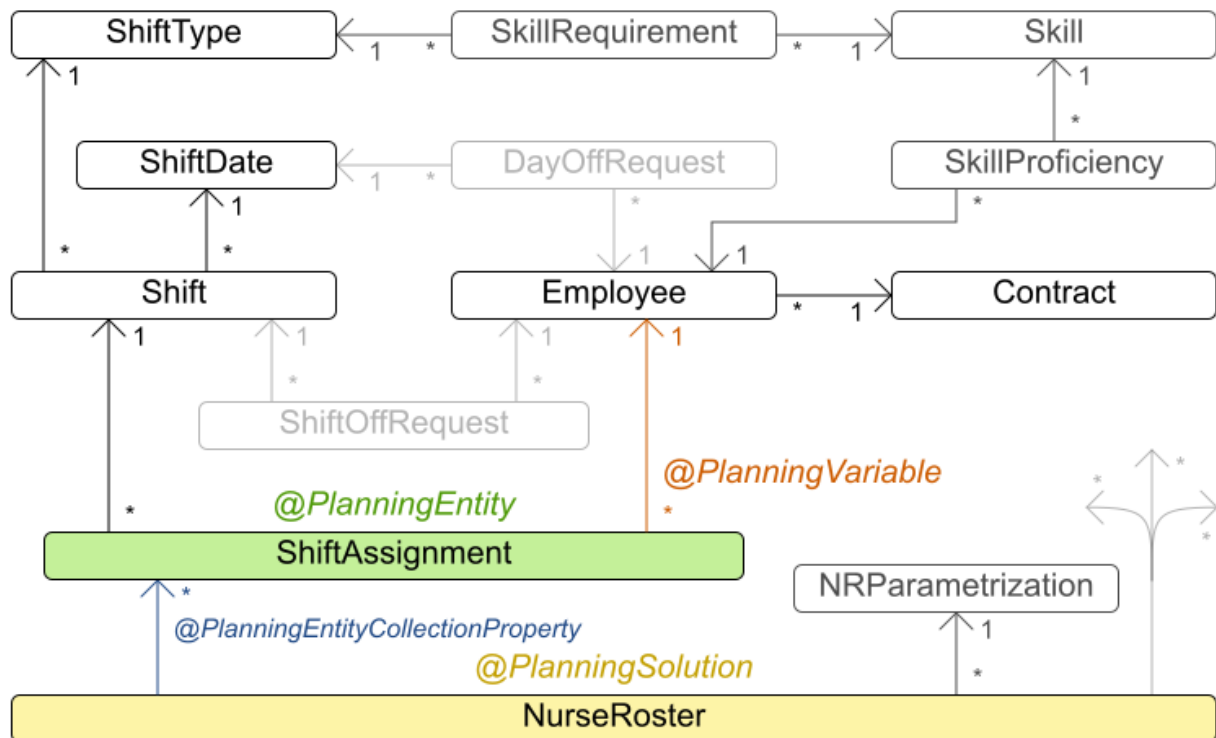
long_late03 has 2 skills, 5 shiftTypes, 9 patterns, 3 contracts, 50 employees, 28 shiftDates, 752 shiftAssignments and 0 requests with a search space of 10^{12} .

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^{12} .

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^{12} .

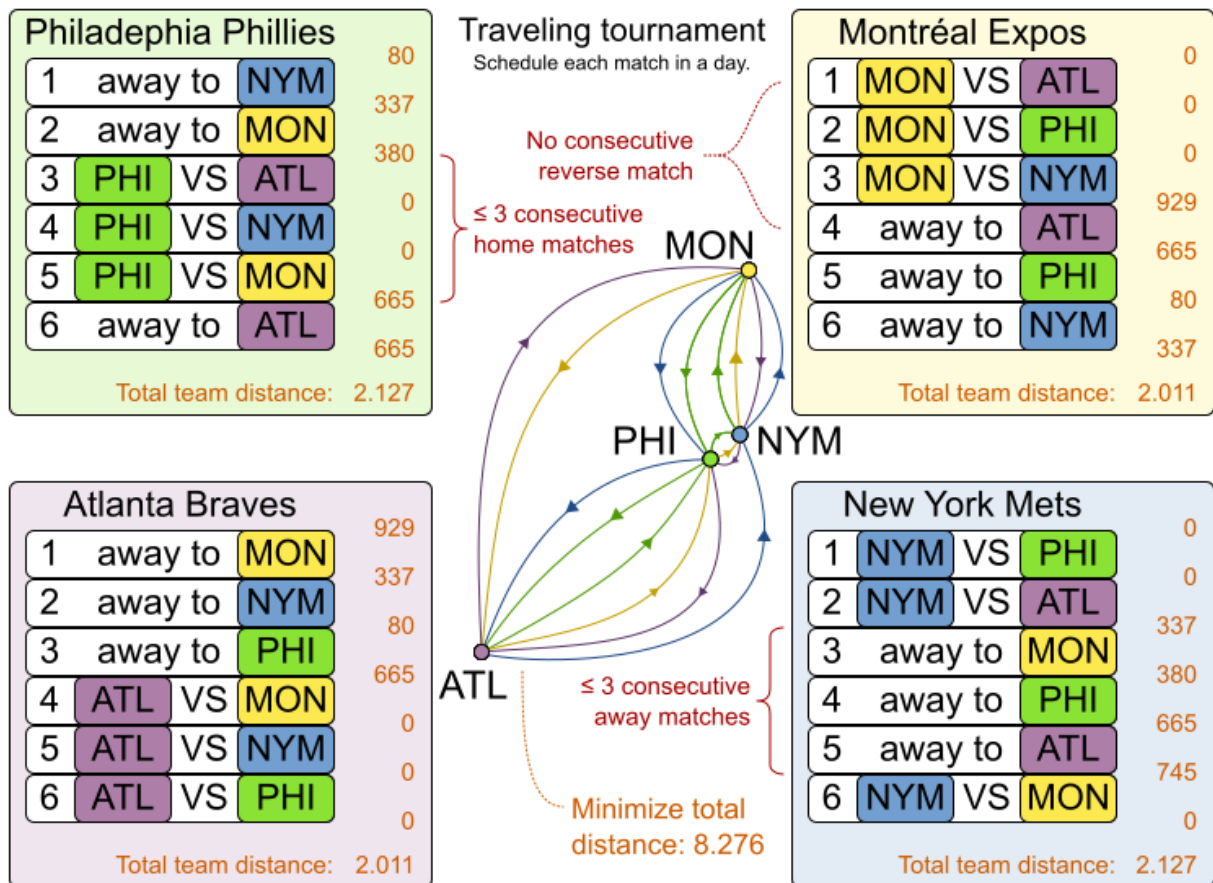
図9.13 ドメインモデル

Nurse rostering class diagram



9.18. 巡回トーナメント問題 (TTP)

n 人数のチーム間の一致をスケジュールします。



ハード制約:

- 各チームは、他のチームとそれぞれ2回(ホームとアウェイ)試合をする。
- 各チームは、各時間枠に1試合だけ行う。
- 3回連続で、ホームまたはアウェイでの試合はできない。
- 繰り返しなし: 同じ対戦相手と2回連続で対戦できない。

ソフト制約:

- 全チームが移動する合計距離を最小限に抑える。

この問題は [Michael Trick の Web サイト \(世界記録が含まれます\)](#) で定義されています。

問題の規模

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} .

9.19. コストを抑えるスケジュール

全タスクを時間内にスケジュールし、機械の電気代を最小限に抑えます。電気代は時間によって異なります。これは、**ジョブショップスケジューリング**の応用です。

ハード制約:

- 開始時間の制限: 各タスクは、最早と最遅の開始時間の制限内に、開始する必要がある。
- 最大容量: マシンに割り当てる各リソースはこの量を超えてはいけない。
- 開始および終了: 各機械は、タスクが割り当てられている間は稼働している必要がある。次のタスクまでの間、起動および終了コストを避けるため、機械をアイドルにすることができる。

中程度の制約:

- 電気代: 全スケジュールの合計電気代を最小限に抑える。
 - 機械の電気代: 稼働中またはアイドル中の機械はそれぞれ、電気を消費し、電気代が発生する (金額は使用時の電気代によって異なる)。
 - タスクの電気代: 各タスクも電気を消費し、電気代が発生する (金額は使用時の電気代によって異なる)。

- 機械の起動および終了コスト: 機械を起動または終了するたびに、追加のコストが発生する。

ソフト制約 (問題に元々設定されている定義に追加):

- 早く開始: なるべく早めにタスクを開始するようにする。

この問題は、[ICON challenge](#) で定義されています。

問題の規模

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

10¹⁶⁵⁹.
instance14 has 2 resources, 20 machines, 288 periods and 500 tasks with a search space of 10¹⁶⁴³.
instance15 has 3 resources, 40 machines, 288 periods and 500 tasks with a search space of 10¹⁷⁸².
instance16 has 3 resources, 40 machines, 288 periods and 500 tasks with a search space of 10¹⁷⁷⁸.
instance17 has 3 resources, 40 machines, 288 periods and 500 tasks with a search space of 10¹⁷⁶⁴.
instance18 has 3 resources, 40 machines, 288 periods and 500 tasks with a search space of 10¹⁷⁶⁹.
instance19 has 3 resources, 40 machines, 288 periods and 500 tasks with a search space of 10¹⁷⁷⁸.
instance20 has 3 resources, 50 machines, 288 periods and 1000 tasks with a search space of 10³⁶⁸⁹.
instance21 has 3 resources, 50 machines, 288 periods and 1000 tasks with a search space of 10³⁶⁷⁸.
instance22 has 3 resources, 50 machines, 288 periods and 1000 tasks with a search space of 10³⁷⁰⁶.
instance23 has 3 resources, 50 machines, 288 periods and 1000 tasks with a search space of 10³⁶⁷⁶.
instance24 has 3 resources, 50 machines, 288 periods and 1000 tasks with a search space of 10³⁶⁸¹.
instance25 has 3 resources, 60 machines, 288 periods and 1000 tasks with a search space of 10³⁷⁷⁴.
instance26 has 3 resources, 60 machines, 288 periods and 1000 tasks with a search space of 10³⁷³⁷.
instance27 has 3 resources, 60 machines, 288 periods and 1000 tasks with a search space of 10³⁷⁴⁴.
instance28 has 3 resources, 60 machines, 288 periods and 1000 tasks with a search space of 10³⁷³¹.
instance29 has 3 resources, 60 machines, 288 periods and 1000 tasks with a search space of 10³⁷⁴⁶.
instance30 has 4 resources, 70 machines, 288 periods and 2000 tasks with a search space of 10⁷⁷¹⁸.
instance31 has 4 resources, 70 machines, 288 periods and 2000 tasks with a search space of 10⁷⁷⁴⁰.
instance32 has 4 resources, 70 machines, 288 periods and 2000 tasks with a search space of 10⁷⁶⁸⁶.
instance33 has 4 resources, 70 machines, 288 periods and 2000 tasks with a search space of 10⁷⁶⁷².
instance34 has 4 resources, 70 machines, 288 periods and 2000 tasks with a search space of 10⁷⁶⁹⁵.
instance35 has 4 resources, 80 machines, 288 periods and 2000 tasks with a search space of 10⁷⁸⁰⁷.
instance36 has 4 resources, 80 machines, 288 periods and 2000 tasks with a search space of 10⁷⁸¹⁴.
instance37 has 4 resources, 80 machines, 288 periods and 2000 tasks with a search space of 10⁷⁷⁶⁴.
instance38 has 4 resources, 80 machines, 288 periods and 2000 tasks with a search space of 10⁷⁷³⁶.
instance39 has 4 resources, 80 machines, 288 periods and 2000 tasks with a search space of 10⁷⁷⁸³.
instance40 has 4 resources, 90 machines, 288 periods and 4000 tasks with a search space of 10¹⁵⁹⁷⁶.
instance41 has 4 resources, 90 machines, 288 periods and 4000 tasks with a search space of

10¹⁵935.
 instance42 has 4 resources, 90 machines, 288 periods and 4000 tasks with a search space of 10¹⁵887.
 instance43 has 4 resources, 90 machines, 288 periods and 4000 tasks with a search space of 10¹⁵896.
 instance44 has 4 resources, 90 machines, 288 periods and 4000 tasks with a search space of 10¹⁵885.
 instance45 has 4 resources, 100 machines, 288 periods and 5000 tasks with a search space of 10²⁰173.
 instance46 has 4 resources, 100 machines, 288 periods and 5000 tasks with a search space of 10²⁰132.
 instance47 has 4 resources, 100 machines, 288 periods and 5000 tasks with a search space of 10²⁰126.
 instance48 has 4 resources, 100 machines, 288 periods and 5000 tasks with a search space of 10²⁰110.
 instance49 has 4 resources, 100 machines, 288 periods and 5000 tasks with a search space of 10²⁰078.

9.20. 投資資産クラスの割り当て (ポートフォリオの最適化)

各資産クラスに投資する相対数を決定します。

ハード制約:

- リスクの最大値: 標準偏差合計は、標準偏差の最大値を超えてはならない。
 - 標準偏差合計の計算は、[Markowitz Portfolio Theory](#) を適用した、資産クラスの相対関係を考慮する必要がある。
- 地域の最大値: 地域ごとに数量の最大値がある。
- セクターの最大値: 各セクターに数量の最大値がある。

ソフト制約:

- 期待収益を最大化する。

問題の規模

de_smet_1 has 1 regions, 3 sectors and 11 asset classes with a search space of 10⁴.
 irrinki_1 has 2 regions, 3 sectors and 6 asset classes with a search space of 10³.

サイズが大きいデータセットは作成/検証されていませんが、問題はないはずです。データに関する適切な情報源として、[このアセット関連の Web サイト](#) を参照してください。

9.21. 会議スケジュール

各会議を時間帯と部屋に割り当てていきます。時間帯は重複させることができます。LibreOffice や Excel で編集可能な *.xlsx ファイルとの読み書きが可能です。

ハード制約:

- 時間帯の会議タイプ: 会議のタイプは、時間帯の会議タイプと一致する必要があります。
- 部屋が使用中の時間帯: その会議の時間帯に、会議用の部屋が利用できなければならない。

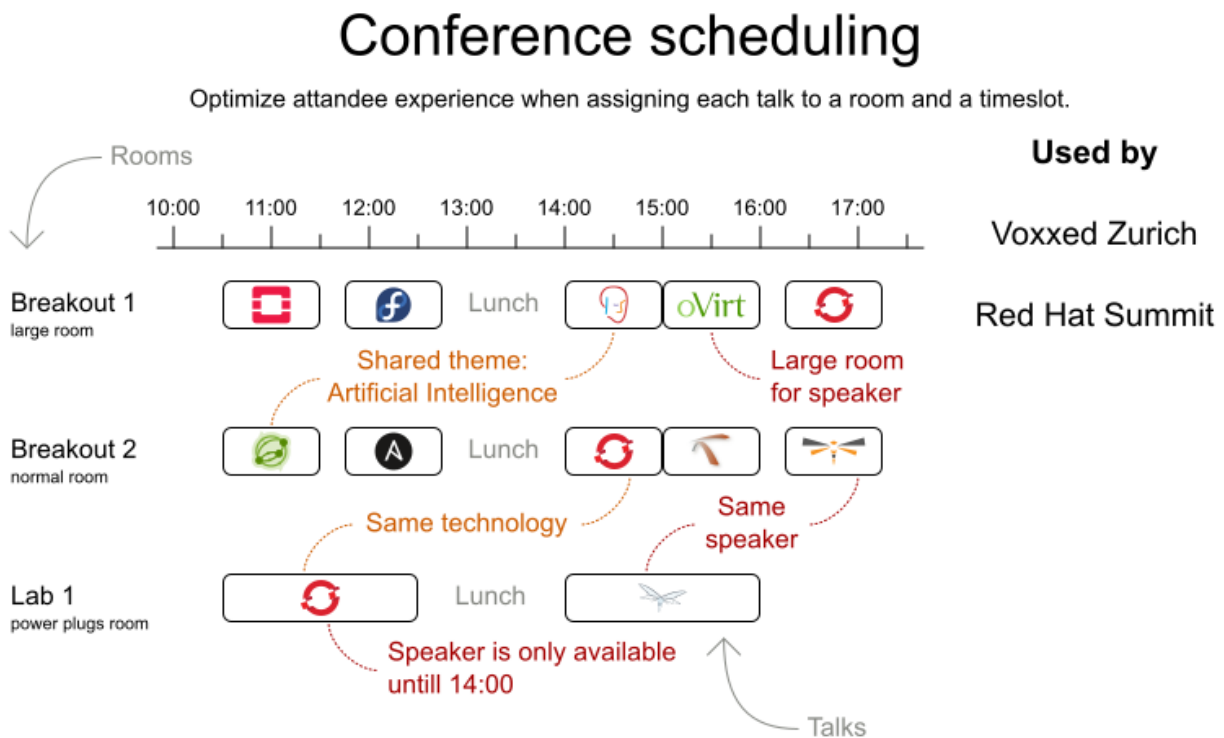
- 部屋の競合: 2つの会議が、同じ時間に同じ会議室を使用することはできない。
- 講演者が空いていない時間帯: 講演者は必ず、会議の時間帯に空いていなければならない。
- 講演者の競合: 同じ時間帯の2つの会議に同じ講演者を割り当てることができない。
- 汎用の時間帯および部屋タグ:
 - 講演者が要求する時間帯タグ: 講演者に、必須時間帯タグが付けられている場合、講演者の会議はそのタグが付いている時間に割り当てる必要がある。
 - 講演者の禁止時間帯タグ: 公演者に、禁止時間帯タグが割り当てられている場合は、そのタグの付いた時間帯に講演者の会議をどれも割り当てることができない。
 - 会議を設定する必要がある時間帯タグ: 会議に必須時間帯タグが付いている場合は、そのタグの付いた時間帯に割り当てる必要がある。
 - 会議の禁止時間帯タグ: 会議に、禁止時間帯タグが割り当てられている場合は、そのタグの付いた時間帯にその会議を割り当てることができない。
 - 講演者が要求する部屋のタグ: 講演者に、必須の部屋タグが付けられている場合、講演者の会議はそのタグが付いている部屋に割り当てる必要がある。
 - 講演者が禁止する部屋のタグ: 講演者に、禁止部屋のタグが付けられている場合、講演者の会議はそのタグが付いている部屋に割り当てることができない。
 - 会議を設定する必要がある部屋タグ: 会議に必須部屋タグが付いている場合は、そのタグの付いた部屋に割り当てる必要がある。
 - 会議の禁止部屋タグ: 会議に、禁止部屋タグが割り当てられている場合は、そのタグの付いた部屋にその会議を割り当てることができない。
- 他の会議と同じ時間帯に設定しないタグ: このタグが付いている会議は、同じ時間帯に重複してスケジュールしてはいけない。
- 受講条件が付いた会議: 受講条件が付いた会議をすべて完了してからでないと対象の会議をスケジュールしてはいけない。

ソフト制約:

- テーマの追跡競合: 同じ時間帯で、テーマのタグが付いた会議の数を最小限に抑える。
- セクターの競合: 同じ時間帯で同じセクタータグの付いた会議の数を最小限に抑える。
- コンテンツの受講者レベルのフロー違反: すべてのコンテンツタグに対して、上級者用の会議の前に入門レベルの会議をスケジュールする。
- 受講者レベルの多様性: すべての時間帯において、異なる受講者レベルの会議数を最大限に増やす。
- 言語の多様性: すべての時間帯において、異なる言語の会議数を最大限を増やす。
- 汎用の時間帯および部屋タグ:
 - 講演者が希望する時間帯タグ: 講演者に、希望の時間帯タグが付けられている場合、講演者の会議はそのタグが付いている時間に割り当てるようにする。

- 講演者が希望しないタイムスロットタグ: 講演者が望ましくないタイムスロットタグを持っている場合、そのタグが付いているタイムスロットには講演を割り当ててはいけません。
- 会議の希望の時間帯タグ: 会議に希望の時間帯タグが付いている場合は、そのタグの付いた時間帯に割り当てるようにする。
- 会議の設定を希望しない時間帯タグ: 会議に、希望しない時間帯タグが付いている場合は、そのタグの付いた時間帯に割り当てないようにする。
- 講演者が希望する部屋のタグ: 講演者に、希望の部屋タグが付けられている場合、講演者の会議はそのタグが付いている部屋に割り当てるようにする。
- 講演者が希望しない部屋のタグ: 講演者に、希望しない部屋タグが付けられている場合、講演者の会議はそのタグが付いている部屋に割り当てないようにする。
- 会議を希望の部屋タグ: 会議に希望の部屋タグが付いている場合は、そのタグの付いた部屋に割り当てないようにする。
- 会議での使用を希望しない部屋タグ: 会議に、希望しない部屋タグが付いている場合、そのタグの付いた部屋に割り当てないようにする。
- 同じ日の会議: テーマタグまたはコンテンツタグを共有する会議は、最低限の日数 (理想的には同じ日) にスケジュールする必要がある。

図9.14 価値提案



問題の規模

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} .

9.22. ロックツアー

次のショーへの移動はロックバンクバスを使用し、空いている日のみショーをスケジュールする。

ハード制約:

- 必要とされるショーをすべてスケジュールする。
- できるだけ多くのショーをスケジュールする。

中程度の制約:

- 収益の機会を最大化する。
- 運転時間を最小限に抑える。
- できるだけ早く到着する。

ソフト制約:

- 長時間の運転は避ける。

問題の規模

47shows has 47 shows with a search space of 10^{59} .

9.23. 航空機乗組員のスケジューリング

パイロットと客室乗務員にフライトを割り当てます。

ハード制約:

- 必須スキル: フライトの割り当てにはそれぞれ、必要とされるスキルがあります。たとえば、フライト AB0001 ではパイロット 2 名と、客室乗務員 3 名が必要です。
- フライトの競合: 各従業員は同じ時間に出勤できるフライトは 1 つだけにします。
- 2 つのフライト間での移動: 2 つのフライトの間で、従業員は到着先の空港と、出発元の空港に移動できる必要がある。たとえば、アンは 10 時にブリュッセルに到着し、15 時にアムステルダムを出発するなどです。
- 従業員の勤務できない日: 従業員はフライトの当日は空いていなければならない。たとえば、アンは 2 月 1 日に休暇を取っているなど。

ソフト制約:

- 最初の仕事が自宅から出発する。

- 最後の仕事が自宅に到着する。
- 総フライト時間を従業員別に平均的に分散する。

問題の規模

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10章 RED HAT ビルドの OPTAPLANNER の例のダウンロード

OptaPlanner サンプルの Red Hat ビルドは、Red Hat カスタマーポータルで入手できる {PRODUCTPAM} アドオンパッケージの一部としてダウンロードできます。

手順

1. Red Hat カスタマーポータルの [Software Downloads](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
 - **製品:** Process Automation Manager
 - **バージョン:** 7.13.4
2. Red Hat Process Automation Manager 7.13 Add Ons をダウンロードします。
3. **rhpam-7.13.4-add-ons.zip** ファイルを展開します。展開した **add-ons** フォルダには、**rhpam-7.13.4-planner-engine.zip** ファイルが含まれます。
4. **rhpam-7.13.4-planner-engine.zip** ファイルを展開します。

結果

展開した **rhpam-7.13.4-planner-engine** ディレクトリーには、以下のサブディレクトリー内にサンプルソースコードのサンプルが含まれます。

- **examples/sources/src/main/java/org/optaplanner/examples**
- **examples/sources/src/main/resources/org/optaplanner/examples**

10.1. OPTAPLANNER サンプルの実行

Red Hat ビルドの OptaPlanner には、さまざまな計画のユースケースのデモとしてサンプルが複数含まれています。この例をダウンロードして使用し、さまざまなタイプのプランニングソリューションを確認します。

前提条件

- [10章 Red Hat ビルドの OptaPlanner の例のダウンロード](#) で記載されているように例をダウンロードしてデプロイメントしている。

手順

1. サンプルを実行するには、**rhpam-7.13.4-planner-engine/examples** ディレクトリーで以下のコマンドのいずれかを入力します。
Linux または Mac の場合:

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

Windows:

```
$ runExamples.bat
```

OptaPlanner Example ウィンドウが開きます。

2. 実行するサンプルを選択します。



注記

Red Hat ビルドの OptaPlanner は GUI に依存しません。デスクトップと同じように、サーバーまたはモバイル JVM 上でも実行できます。

10.2. IDE (INTELLIJ、ECLIPSE、または NETBEANS) での RED HAT ビルドの OPTAPLANNER サンプルの実行

IntelliJ、Eclipse、Netbeans など統合開発環境 (IDE) を使用する場合は、お使いの開発環境にダウンロードした OptaPlanner の例を実行できます。

前提条件

- 10章 *Red Hat ビルドの OptaPlanner の例のダウンロード* の説明に従って、OptaPlanner のサンプルをダウンロードしてデプロイメントしている。

手順

1. OptaPlanner の例を新規プロジェクトとして開きます。
 - a. IntelliJ または Netbeans の場合は、新規プロジェクトとして **examples/sources/pom.xml** を開きます。Maven 統合の指示に従い、インストールを進めてください。この手順では、残りのステップは省略します。
 - b. Eclipse の場合は、**rhpam-7.13.4-planner-engine** ディレクトリーにある **/examples/binaries** ディレクトリーの新規プロジェクトを開きます。
2. **binaries** ディレクトリーにあるすべての JAR ファイルをクラスパスに追加します (**examples/binaries/optaplanner-examples-7.67.0.Final-redhat-00024.jar** ファイルを除く)。
3. **rhpam-7.13.4-planner-engine/examples/sources/** ディレクトリーにある Java ソース **src/main/java** ディレクトリーと Java リソースディレクトリー **src/main/resources** を追加します。
4. 実行設定を作成します。
 - メインクラス: **org.optaplanner.examples.app.OptaPlannerExamplesApp**
 - VM パラメーター (任意): **-Xmx512M -server -Dorg.optaplanner.examples.dataDir=examples/sources/data**
 - 作業ディレクトリー: **examples/sources**
5. 実行設定を実行します。

第11章 BUSINESS CENTRAL での OPTAPLANNER のスタートガイド: 従業員勤務表の例

Business Central で **employee-rostering** のサンプルプロジェクトを構築して、デプロイできます。このプロジェクトは、シフト勤務の計画問題を解決するのに必要な Business Central の各アセットを作成し、Red Hat ビルドの OptaPlanner を使用して実現可能な最適解を見つける方法を示します。

Business Central に事前設定された **employee-rostering** プロジェクトをデプロイすることができます。Business Central を使用してプロジェクトを独自に作成することができます。



注記

Business Central の **employee-rostering** サンプルプロジェクトには、データセットが含まれていません。REST API 呼び出しを使用して XML 形式のデータセットを提供する必要があります。

11.1. BUSINESS CENTRAL への従業員勤務表サンプルプロジェクトのデプロイメント

Business Central には、製品と機能に慣れるために使用できるサンプルプロジェクトが多数あります。従業員の勤務表サンプルプロジェクトは、Red Hat ビルドの OptaPlanner でシフト勤務のユースケースを示すために計画され作成されました。以下の手順に従って、従業員の勤務表サンプルを Business Central にデプロイして実行します。

前提条件

- Red Hat Decision Manager をダウンロードしてインストールしている。インストールオプションについては、[Planning a Red Hat Decision Manager installation](#) を参照してください。
- インストールのドキュメントにあるように、Red Hat Decision Manager が起動し、**admin** パーミッションを持つユーザーとして Business Central にログインしています。

手順

1. Business Central で **Menu** → **Design** → **Projects** の順にクリックします。
2. 事前に設定した **MySpace** スペースで **Try Samples** をクリックします。
3. サンプルプロジェクトのリストから **employee-rostering** を選択し、右上の **OK** をクリックして、プロジェクトをインポートします。
4. アセットリストをコンパイルし、**Build & Deploy** をクリックして、従業員の勤務表サンプルをデプロイします。

本書は、各プロジェクトアセットとその設定について説明します。

11.2. 従業員の勤務表サンプルプロジェクトの再作成

従業員の勤務表サンプルプロジェクトは、Business Central で使用できる事前設定のプロジェクトです。このプロジェクトをデプロイする方法は、[「Business Central への従業員勤務表サンプルプロジェクトのデプロイメント」](#) を参照してください。

ゼロから従業員勤務表を作成できます。この例では、ワークフローを使用して、Business Central で独自の類似プロジェクトを作成します。

11.2.1. 従業員の勤務表プロジェクトの設定

Business Central で Solver の開発を始めるには、プロジェクトを設定する必要があります。

前提条件

- Red Hat Decision Manager をダウンロードしてインストールしている。
- Business Central をデプロイし、**admin** ロールを持つユーザーでログインしている。

手順

1. **Menu** → **Design** → **Projects** → **Add Project** をクリックして、Business Central に新しいプロジェクトを作成します。
2. **Add Project** ウィンドウで、以下のフィールドに入力します。
 - **Name:** **employee-rostering**
 - **Description** (任意): OptaPlanner を使用した従業員の勤務表問題の最適化。スキルに基づいて、従業員をシフトに割り当てます。

任意で、**Configure Advanced Options** をクリックして、**Group ID**、**Artifact ID**、および **Version** に情報を追加します。

- **Group ID:** **employeerostering**
 - **Artifact ID:** **employeerostering**
 - **Version:** **1.0.0-SNAPSHOT**
3. **Add** をクリックして、Business Central プロジェクトリポジトリにプロジェクトを追加します。

11.2.2. プロジェクトファクトおよびプランニングエンティティ

従業員勤務表の計画問題の各ドメインクラスは、以下のいずれかに分類されます。

- **関連性のないクラス:** どのスコア制約にも使用されません。計画に関して言えば、このデータは使用されません。
- **問題ファクト** クラス: スコア制約に使用されますが、(問題が変わらない限り) 計画時には変化しません (例: **Shift**、**Employee**)。問題ファクトクラスのプロパティはすべて問題のプロパティです。
- **プランニングエンティティ** クラス: スコア制約に使用され、計画時に変化します (例: **ShiftAssignment**)。計画時に変更するプロパティは **プランニング変数** です。その他のプロパティは問題プロパティです。
以下の点についてお考え下さい。
 - 計画時にどのクラスを変更しますか？
 - Solver で変更する変数はどのクラスにありますか？

そのクラスが、プランニングエンティティです。

プランニングエンティティークラスは、**@PlanningEntity** アノテーションでアノテートする必要があります。または、ドメインデザイナーで Red Hat ビルドの OptaPlanner ドックを使用して Business Central に定義する必要があります。

各プランニングエンティティークラスには、1つ以上の **プランニング変数** があり、1つ以上の定義プロパティが必要です。

多くのユースケースには、プランニングエンティティークラスが1つだけあり、1つのプランニングエンティティークラスに対してプランニング変数が1つだけ含まれます。

11.2.3. 従業員の勤務表プロジェクトへのデータモデルの作成

このセクションでは、Business Central で従業員の勤務表サンプルプロジェクトを実行するのに必要なデータオブジェクトを作成します。

前提条件

- 「[従業員の勤務表プロジェクトの設定](#)」に従ってプロジェクト設定が完了している。

手順

1. 新規プロジェクトで、プロジェクトパースペクティブの **Data Object** をクリックするか、**Add Asset → Data Object** をクリックして、新しいデータオブジェクトを作成します。
2. 最初のデータオブジェクトの名前を **Timeslot** とし、**パッケージ** で **employeerostring.employeerostring** を選択します。
OK をクリックします。
3. **Data Objects** パースペクティブで **+add field** をクリックして、**Timeslot** データオブジェクトにフィールドを追加します。
4. **id** フィールドで **endTime** と入力します。
5. **Type** の横にあるドロップダウンメニューをクリックし、**LocalDateTime** を選択します。
6. **Create and continue** を別のフィールドに追加します。
7. **id** **startTime** および **Type** **LocalDateTime** を使用して、フィールドを追加します。
8. **Create** をクリックします。
9. 右上の **Save** をクリックして、**Timeslot** データオブジェクトを保存します。
10. 右上の **x** をクリックして、**Data Objects** パースペクティブを閉じ、**Assets** メニューに戻ります。
11. 前述の手順で、以下のデータオブジェクトとその属性を作成します。

表11.1 Skill

id	タイプ
name	String

表11.2 Employee

id	タイプ
name	String
skills	employeerostering.employeerostering.Skill[List]

表11.3 Shift

id	タイプ
requiredSkill	employeerostering.employeerostering.Skill
timeslot	employeerostering.employeerostering.Timeslot

表11.4 DayOffRequest

id	タイプ
date	LocalDate
employee	employeerostering.employeerostering.Employee

表11.5 ShiftAssignment

id	タイプ
employee	employeerostering.employeerostering.Employee
shift	employeerostering.employeerostering.Shift

データオブジェクトの作成例は [デシジョンサービスの使用ガイド](#) を参照してください。

11.2.3.1. 従業員の勤務表プランニングエンティティの作成


従業員勤務表の計画問題を解決するには、プランニングエンティティと Solver を作成する必要があります。プランニングエンティティは、Red Hat ビルドの OptaPlanner ドックで利用可能な属性を使用して、ドメインデザイナーに定義します。

以下の手順に従って、従業員の勤務表サンプルに、**ShiftAssignment** データオブジェクトをプランニングエンティティとして定義します。

前提条件

- 従業員の勤務表サンプルを実行するには、「[従業員の勤務表プロジェクトへのデータモデルの作成](#)」の手順に従って、関連するデータオブジェクトとプランニングエンティティを作成する必要があります。

手順

1. プロジェクトの **Assets** メニューから、**ShiftAssignment** データオブジェクトを開きます。
2. **Data Objects** パースペクティブで、右側の  をクリックして、OptaPlanner のドックを開きます。
3. **Planning Entity** を選択します。
4. **ShiftAssignment** データオブジェクトのフィールドリストで **employee** を選択します。
5. OptaPlanner ドックで **Planning Variable** を選択します。
Value Range Id 入力フィールドに **employeeRange** を入力します。これにより、**@ValueRangeProvider** アノテーションがプランニングエンティティに追加され、デザイナーの **Source** タブをクリックすると表示されます。

プランニング変数の値の範囲は **@ValueRangeProvider** アノテーションで定義されます。**@ValueRangeProvider** アノテーションには **id** プロパティが常にあり、**@PlanningVariable** の **valueRangeProviderRefs** プロパティから参照されます。

6. ドックを閉じ、**Save** をクリックして、データオブジェクトを保存します。

11.2.3.2. 従業員の勤務表プランニングソリューションの作成

従業員勤務表の問題は、定義したプランニングソリューションに依存します。プランニングソリューションは、Red Hat ビルドの OptaPlanner ドックで利用可能な属性を使用して、ドメインデザイナーに定義します。

前提条件

- 「[従業員の勤務表プロジェクトへのデータモデルの作成](#)」 および 「[従業員の勤務表プランニングエンティティの作成](#)」 の手順に従って、従業員の勤務表サンプルを実行するのに必要なデータオブジェクトおよびプランニングエンティティを作成している。


手順

1. 識別子 **EmployeeRoster** でデータオブジェクトを新規作成します。
2. 以下のフィールドを作成します。

表11.6 EmployeeRoster

id	タイプ
dayOffRequestList	employeerostering.employeerostering.DayOffRequest[List]

id	タイプ
shiftAssignmentList	employee rostering.employee rostering.ShiftAssignment[List]
shiftList	employee rostering.employee rostering.Shift[List]
skillList	employee rostering.employee rostering.Skill[List]
timeslotList	employee rostering.employee rostering.Timeslot[List]

3. **Data Objects** パースペクティブで、右側の  をクリックして、OptaPlanner のドックを開きます。
4. **Planning Solution** を選択します。
5. **Solution Score Type** は、デフォルトの **Hard soft score** のままにします。これにより、タイプがソリューションスコアとなる **EmployeeRoster** データオブジェクトに、**score** フィールドが自動的に生成されます。
6. 次の属性で新しいフィールドを追加します。

id	タイプ
employeeList	employee rostering.employee rostering.Employee[List]

7. **employeeList** フィールドを選択した状態で、Red Hat ビルドの OptaPlanner ドックを開いて、**Planning Value Range Provider** ボックスを選択します。
id フィールドに **employeeRange** を入力します。ドックを閉じます。
8. 右上で **Save** をクリックし、アセットを保存します。

11.2.4. 従業員勤務表の制約

従業員の勤務表はプランニングソリューションです。すべての計画問題には、最適解を得るのに満たさなければならない制約が含まれます。

Business Central の従業員の勤務表サンプルプロジェクトには、以下のハード制約およびソフト制約が含まれます。

ハード制約

- 従業員に割り当てられるシフトの数は、1日1つまで。
- 特別な従業員スキルが必要なすべてのシフトは、そのスキルを持つ従業員に割り当てられる。

ソフト制約

- すべての従業員がシフトに割り当てられている。
- 従業員が休暇を取った場合は、シフトが別の従業員に再割り当てされる。

ハード制約およびソフト制約は、Free form DRL デザイナー、またはガイド付きルールを使用して Business Central で定義します。

11.2.4.1. DRL (Drools Rule Language) ルール

DRL (Drools Rule Language) ルールは、**.drl** テキストファイルに直接定義するビジネスルールです。このような DRL ファイルは、Business Central の他のすべてのルールアセットが最終的にレンダリングされるソースとなります。Business Central インターフェイスで DRL ファイルを作成して管理するか、Red Hat CodeReady Studio や別の統合開発環境 (IDE) を使用して Maven または Java プロジェクトの一部として外部で作成することができます。DRL ファイルには、最低でもルールの条件 (**when**) およびアクション (**then**) を定義するルールを1つ以上追加できます。Business Central の DRL デザイナーでは、Java、DRL、および XML の構文が強調表示されます。

DRL ファイルは、以下のコンポーネントで設定されます。

DRL ファイル内のコンポーネント

```
package

import

function // Optional

query // Optional

declare // Optional

global // Optional

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

rule "rule2 name"

...
```

以下の DRL ルールの例では、ローン申し込みのデシジョンサービスで年齢制限を指定します。

申込者の年齢制限に関するルールの例

```
rule "Underage"
  salience 15
  agenda-group "applicationGroup"
```

```

when
  $application : LoanApplication()
  Applicant( age < 21 )
then
  $application.setApproved( false );
  $application.setExplanation( "Underage" );
end

```

DRL ファイルには、ルール、クエリー、関数が1つまたは複数含まれており、このファイルで、ルールやクエリーで割り当て、使用するインポート、グローバル、属性などのリソース宣言を定義できます。DRL パッケージは、DRL ファイルの一番上に表示され、ルールは通常最後に表示されます。他の DRL コンポーネントはどのような順番でも構いません。

ルールごとに、ルールパッケージ内で一意の名前を指定する必要があります。パッケージ内の DRL ファイルで、同じルール名を複数回使用すると、ルールのコンパイルに失敗します。特にルール名にスペースを使用する場合など、ルール名には必ず二重引用符 (**rule "rule name"**) を使用して、コンパイルエラーが発生しないようにしてください。

DRL ルールに関連するデータオブジェクトはすべて、DRL ファイルと同じ Business Central プロジェクトパッケージに置く必要があります。同じパッケージに含まれるアセットはデフォルトでインポートされます。その他のパッケージの既存アセットは、DRL ルールを使用してインポートできます。

11.2.4.2. DRL デザイナーを使用した従業員の勤務表の制約定義

Business Central で Free form DRL デザイナーを使用して、従業員の勤務表サンプルに制約の定義を作成できます。

この手順を使用して、シフトが終わってから 10 時間以上経たないと従業員をシフトに割り当てられない **ハード制約** を作成します。

手順

1. Business Central で、**Menu** → **Design** → **Projects** に移動して、プロジェクト名をクリックします。
2. **Add Asset** → **DRL ファイル** の順にクリックします。
3. **DRL file** 名前フィールドに、**ComplexScoreRules** と入力します。
4. **employee rostering.employee rostering** パッケージを選択します。
5. **+OK** をクリックして DRL ファイルを作成します。
6. DRL デザイナーの **Model** タブで、**Employee10HourShiftSpace** ルールを DRL ファイルとして定義します。

```

package employee rostering.employee rostering;

rule "Employee10HourShiftSpace"
  when
    $shiftAssignment : ShiftAssignment( $employee : employee != null, $shiftEndTime :
shift.timeslot.endTime)
    ShiftAssignment( this != $shiftAssignment, $employee == employee, $shiftEndTime
<= shift.timeslot.endTime,
    $shiftEndTime.until(shift.timeslot.startTime,
java.time.temporal.ChronoUnit.HOURS) <10)

```



```

then
scoreHolder.addHardConstraintMatch(kcontext, -1);
end

```

7. **Save** をクリックして、DRL ファイルを保存します。

DRL ファイルの作成方法は [DRL ルールを使用したデシジョンサービスの作成](#) を参照してください。

11.2.5. ガイド付きルールを使用して従業員の勤務表にルールの作成

Business Central でガイド付きルールデザイナーを使用して、従業員の勤務表にハード制約およびソフト制約を定義するルールを作成できます。

11.2.5.1. ガイド付きルール

ガイド付きルールは、ルール作成のプロセスを提供する、Business Central の UI ベースのガイド付きルールデザイナーで作成するビジネスルールです。ガイド付きルールデザイナーを使用すると、ルールを定義するデータオブジェクトに基づいて、可能なインプットにフィールドおよびオプションを提供します。定義したガイド付きルールは、その他のすべてのルールアセットとともに Drools Rule Language (DRL) ルールにコンパイルされます。

ガイド付きルールに関連するすべてのデータオブジェクトは、ガイド付きルールと同じプロジェクトパッケージに置く必要があります。同じパッケージに含まれるアセットはデフォルトでインポートされます。必要なデータオブジェクトとガイド付きルールを作成したら、ガイド付きルールデザイナーの **Data Objects** タブから、必要なデータオブジェクトがすべてリストされていることを検証したり、**新規アイテム** を追加してその他の既存データオブジェクトをインポートしたりできます。




11.2.5.2. 従業員のシフト数のバランスを取るガイド付きルールの作成

ガイド付きルール **BalanceEmployeesShiftNumber** は、可能な限りバランスを取るよう従業員にシフトを割り当てるソフト制約を作成します。これは、シフトの分配が平等でなくなると増えるスコアペナルティを作成することで行います。ルールによって実装されたスコア式により、Solver がよりバランスの取れるようにシフトを分散させます。

The screenshot shows the 'BalanceEmployeesShiftNumber.rdr1 - Guided Rules' editor. It has tabs for 'Editor', 'Overview', 'Source', and 'Data Objects'. The 'Editor' tab is active, showing the rule configuration. At the top, there are buttons for 'Save', 'Delete', 'Rename', 'Copy', 'Validate', and 'Latest Version'. Below the tabs, there is a dropdown menu for 'EXTENDS' set to '- None -'. The 'WHEN' section contains two conditions: 1. 'There is an Employee [\$employee]' and 'There is a Number [\$shiftCount]'. Below these, it says 'From Accumulate All ShiftAssignment [\$shiftAssignment] with:'. 2. 'employee equal to \$employee'. There are options for 'Custom Code' and 'Function'. The 'THEN' section contains one action: 'Soft Score' with the formula '-(\$shiftCount.intValue() * \$shiftCount.intValue())'. At the bottom, there is a 'Messages' section with a 'Clear' button.

手順

1. Business Central で、**Menu** → **Design** → **Projects** に移動して、プロジェクト名をクリックします。
2. **Add Asset** → **Guided Rule** の順にクリックします。

3. Guided Rule 名に **BalanceEmployeesShiftNumber** と入力し、Package で **employee rostering.employee rostering** を選択します。
4. OK をクリックして、ルールアセットを作成します。
5. WHEN フィールドで  をクリックして、WHEN 条件を追加します。
6. Add a condition to the rule ウィンドウで **Employee** を選択します。+OK をクリックします。
7. **Employee** 条件でクリックして制約を修正し、変数名 **\$employee** を追加します。
8. WHEN 条件 **From Accumulate** を追加します。
 - a. **From Accumulate** 条件の上で click to add pattern をクリックし、ドロップダウンリストでファクトタイプ **Number** を選択します。
 - b. 変数名 **\$shiftCount** を **Number** 条件に追加します。
 - c. **From Accumulate** 条件の下で click to add pattern をクリックして、ドロップダウンリストで **ShiftAssignment** ファクトタイプを選択します。
 - d. 変数名 **\$shiftAssignment** を **ShiftAssignment** ファクトタイプに追加します。
 - e. **ShiftAssignment** 条件を再度クリックし、Add a restriction on a field ドロップダウンリストで **employee** を選択します。
 - f. **employee** 制約の横にあるドロップダウンリストで **equal to** を選択します。
 - g. ドロップダウンボタンの横の  アイコンをクリックして変数を追加し、Field value ウィンドウで **Bound variable** をクリックします。
 - h. ドロップダウンリストで **\$employee** を選択します。
 - i. Function ボックスに **count (\$shiftAssignment)** と入力します。
9. THEN フィールドで  をクリックして、THEN 条件を追加します。
10. Add a new action ウィンドウで **Modify Soft Score** を選択します。+OK をクリックします。
 - a. ボックスに **- (\$shiftCount.intValue()*\$shiftCount.intValue())** と入力します。
11. 右上隅の **Validate** をクリックし、ルール条件がすべて有効であることを確認します。ルールの妥当性確認に失敗したら、エラーメッセージに記載された問題に対応し、ルールの全コンポーネントを見直し、エラーが表示されなくなるまでルールの妥当性確認を行います。
12. **Save** をクリックして、ルールを保存します。

ガイド付きルールの作成方法は [ガイド付きルールを使用したデジジョンサービスの作成](#) を参照してください。

11.2.5.3. 同じ日に複数のシフトを割り当てないようにするガイド付きルールの作成

ガイド付きルール **OneEmployeeShiftPerDay** は、同じ日の複数のシフトに従業員を割り当てないようにするハード制約を作成します。従業員の勤務表サンプルでは、この制約はガイド付きルールデザイナーを使用して作成されます。

手順

1. Business Central で、**Menu → Design → Projects** に移動して、プロジェクト名をクリックします。
2. **Add Asset → Guided Rule** の順にクリックします。
3. **Guided Rule** 名に **OneEmployeeShiftPerDay** と入力し、**Package** で **employee rostering.employee rostering** を選択します。
4. **OK** をクリックして、ルールアセットを作成します。
5. **WHEN** フィールドで **+** をクリックして、**WHEN** 条件を追加します。
6. **Add a condition to the rule** ウィンドウから **Free form DRL** を選択します。
7. Free form の DRL ボックスに、以下の条件を入力します。

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

この条件は、同じ日に別のシフトがすでに割り当てられている従業員にはシフトを割り当てることができないことを示しています。

8. **THEN** フィールドで **+** をクリックして、**THEN** 条件を追加します。
 9. **Add a new action** ウィンドウから **Add Free form DRL** を選択します。
 10. Free form の DRL ボックスに、以下の条件を入力します。
- ```
scoreHolder.addHardConstraintMatch(kcontext, -1);
```
11. 右上隅の **Validate** をクリックし、ルール条件がすべて有効であることを確認します。ルールの妥当性確認に失敗したら、エラーメッセージに記載された問題に対応し、ルールの全コンポーネントを見直し、エラーが表示されなくなるまでルールの妥当性確認を行います。
  12. **Save** をクリックして、ルールを保存します。



ガイド付きルールの作成方法は [ガイド付きルールを使用したデジモンサービスの作成](#) を参照してください。

#### 11.2.5.4. シフト要件にスキルを一致させるガイド付きルールを作成

ガイド付きルール **ShiftRequiredSkillsAreMet** は、すべてのシフトが、適切なスキルセットを持つ従業員に割り当てられるのを確認するハード制約を作成します。従業員の勤務表サンプルでは、この制約はガイド付きルールデザイナーを使用して作成されます。

#### 手順

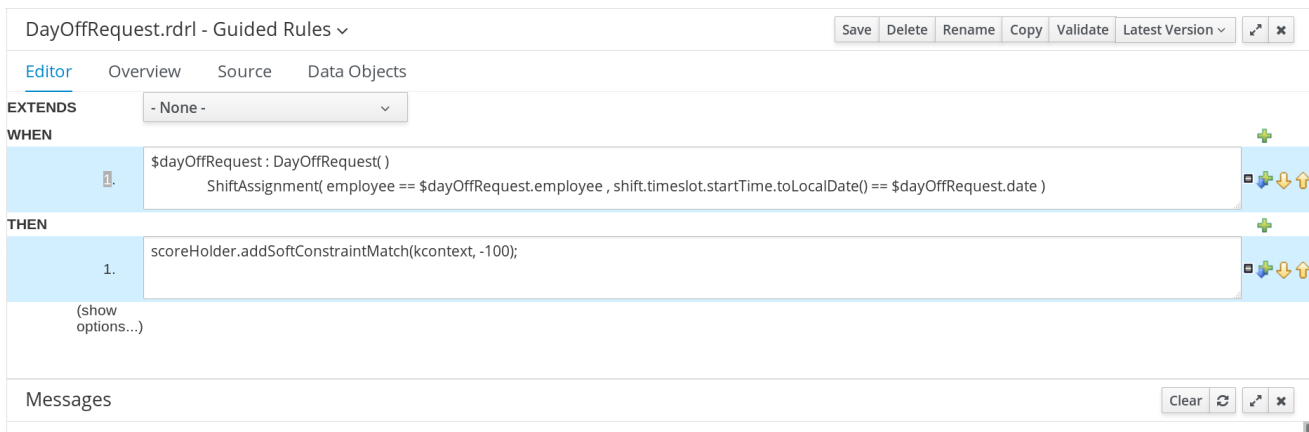
1. Business Central で、**Menu → Design → Projects** に移動して、プロジェクト名をクリックします。
2. **Add Asset → Guided Rule** の順にクリックします。
3. **Guided Rule** 名に **ShiftRequiredSkillsAreMet** と入力し、**Package** で **employee rostering.employee rostering** を選択します。
4. **OK** をクリックして、ルールアセットを作成します。
5. **WHEN** フィールドで **+** をクリックして、**WHEN** 条件を追加します。
6. **Add a condition to the rule** ウィンドウで **ShiftAssignment** を選択します。 **+OK** をクリックします。
7. **ShiftAssignment** 条件をクリックし、**Add a restriction on a field** ドロップダウンリストで **employee** を選択します。
8. デザイナーで、**employee** の横のドロップダウンリストをクリックし、**is not null** を選択します。
9. **ShiftAssignment** 条件をクリックし、**Expression editor** をクリックします。
  - a. デザイナーで、**[not bound]** をクリックし、**Expression editor** を開き、式と変数 **\$requiredSkill** をバインドします。 **Set** をクリックします。
  - b. デザイナーの **\$requiredSkill** の横にあるドロップダウンリストで **shift** を選択し、その隣のドロップダウンリストで **requiredSkill** を選択します。
10. **ShiftAssignment** 条件をクリックし、**Expression editor** をクリックします。
  - a. デザイナーで、**[not bound]** の横にあるドロップダウンリストで **employee** を選択し、その隣のドロップダウンリストで **skills** を選択します。
  - b. その隣のドロップダウンリストでは **Choose** を選択したままにします。

- c. その隣のドロップダウンボックスで、**please choose** を **excludes** に変更します。
  - d. **excludes** の横にある  アイコンをクリックし、Field value ウィンドウで **New formula** ボタンをクリックします。
  - e. 式ボックスに **\$requiredSkill** を追加します。
11. **THEN** フィールドで  をクリックして、**THEN** 条件を追加します。
  12. **Add a new action** ウィンドウで **Modify Hard Score** を選択します。+OK をクリックします。
  13. スコアアクションボックスに **-1** を入力します。
  14. 右上隅の **Validate** をクリックし、ルール条件がすべて有効であることを確認します。ルールの妥当性確認に失敗したら、エラーメッセージに記載された問題に対応し、ルールの全コンポーネントを見直し、エラーが表示されなくなるまでルールの妥当性確認を行います。
  15. **Save** をクリックして、ルールを保存します。

ガイド付きルールの作成方法は [ガイド付きルールを使用したデジジョンサービスの作成](#) を参照してください。

#### 11.2.5.5. 休暇申請を管理するガイド付きルールの作成

ガイド付きルール **DayOffRequest** は、ソフト制約を作成します。この制約では、そのシフトに元々割り当てられていた従業員がその日に就業できなくなった場合に、別の従業員にシフトを再割り当てできるようにできます。従業員の勤務表サンプルでは、この制約はガイド付きルールデザイナーを使用して作成されます。



#### 手順


1. Business Central で、**Menu** → **Design** → **Projects** に移動して、プロジェクト名をクリックします。
2. **Add Asset** → **Guided Rule** の順にクリックします。
3. **Guided Rule** 名に **DayOffRequest** と入力し、**Package** で **employee rostering.employee rostering** を選択します。
4. **OK** をクリックして、ルールアセットを作成します。
5. **WHEN** フィールドで  をクリックして、**WHEN** 条件を追加します。

6. **Add a condition to the rule** ウィンドウから **Free form DRL** を選択します。

7. Free form の DRL ボックスに、以下の条件を入力します。

```
$dayOffRequest : DayOffRequest()
 ShiftAssignment(employee == $dayOffRequest.employee ,
 shift.timeslot.startTime.toLocalDate() == $dayOffRequest.date)
```

この条件は、休暇申請を行った従業員にシフトを割り当てている場合に、その従業員をその日のシフト割り当てから削除できることを示しています。

8. **THEN** フィールドで  をクリックして、**THEN** 条件を追加します。

9. **Add a new action** ウィンドウから **Add Free form DRL** を選択します。

10. Free form の DRL ボックスに、以下の条件を入力します。

```
scoreHolder.addSoftConstraintMatch(kcontext, -100);
```

11. 右上隅の **Validate** をクリックし、ルール条件がすべて有効であることを確認します。ルールの妥当性確認に失敗したら、エラーメッセージに記載された問題に対応し、ルールの全コンポーネントを見直し、エラーが表示されなくなるまでルールの妥当性確認を行います。

12. **Save** をクリックして、ルールを保存します。

ガイド付きルールの作成方法は [ガイド付きルールを使用したデジジョンサービスの作成](#) を参照してください。

### 11.2.6. 従業員の勤務表の Solver 設定の作成

Business Central に Solver 設定を作成して編集できます。Solver 設定デザイナーは、プロジェクトがデプロイされた後に実行できる Solver 設定を作成します。

#### 前提条件

- Red Hat Decision Manager をダウンロードしてインストールしている。
- 従業員の勤務表サンプルに関連するアセットをすべて作成して設定している。

#### 手順

1. Business Central で **Menu** → **Projects** をクリックし、使用するプロジェクトをクリックして開きます。
2. **Assets** パースペクティブで、**Add Asset** → **Solver configuration** をクリックします。
3. **Create new Solver configuration** ウィンドウで、Solver の名前 **EmployeeRosteringSolverConfig** と入力し、**Ok** をクリックします。これにより、**Solver configuration** デザイナーが開きます。
4. **Score Director Factory** 設定セクションで、スコアリングルール定義を含む KIE ベースを定義します。従業員の勤務表サンプルプロジェクトは **defaultKieBase** を使用します。
  - a. KIE ベースに定義した KIE セッションの中から1つ選択します。従業員の勤務表サンプルプロジェクトは **defaultKieSession** を使用します。

5. 右上の **Validate** をクリックし、**Score Director Factory** 設定が正しいことを確認します。妥当性確認に失敗したら、エラーメッセージに記載された問題に対応し、エラーが表示されなくなるまで妥当性確認を行います。
6. **Save** をクリックして、Solver 設定を保存します。

### 11.2.7. 従業員の勤務表プロジェクトに対する Solver の終了設定

一定期間が過ぎたら Solver が終了できるように設定できます。デフォルトでは、プランニングエンジンには、時間制限なく問題を解決できるように指定されています。

従業員の勤務表サンプルプロジェクトは、30 秒間実行するように設定されています。

#### 前提条件

- 従業員の勤務表プロジェクトに、関連するすべてのアセットを作成し、[「従業員の勤務表の Solver 設定の作成」](#) の手順に従って、Business Central に Solver 設定 **EmployeeRosteringSolverConfig** を作成している。

#### 手順

1. **Assets** パースペクティブで **EmployeeRosteringSolverConfig** を開きます。これにより、**Solver 設定 デザイナー**が開きます。
2. **Termination** セクションで **Add** をクリックして、選択した論理グループに新しい終了要素を作成します。
3. ドロップダウンリストから、終了タイプ **Time spent** を選択します。これは、終了条件の入力フィールドとして追加されます。
4. 時間要素の横の矢印を使用して、経過時間を 30 秒に設定します。
5. 右上の **Validate** をクリックし、**Score Director Factory** 設定が正しいことを確認します。妥当性確認に失敗したら、エラーメッセージに記載された問題に対応し、エラーが表示されなくなるまで妥当性確認を行います。
6. **Save** をクリックして、Solver 設定を保存します。

## 11.3. REST API を使用した SOLVER へのアクセス

サンプルの Solver をデプロイするか再作成したら、REST API を使用してアクセスできます。

REST API を使用して Solver インスタンスを登録する必要があります。その後に、データセットを指定して、最適解を取得できます。

#### 前提条件

- 本書の以前のセクションに従い、従業員の勤務表プロジェクトを設定し、デプロイしてください。[「従業員の勤務表サンプルプロジェクトの再作成」](#) に記載のとおり、同じプロジェクトをデプロイするか、[「Business Central への従業員勤務表サンプルプロジェクトのデプロイメント」](#) に記載のとおり、プロジェクトを再作成できます。

### 11.3.1. REST API を使用した Solver の登録

Solver を使用するには、REST API を使用して Solver インスタンスを登録する必要があります。

各 Solver インスタンスで、一度に最適化できる計画問題の数は1つだけです。

## 手順

1. 以下のヘッダーを使用して HTTP 要求を作成します。

```
authorization: admin:admin
X-KIE-ContentType: xstream
content-type: application/xml
```

2. 以下の要求を使用して Solver を登録します。

### PUT

```
http://localhost:8080/kie-
server/services/rest/server/containers/employeerostering_1.0.0-
SNAPSHOT/solvers/EmployeeRosteringSolver
```

### 要求ボディ

```
<solver-instance>
 <solver-config-
file>employeerostering/employeerostering/EmployeeRosteringSolverConfig.solver.xml</s
olver-config-file>
</solver-instance>
```

## 11.3.2. REST API を使用した Solver の呼び出し

Solver インスタンスを登録した後に、REST API を使用して、データセットを Solver に送信して、最適解を取得できます。

## 手順

1. 以下のヘッダーを使用して HTTP 要求を作成します。

```
authorization: admin:admin
X-KIE-ContentType: xstream
content-type: application/xml
```

2. 以下の例のように、データセットを含む Solver に要求を送信します。

### POST

```
http://localhost:8080/kie-
server/services/rest/server/containers/employeerostering_1.0.0-
SNAPSHOT/solvers/EmployeeRosteringSolver/state/solving
```

### 要求ボディ

```
<employeerostering.employeerostering.EmployeeRoster>
 <employeeList>
 <employeerostering.employeerostering.Employee>
 <name>John</name>
 <skills>
 <employeerostering.employeerostering.Skill>
 <name>reading</name>
```



```

 </employee rostering.employee rostering.Skill>
 </skills>
</employee rostering.employee rostering.Employee>
<employee rostering.employee rostering.Employee>
 <name>Mary</name>
 <skills>
 <employee rostering.employee rostering.Skill>
 <name>writing</name>
 </employee rostering.employee rostering.Skill>
 </skills>
</employee rostering.employee rostering.Employee>
<employee rostering.employee rostering.Employee>
 <name>Petr</name>
 <skills>
 <employee rostering.employee rostering.Skill>
 <name>speaking</name>
 </employee rostering.employee rostering.Skill>
 </skills>
</employee rostering.employee rostering.Employee>
</employeeList>
<shiftList>
 <employee rostering.employee rostering.Shift>
 <timeslot>
 <startTime>2017-01-01T00:00:00</startTime>
 <endTime>2017-01-01T01:00:00</endTime>
 </timeslot>
 <requiredSkill
reference="../../../../employeeList/employee rostering.employee rostering.Employee/skills/emp
loyerostering.employee rostering.Skill"/>
 </employee rostering.employee rostering.Shift>
<employee rostering.employee rostering.Shift>
 <timeslot reference="../../../../employee rostering.employee rostering.Shift/timeslot"/>
 <requiredSkill
reference="../../../../employeeList/employee rostering.employee rostering.Employee[3]/skills/emp
loyerostering.employee rostering.Skill"/>
 </employee rostering.employee rostering.Shift>
<employee rostering.employee rostering.Shift>
 <timeslot reference="../../../../employee rostering.employee rostering.Shift/timeslot"/>
 <requiredSkill
reference="../../../../employeeList/employee rostering.employee rostering.Employee[2]/skills/emp
loyerostering.employee rostering.Skill"/>
 </employee rostering.employee rostering.Shift>
</shiftList>
<skillList>
 <employee rostering.employee rostering.Skill
reference="../../../../employeeList/employee rostering.employee rostering.Employee/skills/emplo
yerostering.employee rostering.Skill"/>
 <employee rostering.employee rostering.Skill
reference="../../../../employeeList/employee rostering.employee rostering.Employee[3]/skills/emplo
yerostering.employee rostering.Skill"/>
 <employee rostering.employee rostering.Skill
reference="../../../../employeeList/employee rostering.employee rostering.Employee[2]/skills/emplo
yerostering.employee rostering.Skill"/>
</skillList>
<timeslotList>
 <employee rostering.employee rostering.Timeslot

```

```

reference="../../shiftList/employee rostering.employee rostering.Shift/timeslot"/>
</timeslotList>
<dayOffRequestList/>
<shiftAssignmentList>
 <employee rostering.employee rostering.ShiftAssignment>
 <shift reference="../../shiftList/employee rostering.employee rostering.Shift"/>
 </employee rostering.employee rostering.ShiftAssignment>
 <employee rostering.employee rostering.ShiftAssignment>
 <shift reference="../../shiftList/employee rostering.employee rostering.Shift[3]"/>
 </employee rostering.employee rostering.ShiftAssignment>
 <employee rostering.employee rostering.ShiftAssignment>
 <shift reference="../../shiftList/employee rostering.employee rostering.Shift[2]"/>
 </employee rostering.employee rostering.ShiftAssignment>
</shiftAssignmentList>
</employee rostering.employee rostering.EmployeeRoster>

```

3. 計画問題に最適解をリクエストします。

#### GET

**http://localhost:8080/kie-server/services/rest/server/containers/employee rostering\_1.0.0-SNAPSHOT/solvers/EmployeeRosteringSolver/bestsolution**

#### 応答例

```

<solver-instance>
 <container-id>employee-rostering</container-id>
 <solver-id>solver1</solver-id>
 <solver-config-
file>employee rostering/employee rostering/EmployeeRosteringSolverConfig.solver.xml</s
olver-config-file>
 <status>NOT_SOLVING</status>
 <score
scoreClass="org.optaplanner.core.api.score.buildin.hardsoft.HardSoftScore">0hard/0soft<
/score>
 <best-solution class="employee rostering.employee rostering.EmployeeRoster">
 <employeeList>
 <employee rostering.employee rostering.Employee>
 <name>John</name>
 <skills>
 <employee rostering.employee rostering.Skill>
 <name>reading</name>
 </employee rostering.employee rostering.Skill>
 </skills>
 </employee rostering.employee rostering.Employee>
 <employee rostering.employee rostering.Employee>
 <name>Mary</name>
 <skills>
 <employee rostering.employee rostering.Skill>
 <name>writing</name>
 </employee rostering.employee rostering.Skill>
 </skills>
 </employee rostering.employee rostering.Employee>
 <employee rostering.employee rostering.Employee>
 <name>Petr</name>

```

```

<skills>
 <employee rostering.employee rostering.Skill>
 <name>speaking</name>
 </employee rostering.employee rostering.Skill>
</skills>
</employee rostering.employee rostering.Employee>
</employeeList>
<shiftList>
 <employee rostering.employee rostering.Shift>
 <timeslot>
 <startTime>2017-01-01T00:00:00</startTime>
 <endTime>2017-01-01T01:00:00</endTime>
 </timeslot>
 <requiredSkill
reference="../../../../employeeList/employee rostering.employee rostering.Employee/skills/employee rostering.employee rostering.Skill"/>
 </employee rostering.employee rostering.Shift>
<employee rostering.employee rostering.Shift>
 <timeslot reference="../../../../employee rostering.employee rostering.Shift/timeslot"/>
 <requiredSkill
reference="../../../../employeeList/employee rostering.employee rostering.Employee[3]/skills/employee rostering.employee rostering.Skill"/>
 </employee rostering.employee rostering.Shift>
<employee rostering.employee rostering.Shift>
 <timeslot reference="../../../../employee rostering.employee rostering.Shift/timeslot"/>
 <requiredSkill
reference="../../../../employeeList/employee rostering.employee rostering.Employee[2]/skills/employee rostering.employee rostering.Skill"/>
 </employee rostering.employee rostering.Shift>
</shiftList>
<skillList>
 <employee rostering.employee rostering.Skill
reference="../../../../employeeList/employee rostering.employee rostering.Employee/skills/employee rostering.employee rostering.Skill"/>
 <employee rostering.employee rostering.Skill
reference="../../../../employeeList/employee rostering.employee rostering.Employee[3]/skills/employee rostering.employee rostering.Skill"/>
 <employee rostering.employee rostering.Skill
reference="../../../../employeeList/employee rostering.employee rostering.Employee[2]/skills/employee rostering.employee rostering.Skill"/>
</skillList>
<timeslotList>
 <employee rostering.employee rostering.Timeslot
reference="../../../../shiftList/employee rostering.employee rostering.Shift/timeslot"/>
</timeslotList>
<dayOffRequestList/>
<shiftAssignmentList/>
<score>0hard/0soft</score>
</best-solution>
</solver-instance>

```

## 第12章 OPTAPLANNER および QUARKUS の使用ガイド

<https://code.quarkus.redhat.com> の Web サイトを使用して Red Hat ビルドの OptaPlanner Quarkus の Maven プロジェクトを生成し、アプリケーションで使用する拡張機能を自動的に追加および設定できます。その後、Quarkus Maven リポジトリのダウンロードや、プロジェクトでのオンライン Maven リポジトリの使用が可能になります。

### 12.1. APACHE MAVEN および RED HAT ビルドの QUARKUS

Apache Maven は分散型構築自動化ツールで、ソフトウェアプロジェクトの作成、ビルド、および管理を行うために Java アプリケーション開発で使用されます。Maven は Project Object Model (POM) ファイルと呼ばれる標準の設定ファイルを使用して、プロジェクトの定義や構築プロセスの管理を行います。POM ファイルは、モジュールおよびコンポーネントの依存関係、ビルドの順番、結果となるプロジェクトパッケージのターゲットを記述し、XML ファイルを使用して出力します。これにより、プロジェクトが適切かつ統一された状態でビルドされるようになります。

#### Maven リポジトリ

Maven リポジトリには、Java ライブラリー、プラグイン、およびその他のビルドアーティファクトが格納されます。デフォルトのパブリックリポジトリは Maven 2 Central Repository ですが、複数の開発チームの間で共通のアーティファクトを共有する目的で、社内のプライベートおよび内部リポジトリとすることが可能です。また、サードパーティーのリポジトリも利用できます。

Quarkus プロジェクトでオンライン Maven リポジトリを使用するか、Red Hat build of Quarkus の Maven リポジトリをダウンロードできます。

#### Maven プラグイン

Maven プラグインは、POM ファイルの定義済みの部分で1つ以上のゴールを達成します。Quarkus アプリケーションは以下の Maven プラグインを使用します。

- Quarkus Maven プラグイン (**quarkus-maven-plugin**): Maven による Quarkus プロジェクトの作成を実現して、uber-JAR ファイルの生成をサポートし、開発モードを提供します。
- Maven Surefire プラグイン (**maven-surefire-plugin**): ビルドライフサイクルのテストフェーズで使用され、アプリケーションでユニットテストを実行します。プラグインは、テストレポートが含まれるテキストファイルと XML ファイルを生成します。

#### 12.1.1. オンラインリポジトリの Maven の settings.xml ファイルの設定

ユーザーの **settings.xml** ファイルを設定して、Maven プロジェクトでオンライン Maven リポジトリを使用できます。これは、推奨の手法です。リポジトリマネージャーまたは共有サーバー上のリポジトリと使用する Maven 設定は、プロジェクトの制御および管理性を向上させます。



#### 注記

Maven の **settings.xml** ファイルを変更してリポジトリを設定する場合、変更はすべての Maven プロジェクトに適用されます。

#### 手順

1. テキストエディターまたは統合開発環境 (IDE) で Maven の `~/m2/settings.xml` ファイルを開きます。



### 注記

~/m2/ ディレクトリーに **settings.xml** ファイルがない場合には、**\$MAVEN\_HOME/m2/conf/** ディレクトリーから ~/m2/ ディレクトリーに **settings.xml** ファイルをコピーします。

- 以下の行を Maven の **settings.xml** ファイルの **<profiles>** 要素に追加します。

```

<!-- 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>

```

- 以下の行を **settings.xml** ファイルの **<activeProfiles>** 要素に追加し、ファイルを保存します。

```

<activeProfile>red-hat-enterprise-maven-repository</activeProfile>

```

## 12.1.2. Quarkus Maven リポジトリーのダウンロードおよび設定

オンライン Maven リポジトリーを使用しない場合は、Quarkus Maven リポジトリーをダウンロードして設定できます。Quarkus Maven リポジトリーには、Java 開発者がアプリケーションの構築に使用する要件の多くが含まれています。この手順では、Maven の **settings.xml** ファイルを編集し、Quarkus Maven リポジトリーを設定する方法を説明します。



### 注記

Maven の **settings.xml** ファイルを変更してリポジトリーを設定する場合、変更はすべての Maven プロジェクトに適用されます。

### 手順

1. Red Hat カスタマーポータルの [Software Downloads](#) ページから Red Hat build of Quarkus Maven リポジトリの ZIP ファイルをダウンロードします。
2. ダウンロードしたアーカイブをデプロイメントします。
3. `~/.m2/` ディレクトリに移動し、テキストエディターまたは統合開発環境 (IDE) で Maven の **settings.xml** ファイルを開きます。
4. 以下の行を **settings.xml** ファイルの `<profiles>` 要素に追加します。ここで、**QUARKUS\_MAVEN\_REPOSITORY** はダウンロードした Maven リポジトリのパスです。**QUARKUS\_MAVEN\_REPOSITORY** の形式は `file://$PATH` でなければなりません。たとえば `file:///home/userX/rh-quarkus-2.13.GA-maven-repository/maven-repository` のようになります。

```
<!-- 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. 以下の行を **settings.xml** ファイルの `<activeProfiles>` 要素に追加し、ファイルを保存します。

```
<activeProfile>red-hat-enterprise-maven-repository</activeProfile>
```

## 重要

Maven リポジトリに古いアーティファクトが含まれる場合は、プロジェクトをビルドまたはデプロイしたときに以下のいずれかの Maven エラーメッセージが表示されることがあります。ここで、**ARTIFACT\_NAME** は不明なアーティファクトの名前で、**PROJECT\_NAME** は構築を試みているプロジェクトの名前になります。

- **Missing artifact PROJECT\_NAME**
- **[ERROR] Failed to execute goal on project ARTIFACT\_NAME; Could not resolve dependencies for PROJECT\_NAME**

この問題を解決するには、`~/.m2/repository` ディレクトリにあるローカルリポジトリのキャッシュバージョンを削除し、最新の Maven アーティファクトを強制的にダウンロードします。

## 12.2. MAVEN プラグインを使用した OPTAPLANNER RED HAT ビルドの QUARKUS MAVEN プロジェクトの作成

Apache Maven および Quarkus Maven プラグインを使用して、Red Hat ビルドの OptaPlanner および Quarkus アプリケーションの使用を開始できます。

### 前提条件

- OpenJDK 11 以降がインストールされている。Red Hat ビルドの Open JDK は Red Hat カスタマーポータル (ログインが必要) の [ソフトウェアダウンロード](#) ページから入手できます。
- Apache Maven 3.6 以降がインストールされている。Maven は [Apache Maven Project](#) の Web サイトから入手できます。

### 手順

1. コマンドターミナルで以下のコマンドを入力し、Maven が JDK 11 を使用していること、そして Maven のバージョンが 3.6 以上であることを確認します。

```
mvn --version
```

2. 上記のコマンドで JDK 11 が返されない場合は、JDK 11 へのパスを PATH 環境変数に追加し、上記のコマンドを再度入力します。
3. Quarkus OptaPlanner クイックスタートプロジェクトを生成するには、以下のコマンドを入力します。

```
mvn com.redhat.quarkus.platform:quarkus-maven-plugin:2.13.Final-redhat-00006:create \
 -DprojectId=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
```

このコマンドは、`./optaplanner-quickstart` ディレクトリで以下の要素を作成します。

- Maven の構造

- `src/main/docker` の **Dockerfile** ファイルの例
- アプリケーションの設定ファイル

表12.1 `mvn io.quarkus:quarkus-maven-plugin:2.13.Final-redhat-00006:create` コマンドで使用したプロパティ

プロパティ	説明
<code>projectGroupId</code>	プロジェクトのグループ ID。
<code>projectArtifactId</code>	プロジェクトのアーティファクト ID。
<code>extensions</code>	このプロジェクトで使用する Quarkus 拡張のコンマ区切りリスト。Quarkus 拡張の全リストについては、特定のコマンドラインで <code>mvn quarkus:list-extensions</code> を入力します。
<code>noExamples</code>	テストまたはクラスを使用せずに、プロジェクト構造でプロジェクトを作成します。

`projectGroupId` および `projectArtifactId` プロパティの値を使用して、プロジェクトバージョンを生成します。デフォルトのプロジェクトバージョンは **1.0.0-SNAPSHOT** です。

4. OptaPlanner プロジェクトを表示するには、OptaPlanner Quickstarts ディレクトリーに移動します。

```
cd optaplanner-quickstart
```

5. `pom.xml` ファイルを確認します。コンテンツの例を以下に示します。

```
<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>
```

## 12.3. CODE.QUARKUS.REDHAT.COM を使用した RED HAT ビルドの QUARKUS MAVEN プロジェクトの作成

<https://code.quarkus.redhat.com> の Web サイトを使用して Red Hat ビルドの OptaPlanner Quarkus の Maven プロジェクトを生成し、アプリケーションで使用する拡張機能を自動的に追加および設定できます。さらに、[code.quarkus.redhat.com](https://code.quarkus.redhat.com) は、プロジェクトをネイティブ実行可能ファイルにコンパイルするために必要な設定パラメーターを自動的に管理します。

本セクションでは、以下のトピックを含む OptaPlanner Maven プロジェクトを生成するプロセスについて説明します。

- アプリケーションの基本情報を指定する
- プロジェクトに追加する機能拡張の選択
- プロジェクトファイルでダウンロード可能なアーカイブの生成
- アプリケーションのコンパイルおよび起動のカスタムコマンドの使用

### 前提条件

- Web ブラウザーがある。

### 手順

1. Web ブラウザーで <https://code.quarkus.redhat.com> を開きます。
2. プロジェクトの詳細を指定します。
3. プロジェクトのグループ名を入力します。名前の形式は、Java パッケージ命名規則に従います (例: **com.example**)。
4. プロジェクトから生成された Maven アーティファクトに使用する名前を入力します (例: **code-with-quarkus**)。

5. **Build Tool** > **Maven** を選択して、作成するプロジェクトが Maven プロジェクトであることを指定します。選択するビルドツールにより、以下の項目が決定されます。

- 生成されたプロジェクトのディレクトリー構造。
- 生成されたプロジェクトで使用される設定ファイルの形式。
- アプリケーションのコンパイルおよび起動用のカスタムビルドスクリプトおよびコマンド (プロジェクトの生成後に **code.quarkus.redhat.com** が表示)。



### 注記

Red Hat は、**code.quarkus.redhat.com** を使用した OptaPlanner Maven プロジェクトの作成だけをサポートします。Red Hat では、Gradle プロジェクトの生成はサポートしていません。

6. プロジェクトから生成されたアーティファクトで使用するバージョンを入力します。このフィールドのデフォルト値は **1.0.0-SNAPSHOT** です。semantic versioning の使用が推奨されますが、必要に応じて、別のタイプのバージョンを使用できます。
7. プロジェクトをパッケージ化する時に、ビルドツールが生成するアーティファクトのパッケージ名を入力します。  
パッケージ名は、Java パッケージの命名規則に従い、プロジェクトに使用するグループ名と一致するはずですが、別の名前を指定することもできます。



### 注記

**code.quarkus.redhat.com** Web サイトは、OptaPlanner の最新リリースを自動的に使用します。プロジェクトの生成後に、**pom.xml** ファイルで BOM バージョンを手動で変更できます。

8. 以下の拡張を選択して、依存関係として組み込みます。
- RESTEasy JAX-RS (quarkus-resteasy)
  - RESTEasy Jackson (quarkus-resteasy-jackson)
  - OptaPlanner AI 制約ソルバー (optaplanner-quarkus)
  - OptaPlanner Jackson (optaplanner-quarkus-jackson)
- Red Hat は、リストにある個別の拡張に対してさまざまなレベルのサポートを提供します。レベルは、各拡張名の横にあるラベルで示されています。
- **SUPPORTED** 拡張: Red Hat は、実稼働環境のエンタープライズアプリケーションでの使用を完全にサポートします。
  - **TECH-PREVIEW** 拡張: Red Hat は、[テクノロジープレビュー機能のサポート範囲](#)に基づき、限定的に、実稼働環境でのサポートを提供します。
  - **DEV-SUPPORT** 拡張: Red Hat は、実稼働環境での使用をサポートしていません。ただし、新規アプリケーションの開発での使用に対しては、Red Hat 開発者がこれらのコア機能をサポートしています。
  - **DEPRECATED** 拡張: 同じ機能を提供する新しいテクノロジーまたは実装に置き換える予定です。

Red Hat では、ラベル付けされていない拡張の実稼働環境での使用はサポートしていません。

9. **Generate your application** を選択して選択内容を確認し、生成されたプロジェクトを含むアーカイブのダウンロードリンクのオーバーレイ画面を表示します。オーバーレイ画面には、アプリケーションのコンパイルおよび起動に使用できるカスタムコマンドも表示されます。
10. **Download the ZIP** を選択して、生成されたプロジェクトファイルを含むアーカイブをマシンに保存します。
11. アーカイブの内容をデプロイメントします。
12. デプロイメントしたプロジェクトファイルが含まれるディレクトリーに移動します。

```
cd <directory_name>
```

13. 開発モードでアプリケーションをコンパイルして起動します。

```
./mvnw compile quarkus:dev
```

## 12.4. QUARKUS CLI を使用した RED HAT ビルドの QUARKUS MAVEN プロジェクトの作成

Quarkus コマンドラインインターフェイス (CLI) を使用して、Quarkus OptaPlanner プロジェクトを作成できます。

### 前提条件

- Quarkus CLI をインストールしている。詳細は、[Building Quarkus Apps with Quarkus Command Line Interface](#) を参照してください。

### 手順

1. Quarkus アプリケーションを作成します。

```
quarkus create app -P io.quarkus:quarkus-bom:2.13.Final-redhat-00006
```

2. 利用可能な拡張機能を表示するには、以下のコマンドを入力します。

```
quarkus ext -i
```

このコマンドは、以下の拡張機能を返します。

```
optaplanner-quarkus
optaplanner-quarkus-benchmark
optaplanner-quarkus-jackson
optaplanner-quarkus-jsonb
```

3. 以下のコマンドを入力して、エクステンションをプロジェクトの **pom.xml** ファイルに追加します。

```
quarkus ext add resteasy-jackson
quarkus ext add optaplanner-quarkus
quarkus ext add optaplanner-quarkus-jackson
```

4. テキストエディターで **pom.xml** ファイルを開きます。ファイルの内容は以下の例のようになります。

```
<?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>
 </dependencies>
```

```

 <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>
 </systemPropertyVariables>
 </configuration>
 </execution>
 </executions>
 </plugin>
 </plugins>
 </build>
 <properties>
 <quarkus.package.type>native</quarkus.package.type>
 </properties>
</profile>
</profiles>
</project>
```

## 付録A バージョン情報

本書の最終更新日: 2023 年 9 月 5 日 (火)

## 付録B お問い合わせ先

Red Hat Decision Manager ドキュメントチーム: [brms-docs@redhat.com](mailto:brms-docs@redhat.com)