Institute of Systems Science
National University of Singapore

GRADUATE CERTIFICATE
INTELLIGENT REASONING SYSTEMS

Workshop Project (Continuous Assessment) Guide

Subject: *Reasoning Systems*

Agenda : Course Assessment & Grading

⟹ **EEP & MTech Stackable**

- **Paper Assessment** on last lecture day
  - [**Individual** 50 marks] 1 hour open book exam (course level)
- **Workshop Project Deliverables** due last lecture day
  - [**Individual** 25 marks] A runnable standalone bespoke hybrid reasoning system
  - [**Individual** 25 marks] A project report with relevant attachments, including
    - System Design / Knowledge Models
    - Use/Test cases

⟹ **MTech Thru-Train**

- **Paper Assessment** on last lecture day
  - [**Individual** 50 marks] 1 hour open book exam (course level)
- **Workshop Project Deliverables** due last lecture day + 14
  - [**Group** 20 marks] A runnable standalone bespoke hybrid reasoning system
  - [**Group** 20 marks] A project report with relevant attachments, including
    - System Design / Knowledge Models
    - System Development & Implementation in tools, e.g. KIE suite
    - System User Guide
  - [**Group** 10 marks] A 5 minute video presentation, covering
    - System Design & Use Case Demo

- **Source impactful real life business scenario for workshop project.**
- **Conduct comprehensive research and reference reading.**
- **Make your own reasonable assumptions where necessary.**

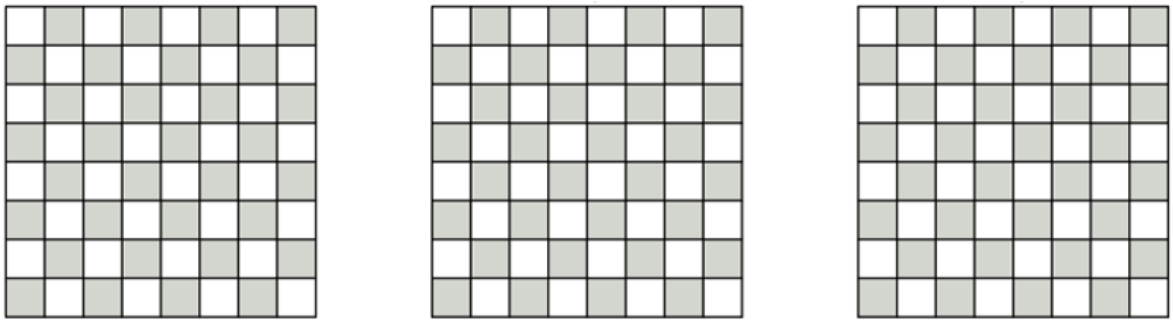
Workshop 1 Guide

WORKSHOP SEARCH REPRESENTATION

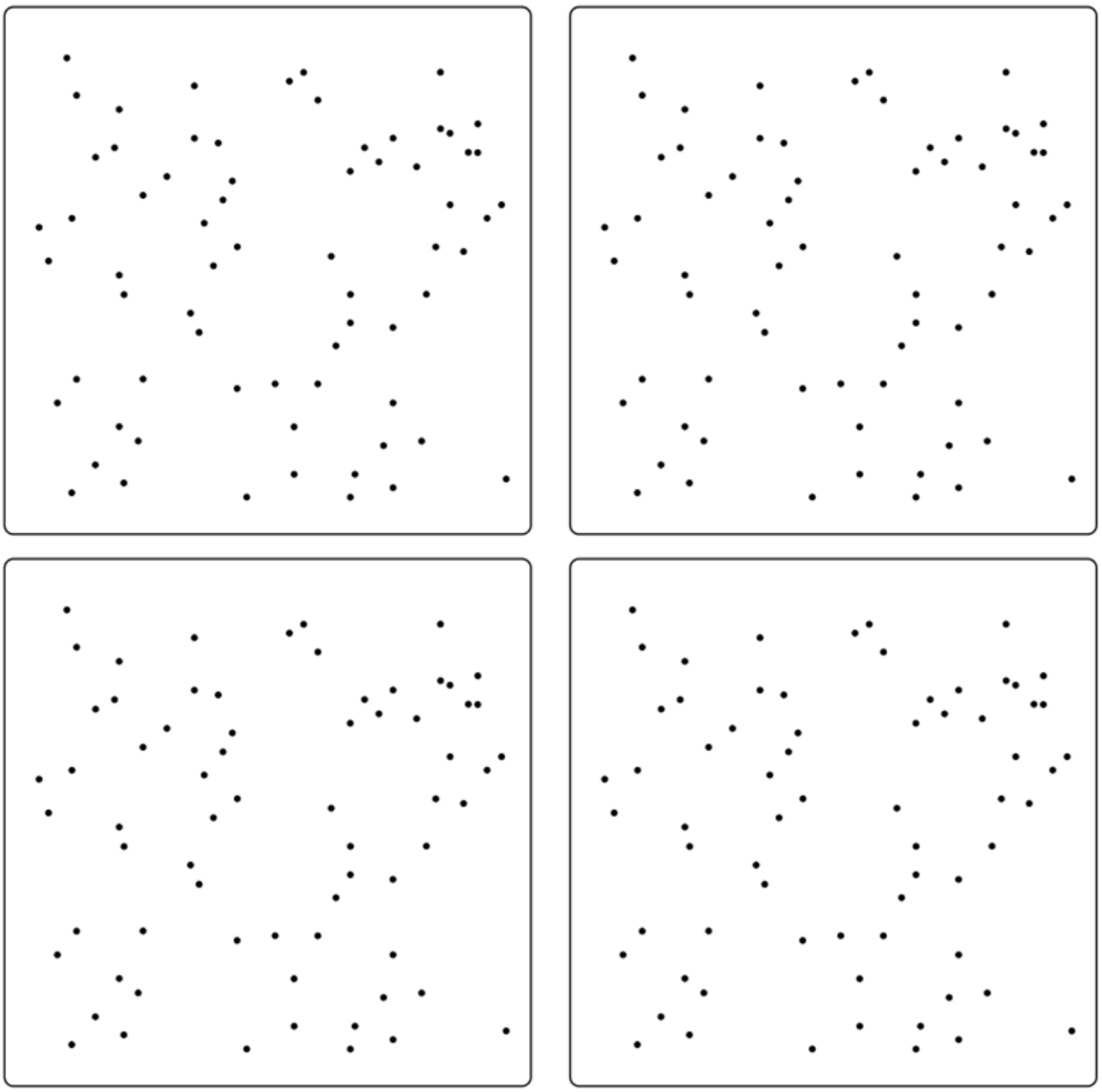
- **Search Modelling & Representation**
  - Pen & Paper Planning
  - Robot Navigation
  - Vehicle Route Planning (VRP)
- **KIE OptaPlanner Tutorial**
  - Optimizing Vehicle Route Planning (VRP)
  - Optimizing Europe Travelling Sales Person (TSP)

Workshop 1.1 [Individual]

1) Place 8 queens on this chessboard so no 2 queens can attack each other.

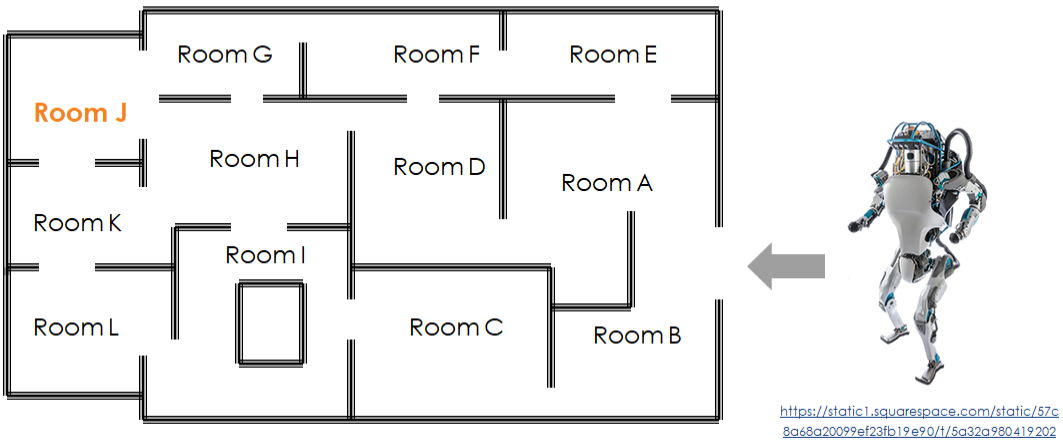


2) Draw the shortest line that connects all dots and returns to its origin.



Workshop 1.2 [Individual]

- **Robotics: How to rapidly navigate to Room J ?**



https://static1.squarespace.com/static/57c8a68a20099ef23fb19e90/t/5a32a9804192022be97f9d10/1513269668629/Atlas.png

Design and draft robot navigation search representation here:

Write down DFS search order:

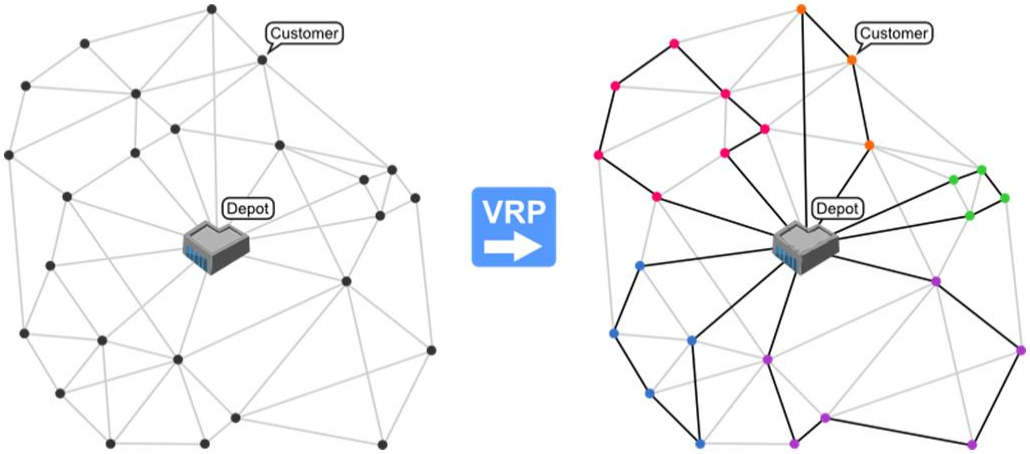
Write down BFS search order:

Construct reasonable heuristics, then design heuristic search strategy to enhance above DFS/BFS brute force search:

Workshop 1.3 [Group]

Form groups: 4-6 learners per group.

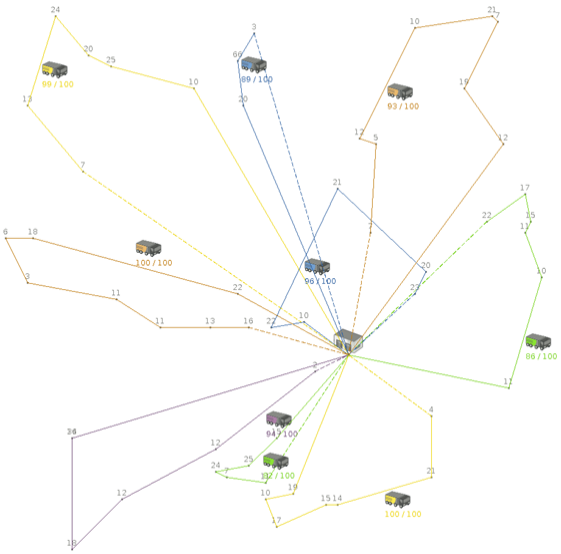
Vehicle Route Problem



http://neo.lcc.uma.es/vrp/vehicle-routing-problem/

Optimizing Vehicle Route Planning

- We are a logistic company owning a warehouse and 9 delivery trucks. This morning we received 54 customer orders, with different load demand, and different locations. Our truck's maximum load capacity is 100 TVs.
- We want to delivery all customer orders using fewer gasoline. Hence, we'd like to have shortest distance of combined truck delivery routes.

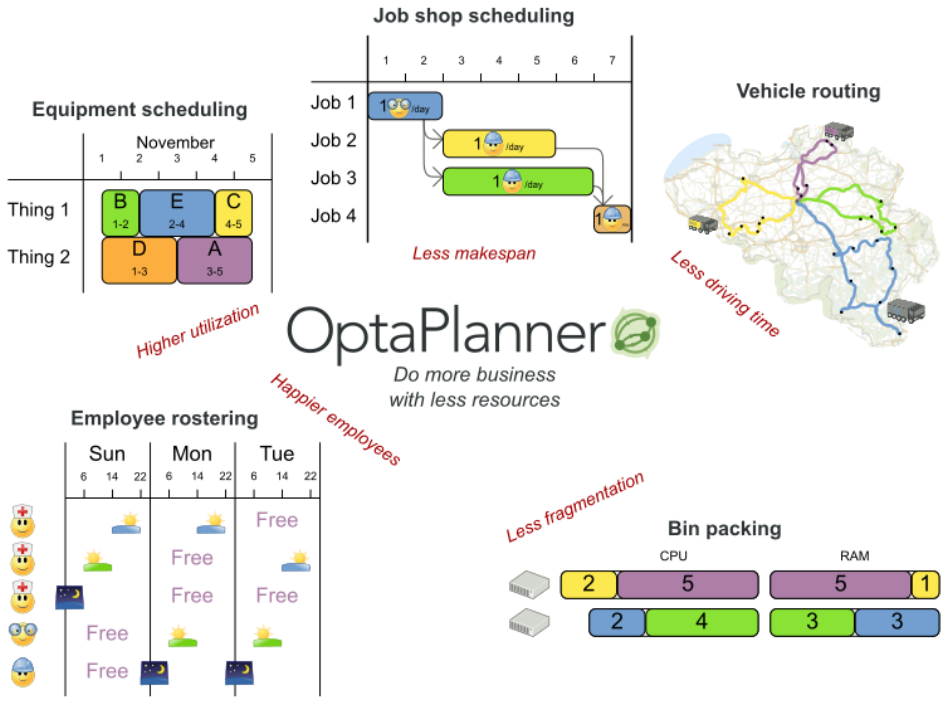


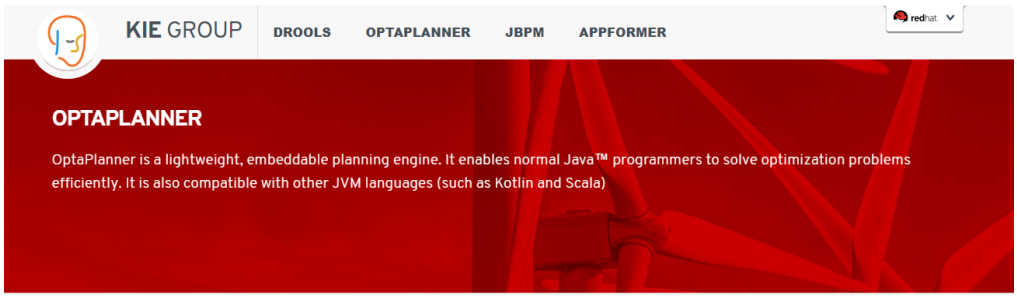
Design and draft VRP search representation here:

Strategy to avoid looping (revisiting same customer):

Workshop 1.4 [Individual]

KIE OptaPlanner Tutorial: Installation; KIE Workbench; IDE plug-in; Case Study





DROOLS

Drools is a business rule management system with a forward-chaining and backward-chaining inference based rules engine, allowing fast and reliable evaluation of business rules and complex event processing.

Read more →

OPTAPLANNER

OptaPlanner is a constraint solver that optimizes use cases such as employee rostering, vehicle routing, task assignment and cloud optimization.

Read more →

JBPM

jBPM is a flexible Business Process Management suite allowing you to model your business goals by describing the steps that need to be executed to achieve those goals.

Read more →

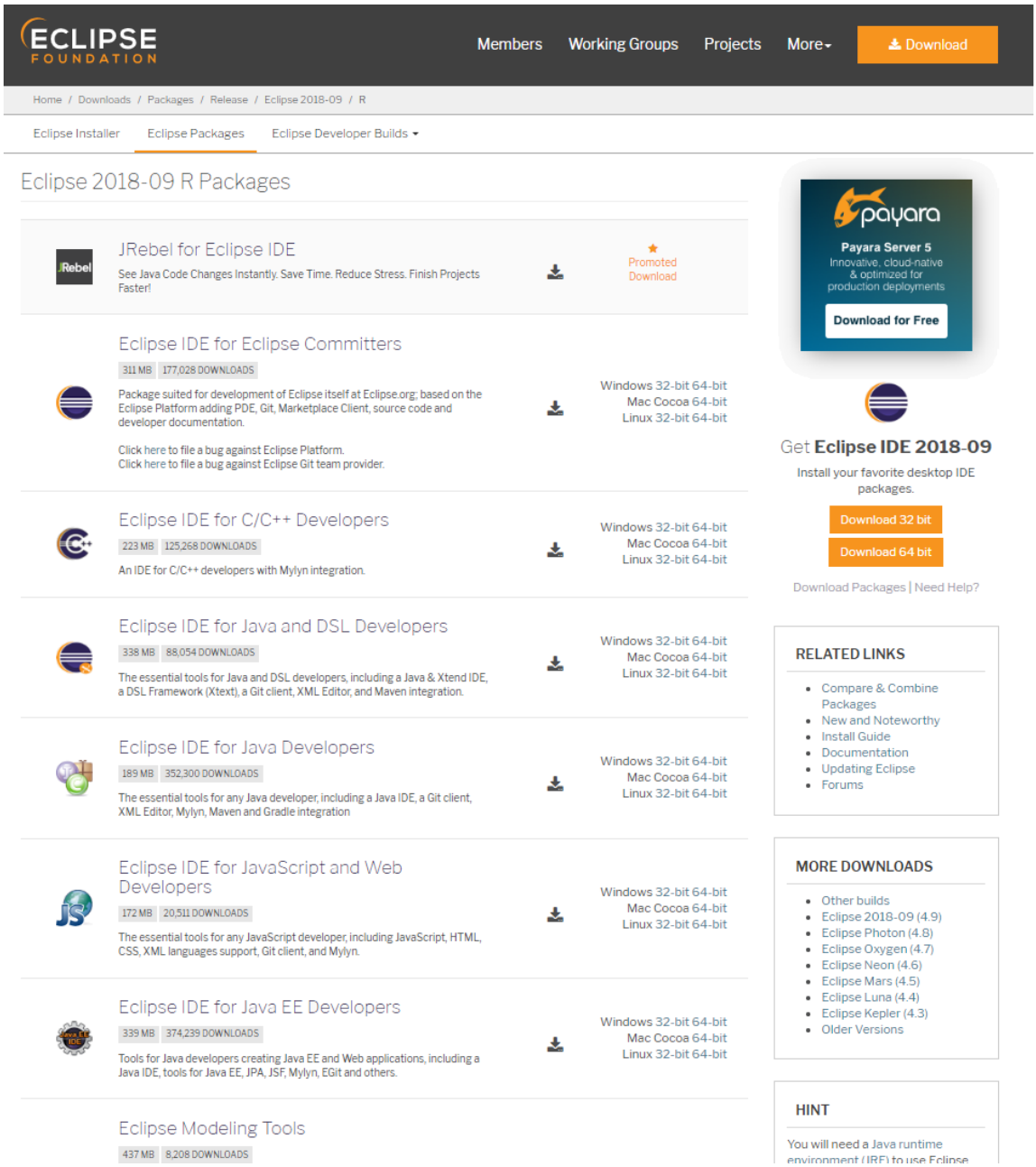
APPFORMER

AppFormer is a low code platform to develop modern applications. It's a powerful tool for developers that can easily build applications by mashing up components and connect them to other Red Hat modules and software.

We make building apps looks easy.

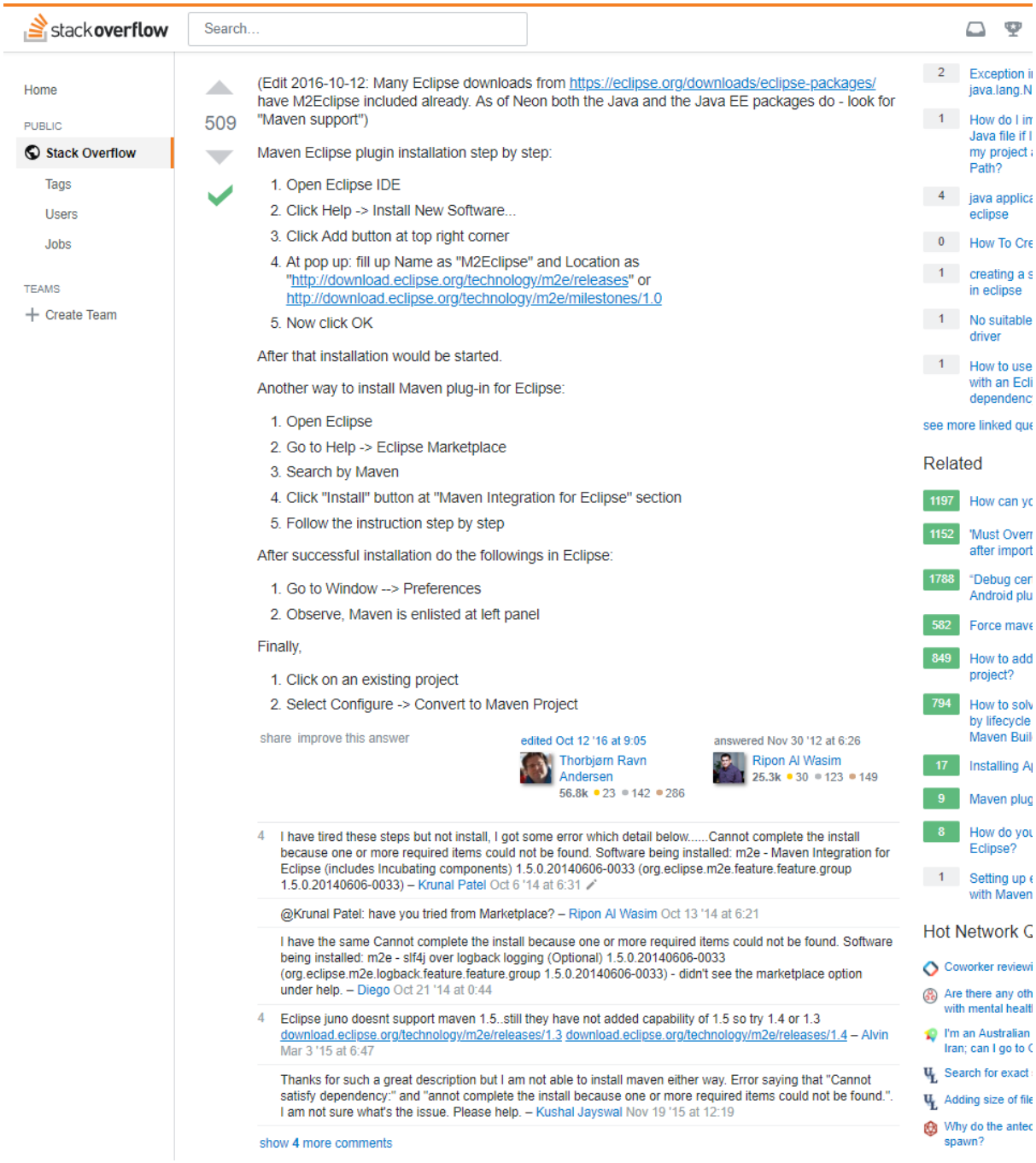
Read more →

Above: OptaPlanner Installation
https://www.optaplanner.org/



Above: Install Eclipse IDE

https://www.eclipse.org/downloads/packages/



stackoverflow Search...

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

TEAMS

Create Team

509

(Edit 2016-10-12: Many Eclipse downloads from https://eclipse.org/downloads/eclipse-packages/ have M2Eclipse included already. As of Neon both the Java and the Java EE packages do - look for "Maven support")

Maven Eclipse plugin installation step by step:

1. Open Eclipse IDE
2. Click Help -> Install New Software...
3. Click Add button at top right corner
4. At pop up: fill up Name as "M2Eclipse" and Location as "http://download.eclipse.org/technology/m2e/releases" or http://download.eclipse.org/technology/m2e/milestones/1.0
5. Now click OK

After that installation would be started.

Another way to install Maven plug-in for Eclipse:

1. Open Eclipse
2. Go to Help -> Eclipse Marketplace
3. Search by Maven
4. Click "Install" button at "Maven Integration for Eclipse" section
5. Follow the instruction step by step

After successful installation do the followings in Eclipse:

1. Go to Window --> Preferences
2. Observe, Maven is enlisted at left panel

Finally,

1. Click on an existing project
2. Select Configure -> Convert to Maven Project

share improve this answer

edited Oct 12 '16 at 9:05 Thorbjørn Ravn Andersen 56.8k • 23 • 142 • 286

answered Nov 30 '12 at 6:26 Ripon Al Wasim 25.3k • 30 • 123 • 149

4 I have tired these steps but not install, I got some error which detail below.....Cannot complete the install because one or more required items could not be found. Software being installed: m2e - Maven Integration for Eclipse (includes Incubating components) 1.5.0.20140606-0033 (org.eclipse.m2e.feature.feature.group 1.5.0.20140606-0033) – Krunal Patel Oct 6 '14 at 6:31

@Krunal Patel: have you tried from Marketplace? – Ripon Al Wasim Oct 13 '14 at 6:21

I have the same Cannot complete the install because one or more required items could not be found. Software being installed: m2e - slf4j over logback logging (Optional) 1.5.0.20140606-0033 (org.eclipse.m2e.logback.feature.feature.group 1.5.0.20140606-0033) - didn't see the marketplace option under help. – Diego Oct 21 '14 at 0:44

4 Eclipse juno doesnt support maven 1.5..still they have not added capability of 1.5 so try 1.4 or 1.3 download.eclipse.org/technology/m2e/releases/1.3 download.eclipse.org/technology/m2e/releases/1.4 – Alvin Mar 3 '15 at 6:47

Thanks for such a great description but I am not able to install maven either way. Error saying that "Cannot satisfy dependency:" and "annot complete the install because one or more required items could not be found.". I am not sure what's the issue. Please help. – Kushal Jayswal Nov 19 '15 at 12:19

show 4 more comments

2 Exception i java.lang.N

1 How do I im Java file if I my project Path?

4 java applica eclipse

0 How To Cre

1 creating a s in eclipse

1 No suitable driver

1 How to use with an Ecli dependenc

see more linked que

Related

1197 How can yc

1152 'Must Overr after import

1788 "Debug cer Android plu

582 Force mave

849 How to add project?

794 How to solv by lifecycle Maven Buil

17 Installing Ap

9 Maven plug

8 How do you Eclipse?

1 Setting up e with Maven

Hot Network Q

Coworker reviewi

Are there any oth with mental healt

I'm an Australian Iran; can I go to

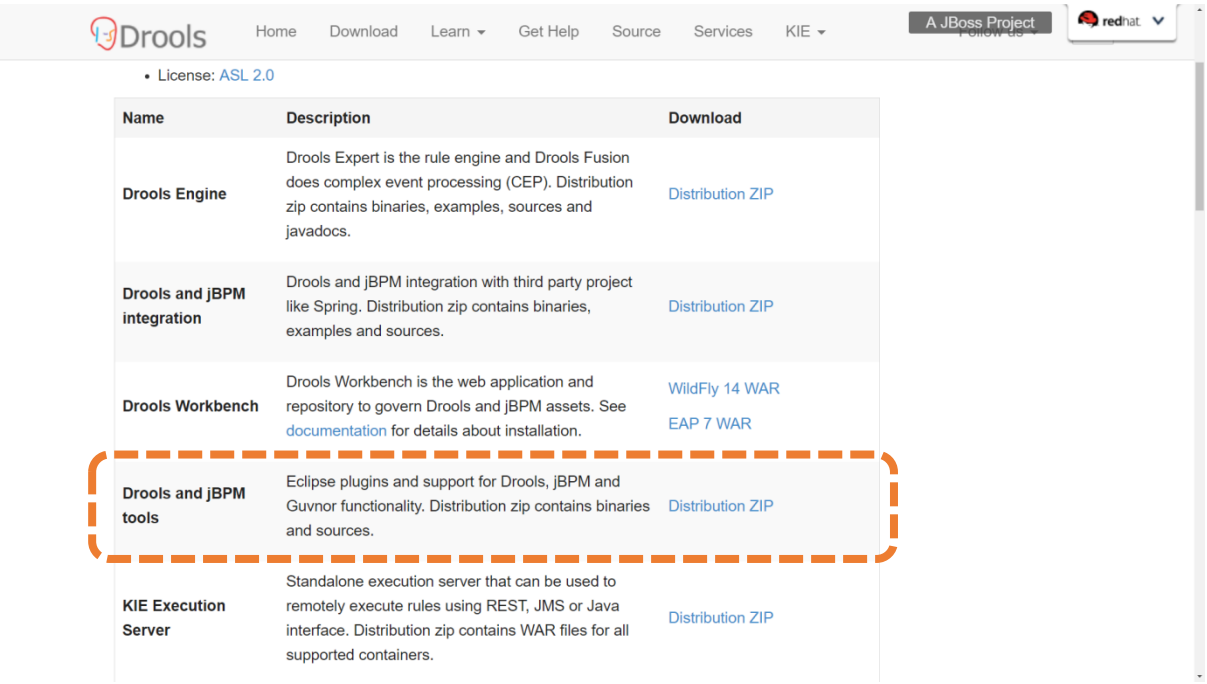
Search for exact

Adding size of file

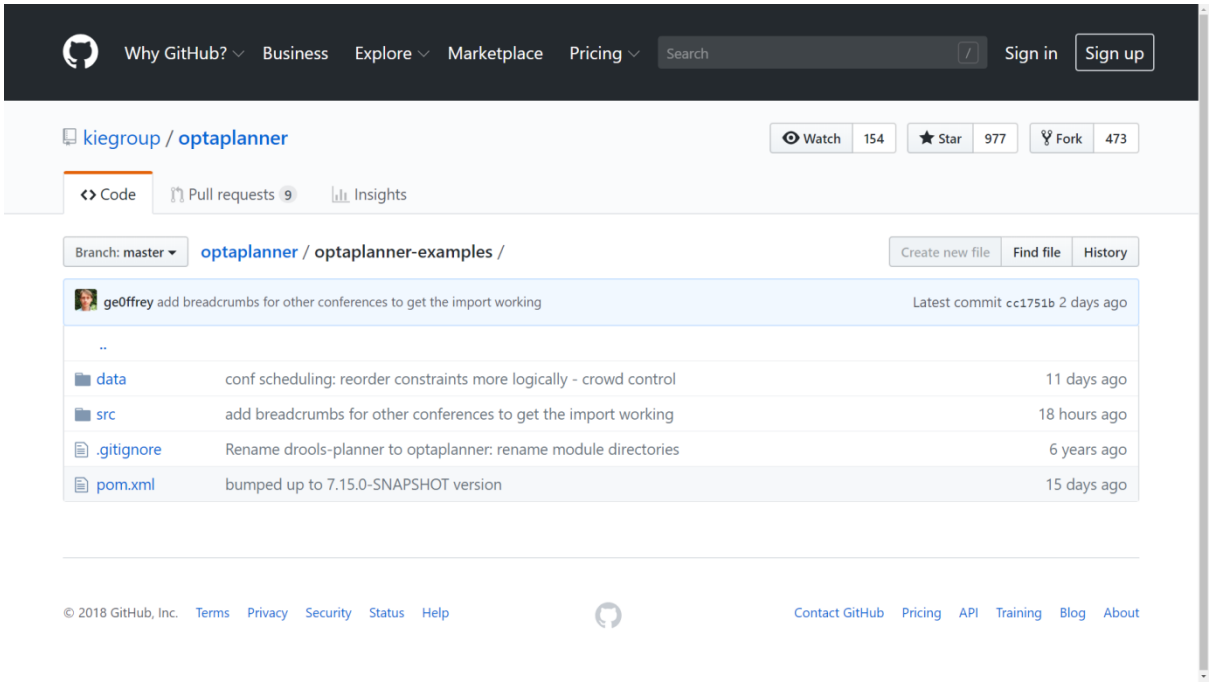
Why do the antec spawn?

Above: Install Maven in Eclipse

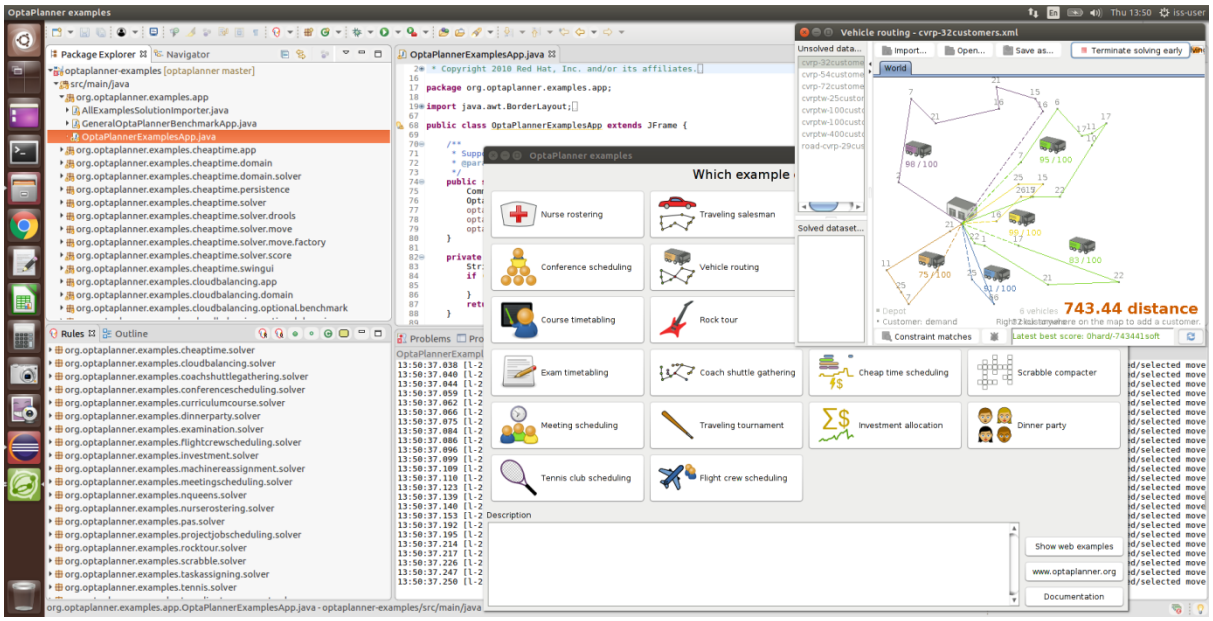
https://stackoverflow.com/questions/8620127/maven-in-eclipse-step-by-step-installation



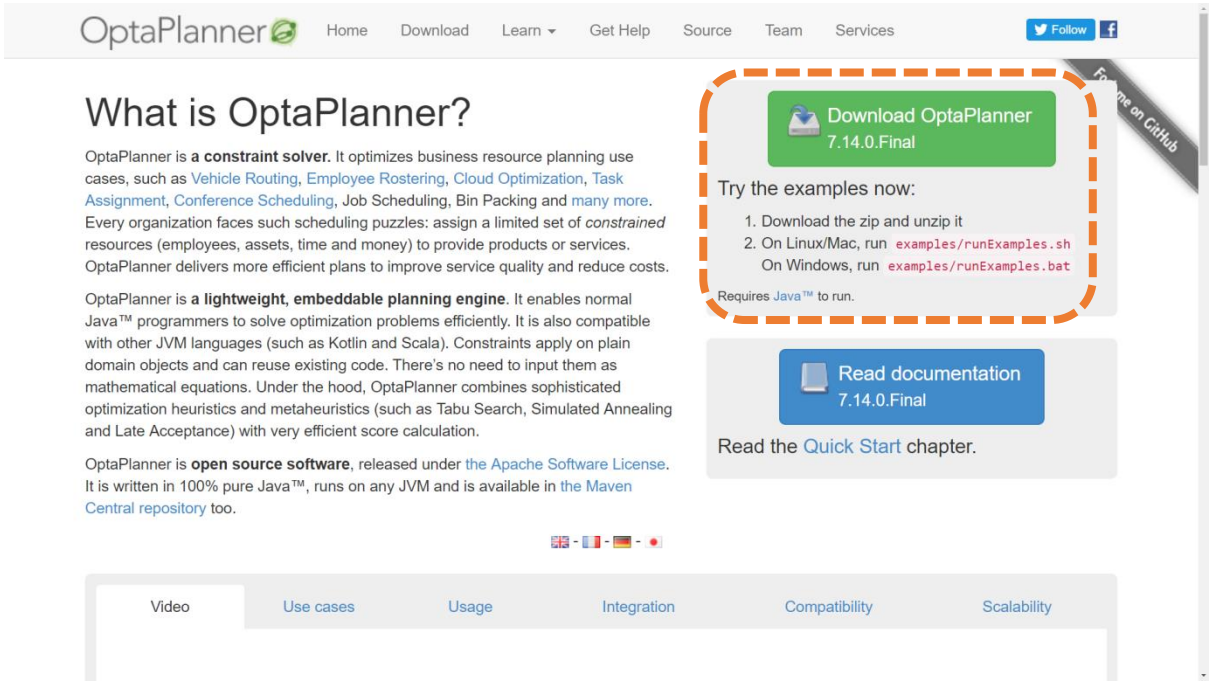
Above: Install KIE plug-in for Eclipse
https://www.drools.org/download/download.html



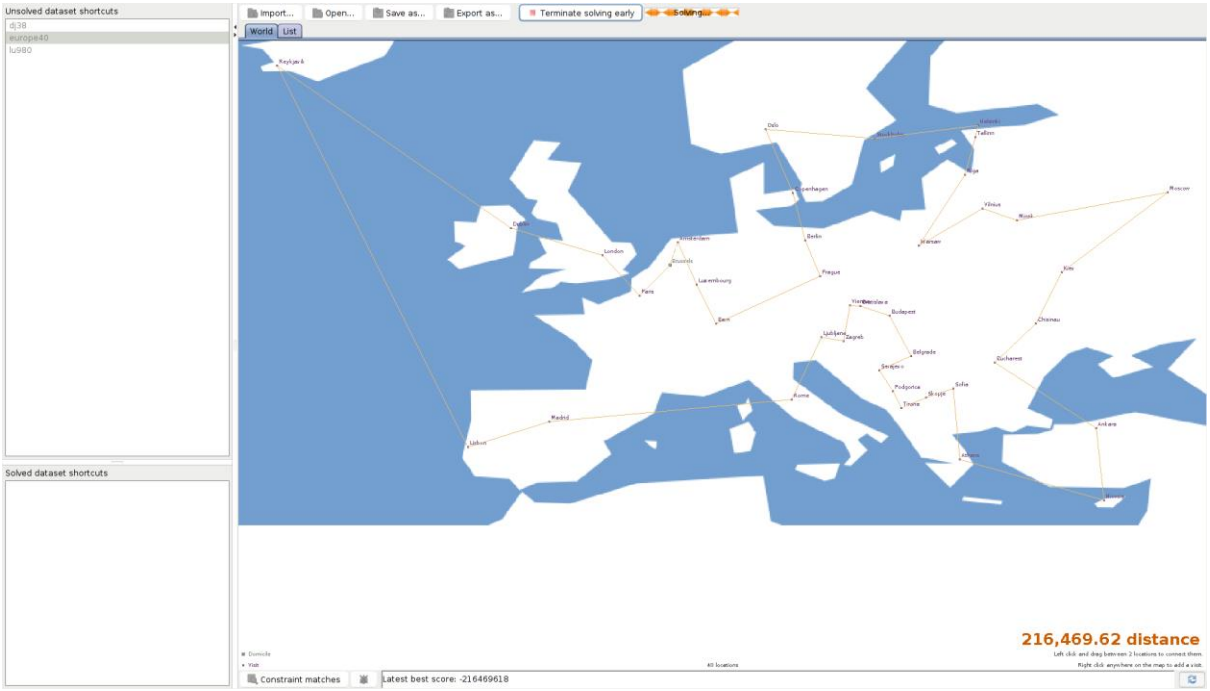
Above: Download OptaPlanner source code and example cases
https://github.com/kiegroup/optaplanner

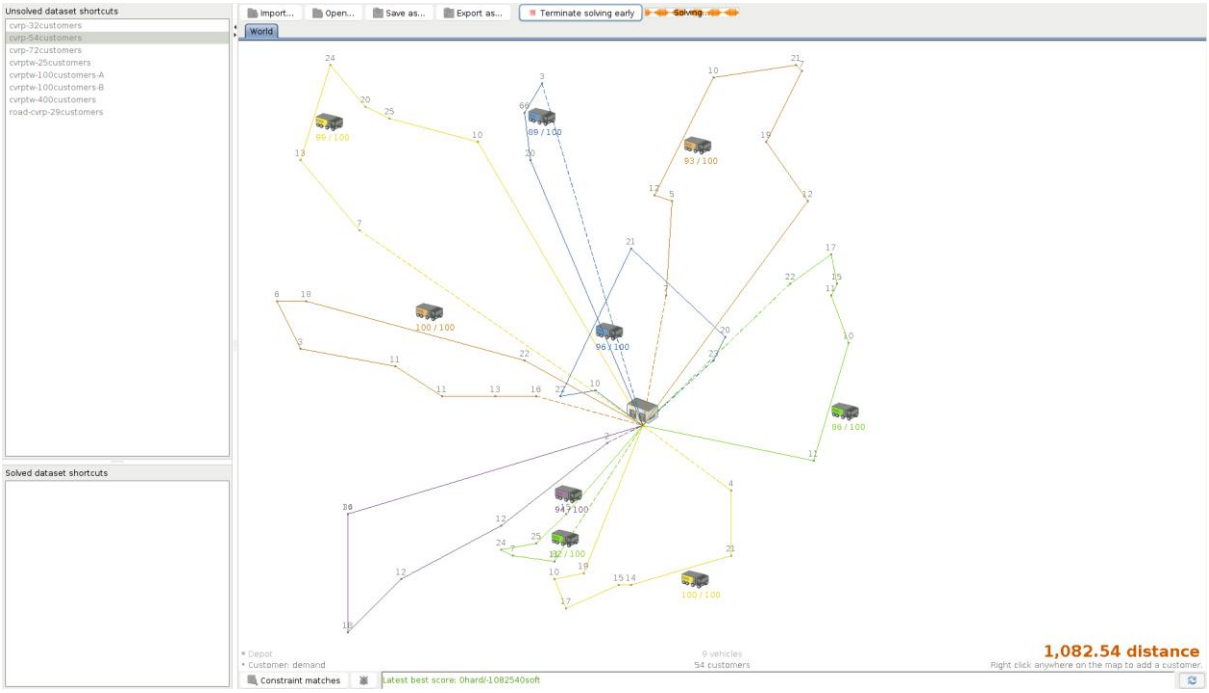


Above: Import OptaPlanner example cases as Maven project in Eclipse

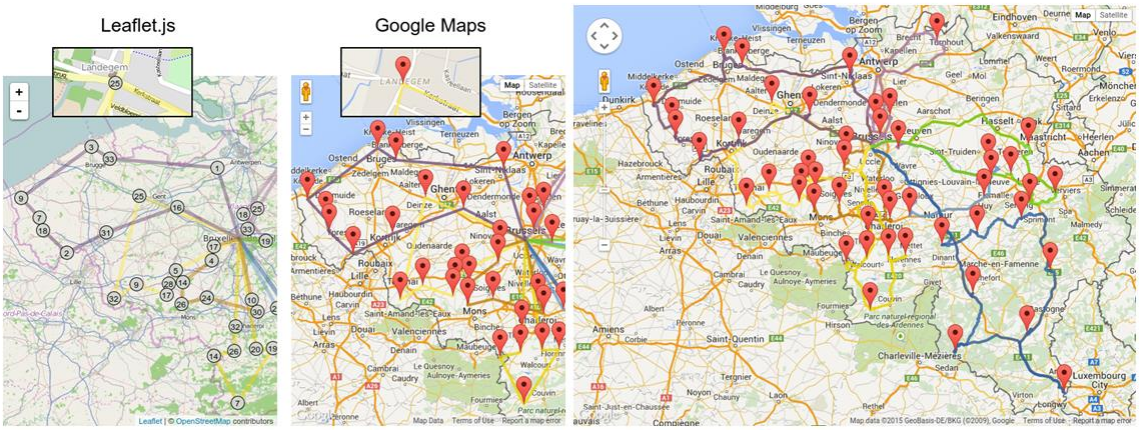


Above: Standalone OptaPlanner
https://www.optaplanner.org/


Unsolved dataset shortcuts
dj38
europe40
lu980
Solved dataset shortcuts
Import...
Open...
Save as...
Export as...
Terminate solving early
Solving...
World
List
216,469.62 distance
Constraint matches
Latest best score: -216469618


Unsolved dataset shortcuts
cvrp-32customers
cvrp-54customers
cvrp-72customers
cvrptw-25customers
cvrptw-100customers-A
cvrptw-100customers-B
cvrptw-400customers
road-cvrp-29customers
Solved dataset shortcuts
Import...
Open...
Save as...
Export as...
Terminate solving early
Solving...
World
1,082.54 distance
Constraint matches
Latest best score: 0hard/-1082540soft

Visualizing Vehicle Routing with Leaflet and Google Maps



https://www.optaplanner.org/blog/2015/03/10/VisualizingVehicleRoutingWithLeafletAndGoogleMaps.html

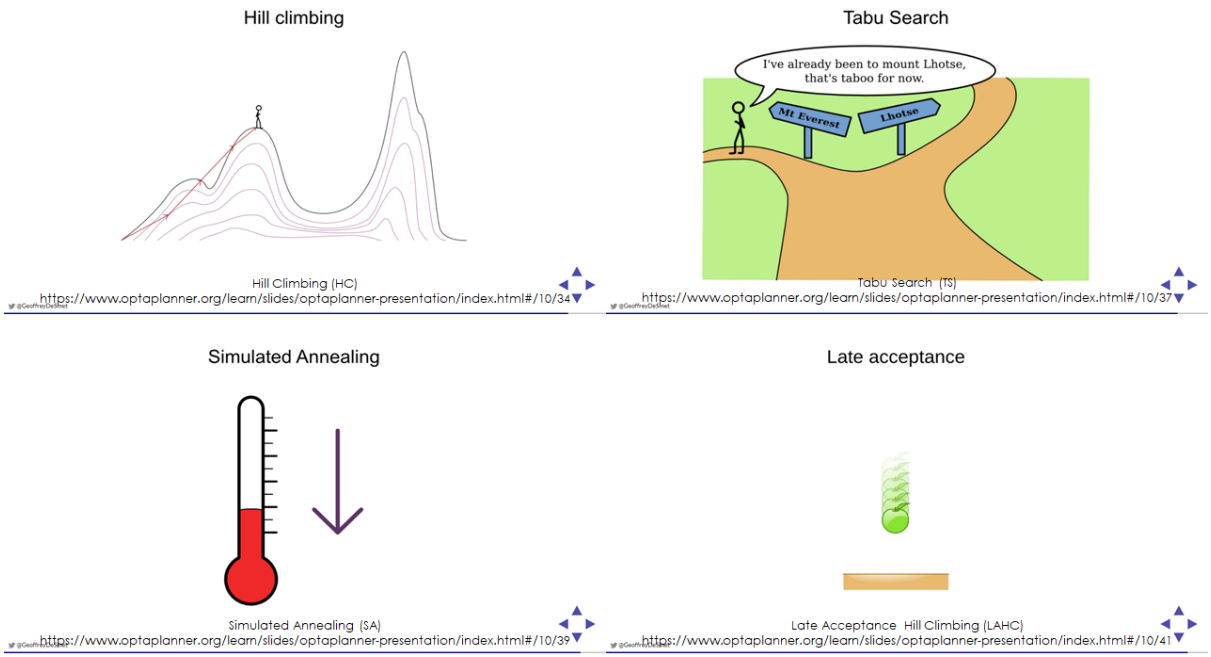
VRP Reference

- http://www.optaplanner.org/learn/useCases/vehicleRoutingProblem.html
- http://www.optaplanner.org/learn/slides/optaplanner-presentation/index.html#/2
- https://www.optaplanner.org/blog/2015/03/10/VisualizingVehicleRoutingWithLeafletAndGoogleMaps.html

Workshop 2 Guide

WORKSHOP SEARCH REASONING

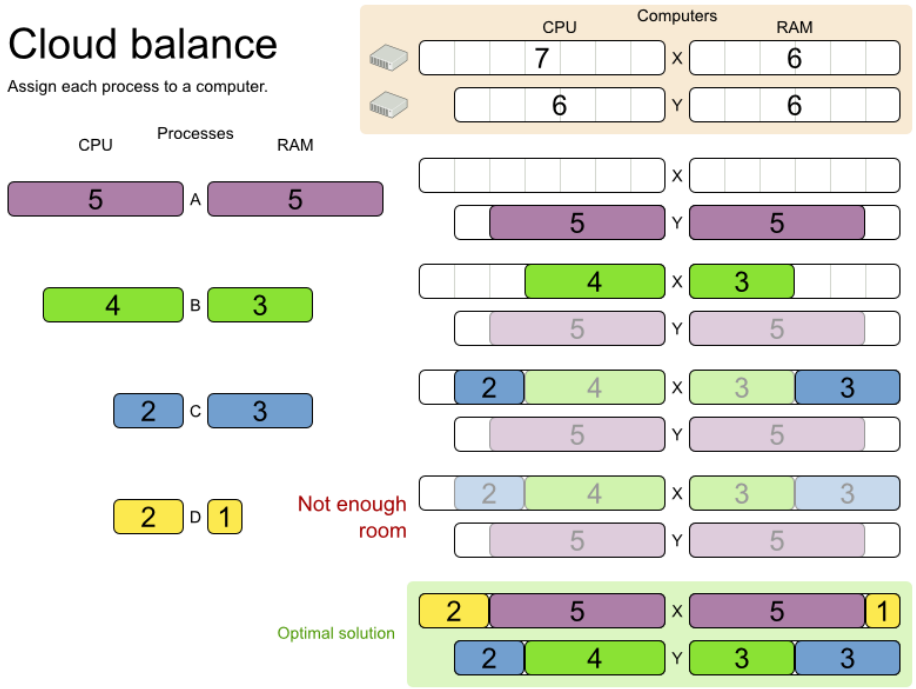
- **KIE OptaPlanner Deep Dive**
  - Search Algorithms in Action
  - Cloud Computer Balancing
- **KIE OptaPlanner Development – Individual Work**
  - Choose one OptaPlanner use case, e.g. course curriculum scheduling, airport gate assignment, etc.
  - Analyse, adapt, import, and solve using KIE Workbench or programming.

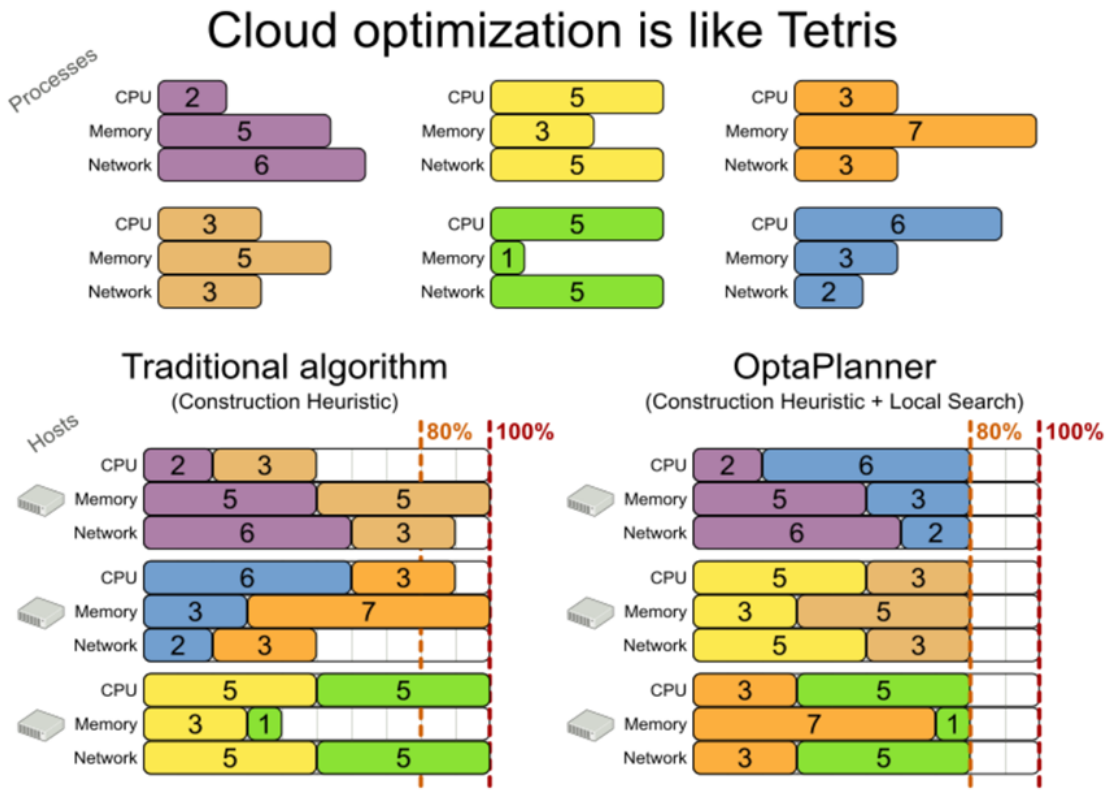


Above: Search Algorithms in Action

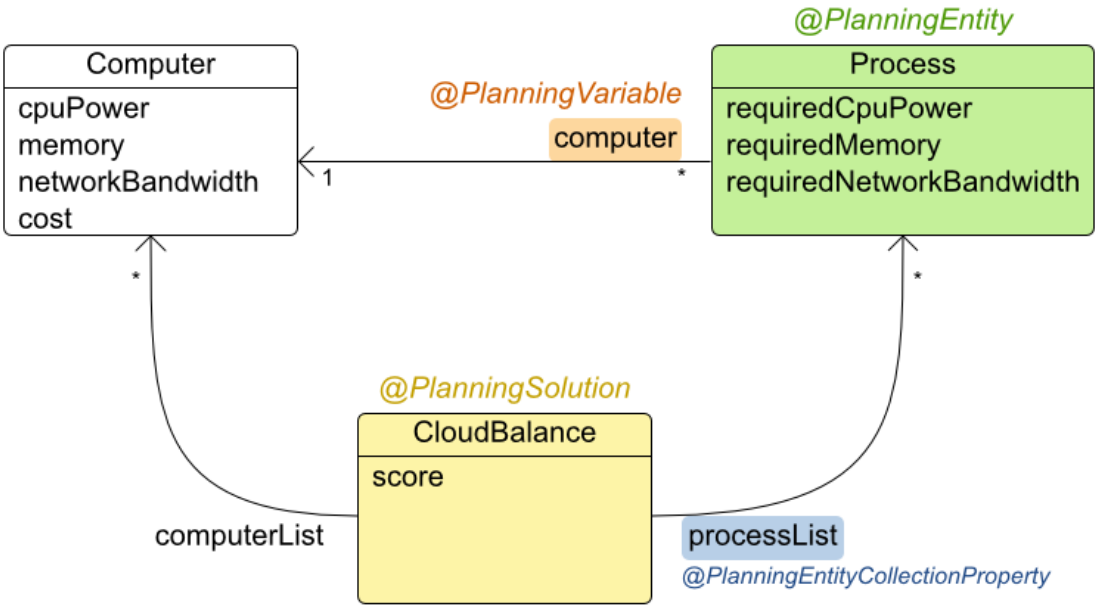
Workshop 2.1 [Individual]

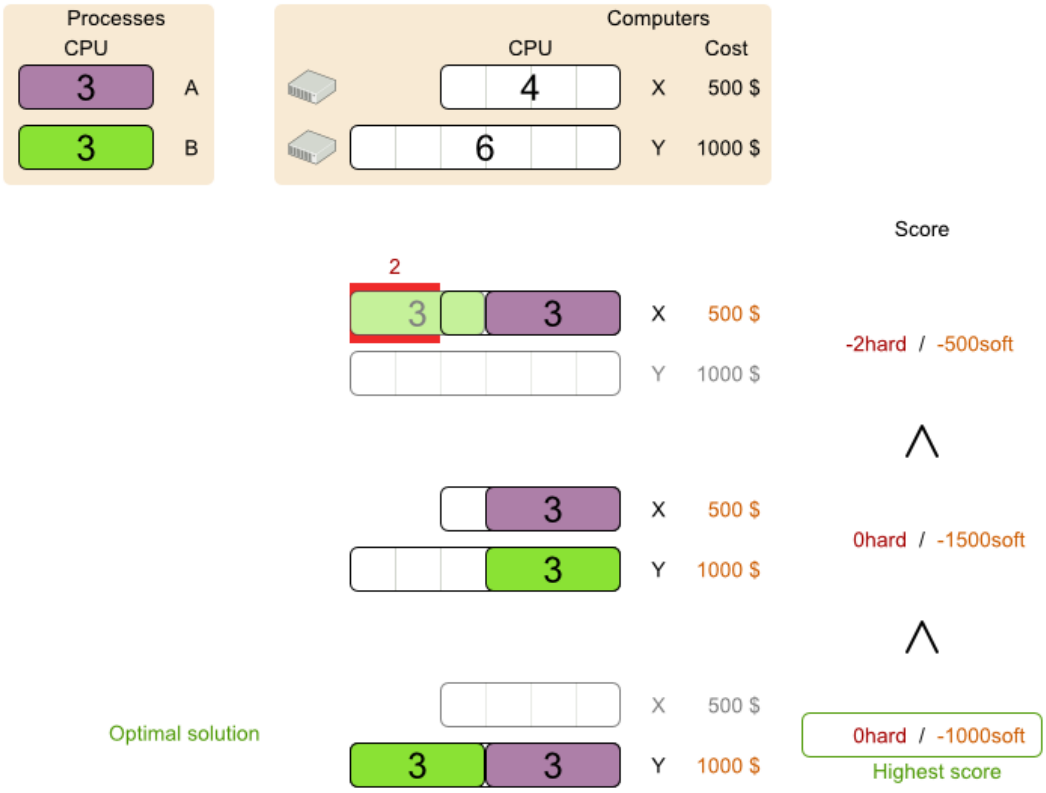
Analyze and execute cloud computer balancing system using both Eclipse and KIE Workbench.



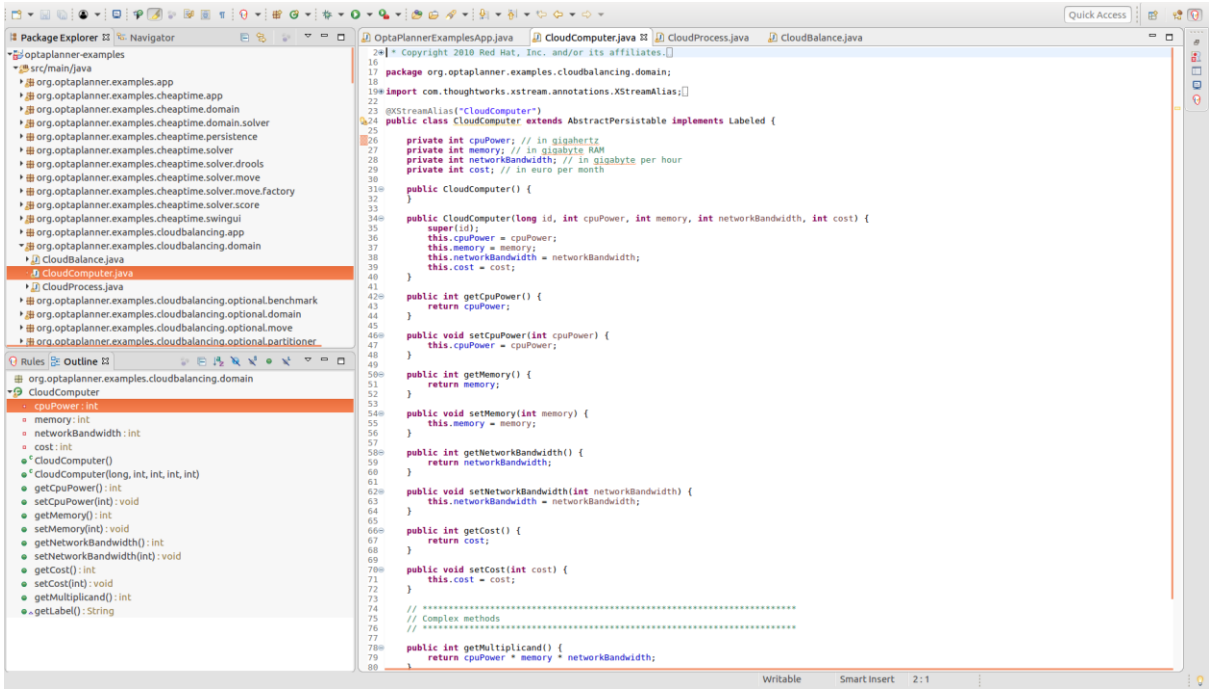


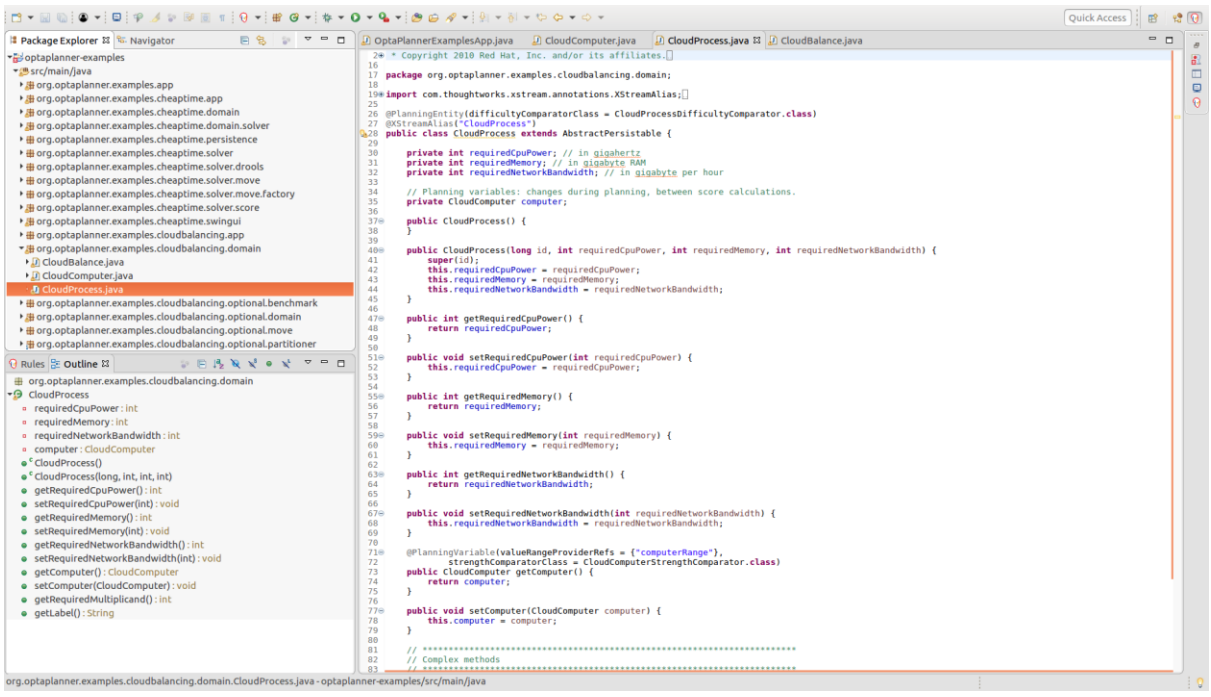
Cloud balance class diagram

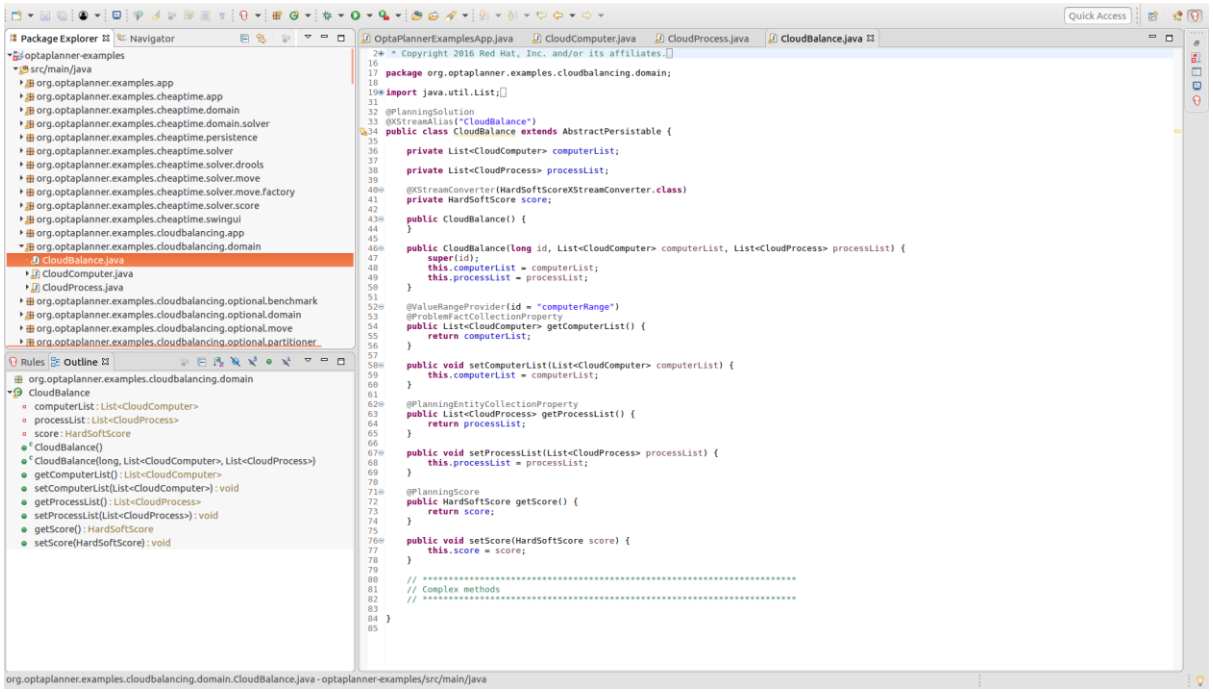


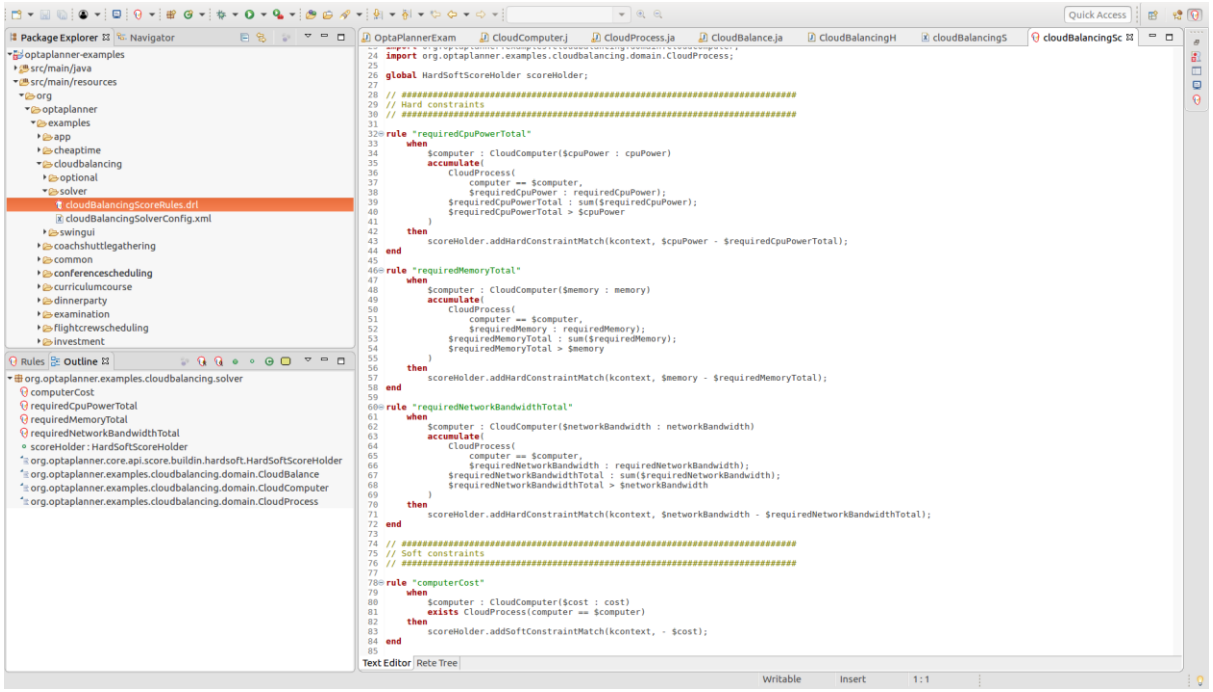


Solver in Java / Eclipse






Package Explorer
Navigator
optaplanner-examples
src/main/java
org.optaplanner.examples.app
org.optaplanner.examples.cheaptime.app
org.optaplanner.examples.cheaptime.domain
org.optaplanner.examples.cheaptime.domain.solver
org.optaplanner.examples.cheaptime.persistence
org.optaplanner.examples.cheaptime.solver
org.optaplanner.examples.cheaptime.solver.drools
org.optaplanner.examples.cheaptime.solver.move
org.optaplanner.examples.cheaptime.solver.move.factory
org.optaplanner.examples.cheaptime.solver.score
org.optaplanner.examples.cheaptime.swingui
org.optaplanner.examples.cloudbalancing.app
org.optaplanner.examples.cloudbalancing.domain
CloudBalance.java
CloudComputer.java
CloudProcess.java
org.optaplanner.examples.cloudbalancing.optional.benchmark
org.optaplanner.examples.cloudbalancing.optional.domain
org.optaplanner.examples.cloudbalancing.optional.move
org.optaplanner.examples.cloudbalancing.optional.partitioner
Rules
Outline
org.optaplanner.examples.cloudbalancing.domain
CloudBalance
computerList : List<CloudComputer>
processList : List<CloudProcess>
score : HardSoftScore
CloudBalance()
CloudBalance(long, List<CloudComputer>, List<CloudProcess>)
getComputerList() : List<CloudComputer>
setComputerList(List<CloudComputer>) : void
getProcessList() : List<CloudProcess>
setProcessList(List<CloudProcess>) : void
getScore() : HardSoftScore
setScore(HardSoftScore) : void
OptaPlannerExamplesApp.java
CloudComputer.java
CloudProcess.java
CloudBalance.java
Quick Access
org.optaplanner.examples.cloudbalancing.domain.CloudBalance.java - optaplanner-examples/src/main/java


Package Explorer
Navigator
optaplanner-examples
src/main/java
src/main/resources
org
optaplanner
examples
app
cheaptime
cloudbalancing
optional
solver
cloudBalancingScoreRules.drl
cloudBalancingSolverConfig.xml
swingui
coachshuttlegathering
common
conferencescheduling
curriculumcourse
dinnerparty
examination
flightcrewscheduling
investment
Rules
Outline
org.optaplanner.examples.cloudbalancing.solver
computerCost
requiredCpuPowerTotal
requiredMemoryTotal
requiredNetworkBandwidthTotal
scoreHolder : HardSoftScoreHolder
org.optaplanner.core.api.score.buildin.hardsoft.HardSoftScoreHolder
org.optaplanner.examples.cloudbalancing.domain.CloudBalance
org.optaplanner.examples.cloudbalancing.domain.CloudComputer
org.optaplanner.examples.cloudbalancing.domain.CloudProcess
OptaPlannerExam
CloudComputer.j
CloudProcess.ja
CloudBalance.ja
CloudBalancingH
cloudBalancingS
cloudBalancingSc
Quick Access
Text Editor
Rete Tree
Writable
Insert
1 : 1

2.5.1. Easy Java Score Configuration

One way to define a score function is to implement the interface `EasyScoreCalculator` in plain Java.

```
<scoreDirectorFactory>
  <easyScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancing
</scoreDirectorFactory>
```

Just implement the `calculateScore(Solution)` method to return a `HardSoftScore` instance.

Example 6. CloudBalancingEasyScoreCalculator.java

```
public class CloudBalancingEasyScoreCalculator implements EasyScoreCalculator<CloudBalance> {

    /**
     * A very simple implementation. The double loop can easily be removed by using Maps as sho
     * {@link CloudBalancingMapBasedEasyScoreCalculator#calculateScore(CloudBalance)}.
     */
    public HardSoftScore calculateScore(CloudBalance cloudBalance) {
        int hardScore = 0;
        int softScore = 0;
        for (CloudComputer computer : cloudBalance.getComputerList()) {
            int cpuPowerUsage = 0;
            int memoryUsage = 0;
            int networkBandwidthUsage = 0;
            boolean used = false;

            // Calculate usage
            for (CloudProcess process : cloudBalance.getProcessList()) {
                if (computer.equals(process.getComputer())) {
                    cpuPowerUsage += process.getRequiredCpuPower();
                    memoryUsage += process.getRequiredMemory();
                    networkBandwidthUsage += process.getRequiredNetworkBandwidth();
                    used = true;
                }
            }

            // Hard constraints
            int cpuPowerAvailable = computer.getCpuPower() - cpuPowerUsage;
            if (cpuPowerAvailable < 0) {
                hardScore += cpuPowerAvailable;
            }
            int memoryAvailable = computer.getMemory() - memoryUsage;
            if (memoryAvailable < 0) {
                hardScore += memoryAvailable;
            }
            int networkBandwidthAvailable = computer.getNetworkBandwidth() - networkBandwidthUs
            if (networkBandwidthAvailable < 0) {
                hardScore += networkBandwidthAvailable;
            }

            // Soft constraints
            if (used) {
                softScore -= computer.getCost();
            }
        }
        return HardSoftScore.valueOf(hardScore, softScore);
    }

}
```

Even if we optimize the code above to use `Map`s to iterate through the `processList` only once, *it is still slow* because it does not do incremental score calculation. To fix that, either use incremental Java score calculation or Drools score calculation.

2.5.2. Drools Score Configuration

Drools score calculation uses incremental calculation, where every score constraint is written as one or more score rules.

Using the Drools rule engine for score calculation, allows you to integrate with other Drools technologies, such as decision tables (XLS or web based), the KIE Workbench, ...

Prerequisite To use the Drools rule engine as a score function, simply add a `scoreDrl` resource in the classpath:

```
<scoreDirectorFactory>
  <scoreDrl>org/optaplanner/examples/cloudbalancing/solver/cloudBalancingScoreRules.drl</scoreDrl>
</scoreDirectorFactory>
```

1. We want to make sure that all computers have enough CPU, RAM and network bandwidth to support all their processes, so we make these hard constraints:

Example 7. cloudBalancingScoreRules.drl - Hard Constraints

```
...

import org.optaplanner.examples.cloudbalancing.domain.CloudBalance;
import org.optaplanner.examples.cloudbalancing.domain.CloudComputer;
import org.optaplanner.examples.cloudbalancing.domain.CloudProcess;

global HardSoftScoreHolder scoreHolder;

// ############################################################################
// Hard constraints
// ############################################################################

rule "requiredCpuPowerTotal"
    when
        $computer : CloudComputer($cpuPower : cpuPower)
        accumulate(
            CloudProcess(
                computer == $computer,
                $requiredCpuPower : requiredCpuPower);
            $requiredCpuPowerTotal : sum($requiredCpuPower);
            $requiredCpuPowerTotal > $cpuPower
        )
    then
        scoreHolder.addHardConstraintMatch(kcontext, $cpuPower - $requiredCpuPowerTotal);
end

rule "requiredMemoryTotal"
    ...
end

rule "requiredNetworkBandwidthTotal"
    ...
end
```

2. If those constraints are met, we want to minimize the maintenance cost, so we add that as a soft constraint:

Example 8. cloudBalancingScoreRules.drl - Soft Constraints

```
// ############################################################################
// Soft constraints
// ############################################################################

rule "computerCost"
    when
        $computer : CloudComputer($cost : cost)
        exists CloudProcess(computer == $computer)
    then
        scoreHolder.addSoftConstraintMatch(kcontext, - $cost);
```

WORKSHOP SEARCH REASONING
KIE OptaPlanner Deep Dive – Cloud Computer Balancing

Example 4. CloudBalancingHelloWorld.java

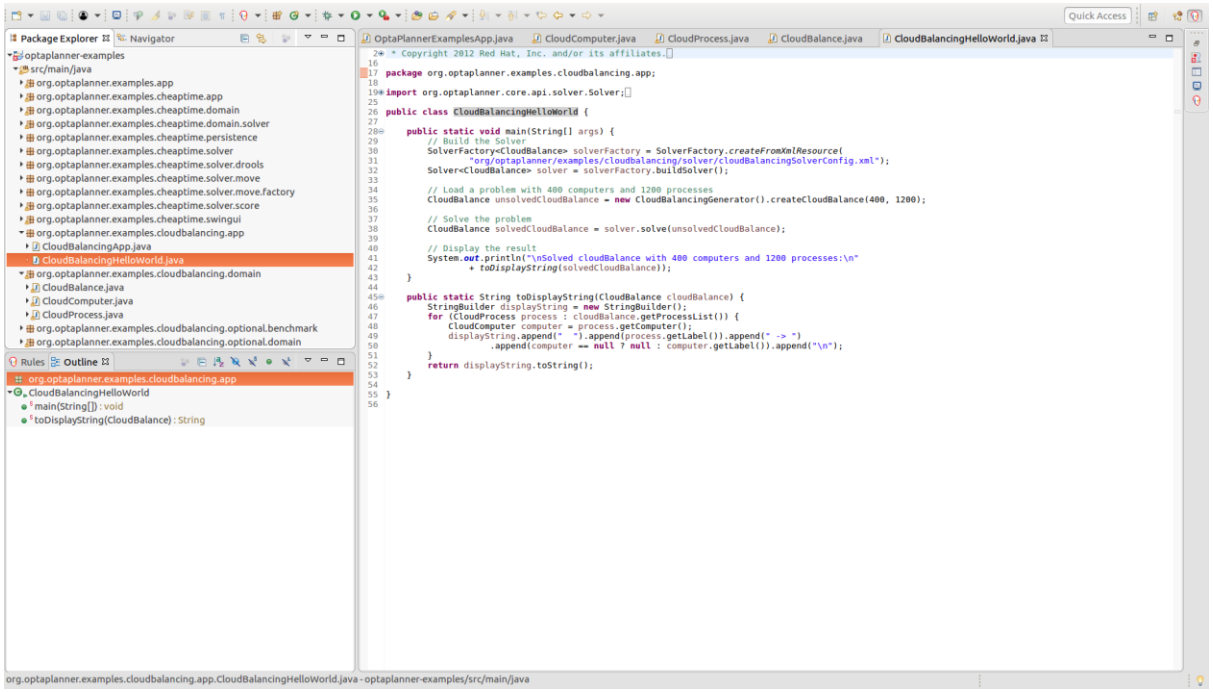
```
public class CloudBalancingHelloWorld {

    public static void main(String[] args) {
        // Build the Solver
        SolverFactory<CloudBalance> solverFactory = SolverFactory.createFromXmlResource(
                "org/optaplanner/examples/cloudbalancing/solver/cloudBalancingSolverConfig.xml"
        Solver<CloudBalance> solver = solverFactory.buildSolver();

        // Load a problem with 400 computers and 1200 processes
        CloudBalance unsolvedCloudBalance = new CloudBalancingGenerator().createCloudBalance(40

        // Solve the problem
        CloudBalance solvedCloudBalance = solver.solve(unsolvedCloudBalance);

        // Display the result
        System.out.println("\nSolved cloudBalance with 400 computers and 1200 processes:\n"
                + toDisplayString(solvedCloudBalance));
    }

    ...
}
```

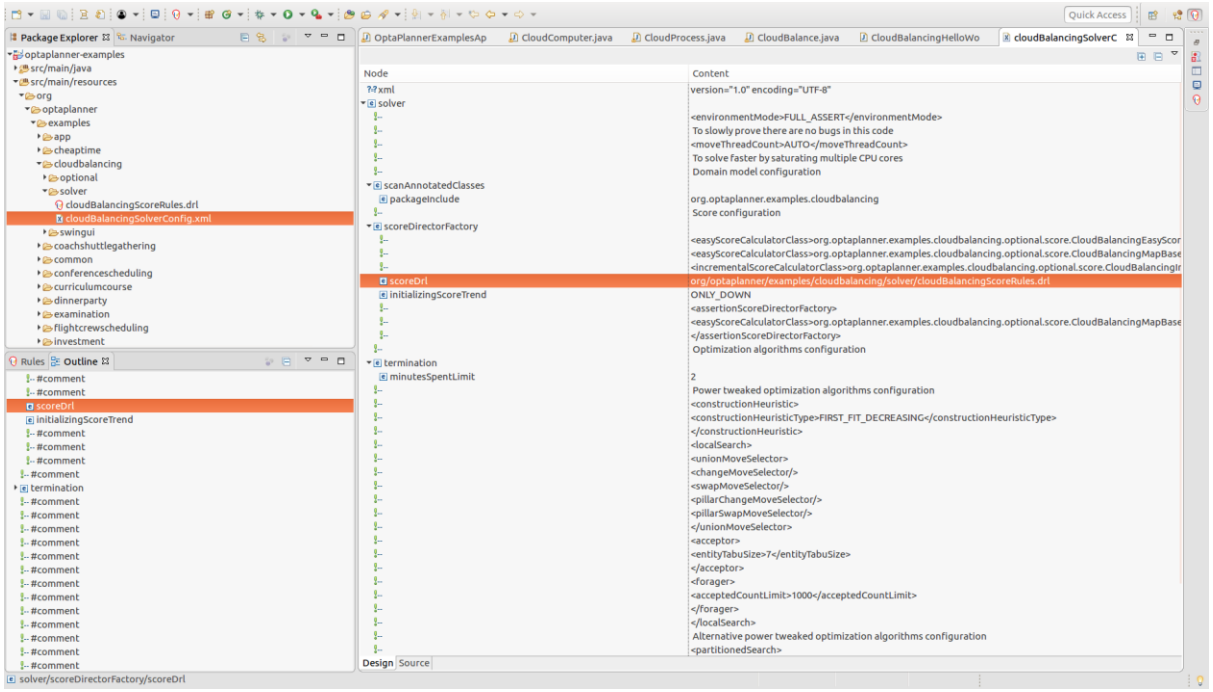
WORKSHOP SEARCH REASONING
KIE OptaPlanner Deep Dive – Cloud Computer Balancing

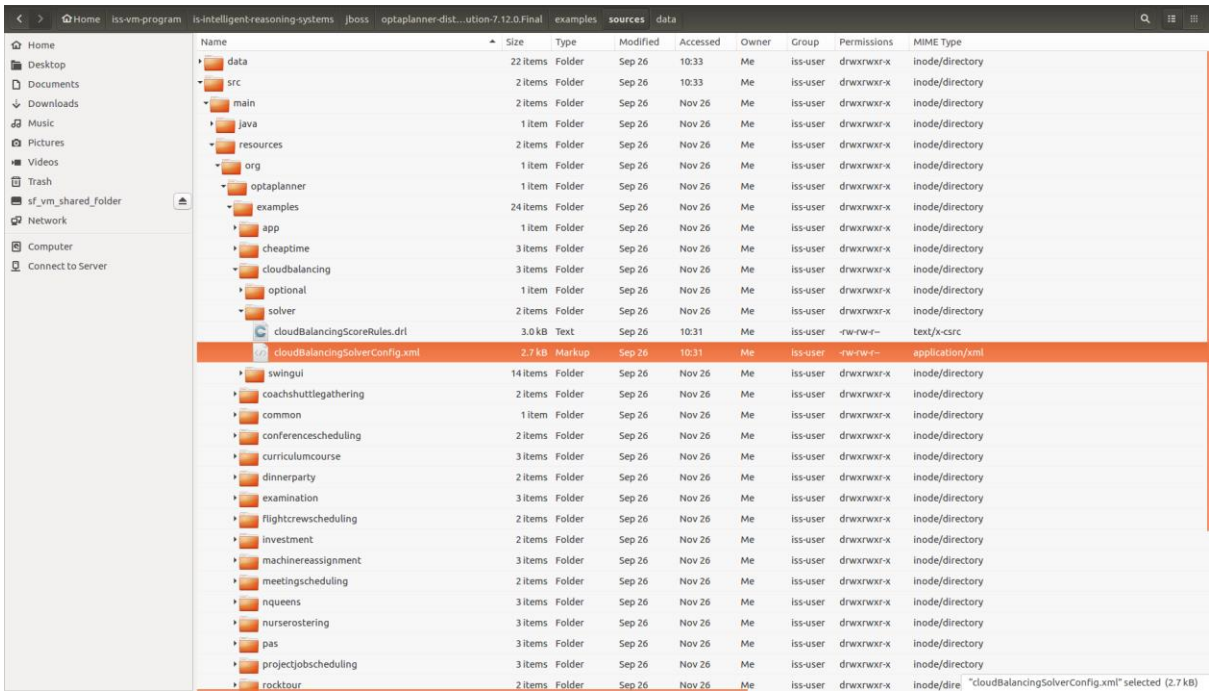
The solver configuration file determines how the solving process works; it is considered a part of the code. The file is named `cloudBalancingSolverConfig.xml`.

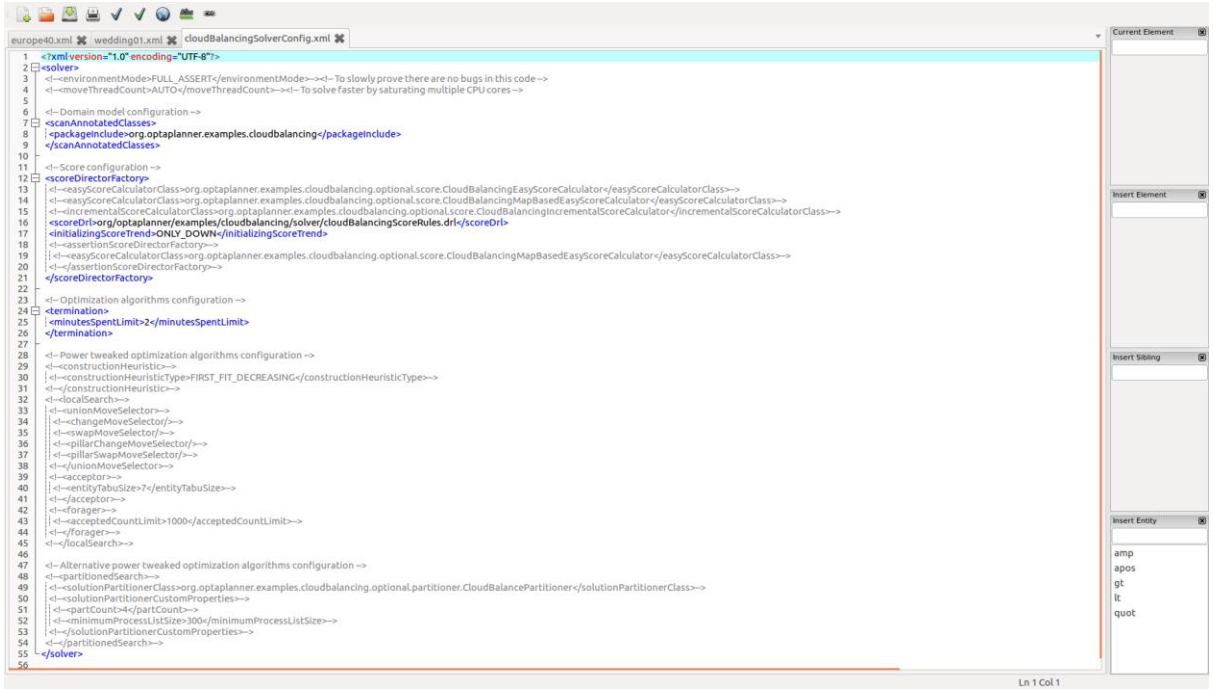
Example 5. cloudBalancingSolverConfig.xml

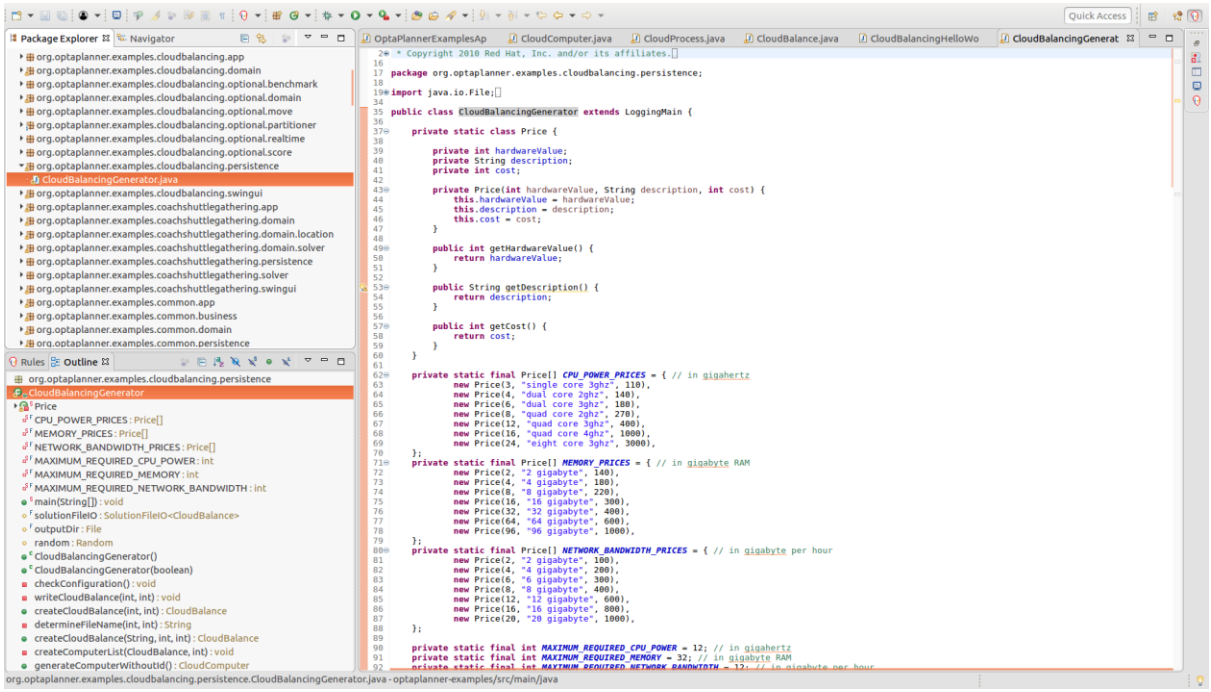
```
<?xml version="1.0" encoding="UTF-8"?>
<solver>
  <!-- Domain model configuration -->
  <scanAnnotatedClasses/>

  <!-- Score configuration -->
  <scoreDirectorFactory>
    <easyScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalan
    <!--<scoreDrl>org/optaplanner/examples/cloudbalancing/solver/cloudBalancingScoreRules.drl</
  </scoreDirectorFactory>

  <!-- Optimization algorithms configuration -->
  <termination>
    <secondsSpentLimit>30</secondsSpentLimit>
  </termination>
</solver>
```

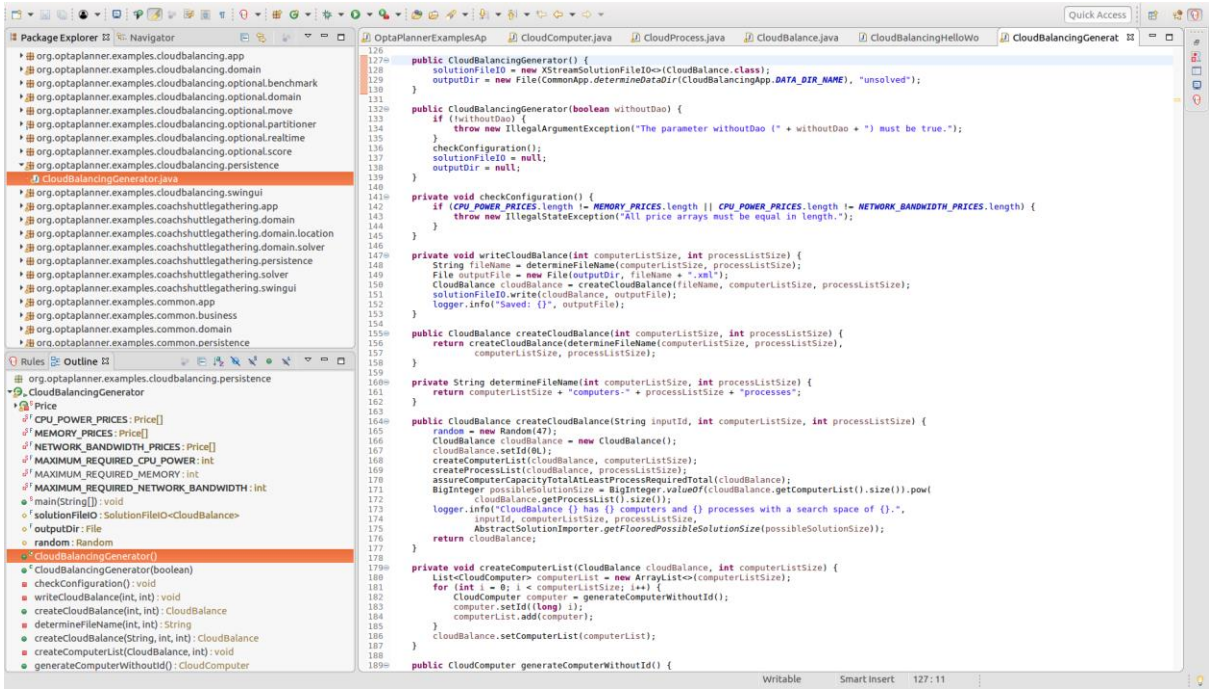

Package Explorer
Navigator
optaplanner-examples
src/main/java
org.optaplanner.examples.app
org.optaplanner.examples.cheaptime.app
org.optaplanner.examples.cheaptime.domain
org.optaplanner.examples.cheaptime.domain.solver
org.optaplanner.examples.cheaptime.persistence
org.optaplanner.examples.cheaptime.solver
org.optaplanner.examples.cheaptime.solver.drools
org.optaplanner.examples.cheaptime.solver.move
org.optaplanner.examples.cheaptime.solver.move.factory
org.optaplanner.examples.cheaptime.solver.score
org.optaplanner.examples.cheaptime.swingui
org.optaplanner.examples.cloudbalancing.app
CloudBalancingApp.java
CloudBalancingHelloWorld.java
org.optaplanner.examples.cloudbalancing.domain
CloudBalance.java
CloudComputer.java
CloudProcess.java
org.optaplanner.examples.cloudbalancing.optional.benchmark
org.optaplanner.examples.cloudbalancing.optional.domain
Rules
Outline
org.optaplanner.examples.cloudbalancing.app
CloudBalancingHelloWorld
main(String[]) : void
toDisplayString(CloudBalance) : String
OptaPlannerExamplesApp.java
CloudComputer.java
CloudProcess.java
CloudBalance.java
CloudBalancingHelloWorld.java
Quick Access
org.optaplanner.examples.cloudbalancing.app.CloudBalancingHelloWorld.java - optaplanner-examples/src/main/java

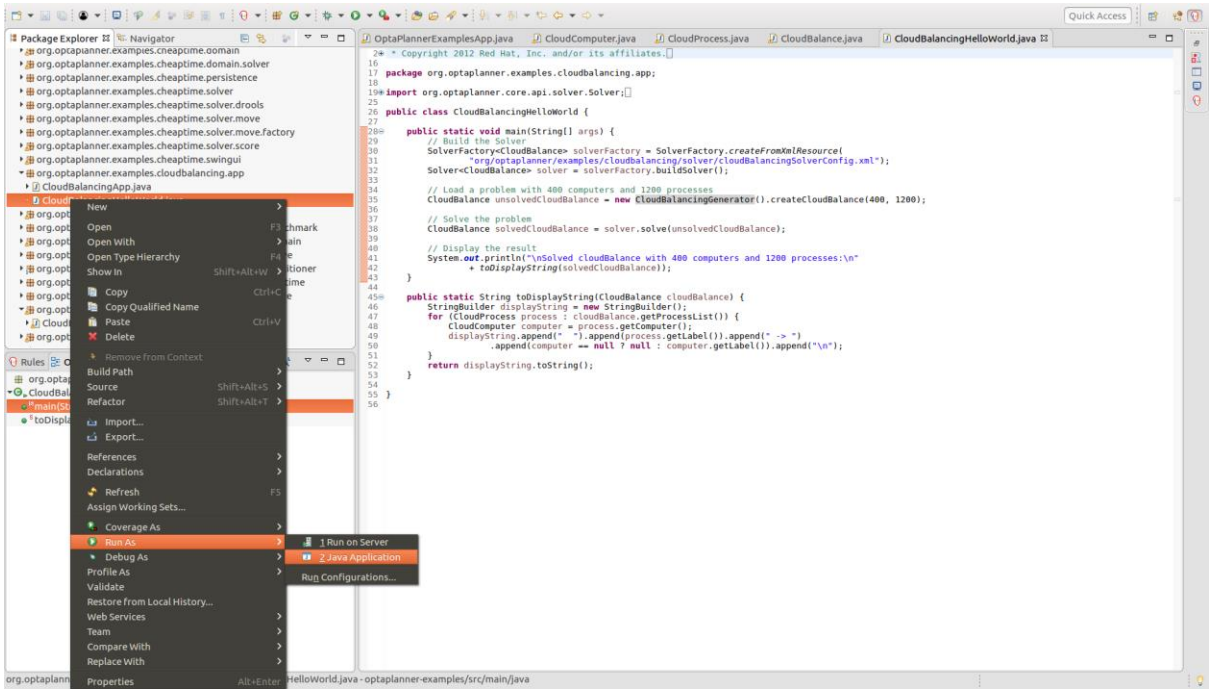

Package Explorer
Navigator
optaplanner-examples
src/main/java
src/main/resources
org
optaplanner
examples
app
cheaptime
cloudbalancing
optional
solver
cloudBalancingScoreRules.drl
cloudBalancingSolverConfig.xml
swingui
coachshuttlegathering
common
conferencescheduling
curriculumcourse
dinnerparty
examination
flightcrewscheduling
investment
Rules
Outline
#comment
scoreDrl
initializingScoreTrend
termination
minutesSpentLimit
OptaPlannerExamplesAp
CloudComputer.java
CloudProcess.java
CloudBalance.java
CloudBalancingHelloWo
cloudBalancingSolverC
Node
Content
?xml
version="1.0" encoding="UTF-8"
solver
<environmentMode>FULL_ASSERT</environmentMode>
To slowly prove there are no bugs in this code
<moveThreadCount>AUTO</moveThreadCount>
To solve faster by saturating multiple CPU cores
Domain model configuration
scanAnnotatedClasses
packageInclude
org.optaplanner.examples.cloudbalancing
Score configuration
scoreDirectorFactory
<easyScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancingEasyScor
<easyScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancingMapBase
<incrementalScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancingIr
scoreDrl
org/optaplanner/examples/cloudbalancing/solver/cloudBalancingScoreRules.drl
initializingScoreTrend
ONLY_DOWN
<assertionScoreDirectorFactory>
<easyScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancingMapBase
</assertionScoreDirectorFactory>
Optimization algorithms configuration
termination
minutesSpentLimit
2
Power tweaked optimization algorithms configuration
<constructionHeuristic>
<constructionHeuristicType>FIRST_FIT_DECREASING</constructionHeuristicType>
</constructionHeuristic>
<localSearch>
<unionMoveSelector>
<changeMoveSelector/>
<swapMoveSelector/>
<pillarChangeMoveSelector/>
<pillarSwapMoveSelector/>
</unionMoveSelector>
<acceptor>
<entityTabuSize>7</entityTabuSize>
</acceptor>
<forager>
<acceptedCountLimit>1000</acceptedCountLimit>
</forager>
</localSearch>
Alternative power tweaked optimization algorithms configuration
<partitionedSearch>
Design
Source
solver/scoreDirectorFactory/scoreDrl

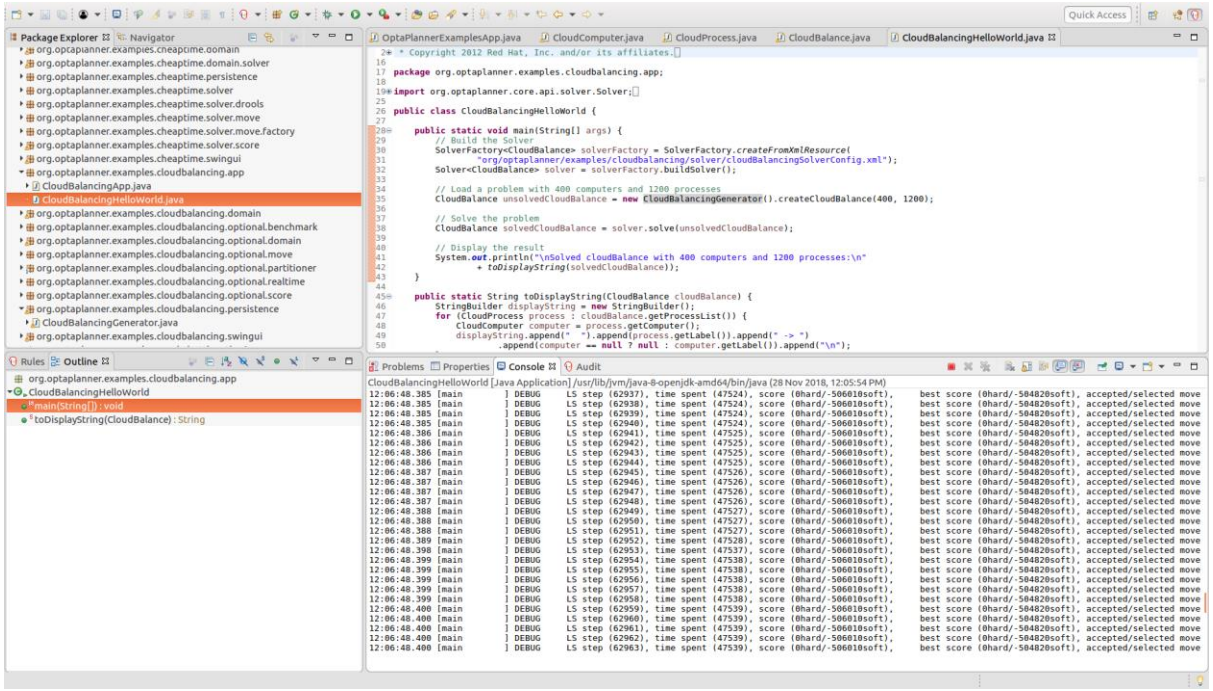

Home
iss-vm-program
is-intelligent-reasoning-systems
jboss
optaplanner-dist...ution-7.12.0.Final
examples
sources
data
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
sf_vm_shared_folder
Network
Computer
Connect to Server
Name
Size
Type
Modified
Accessed
Owner
Group
Permissions
MIME Type
data 22 items Folder Sep 26 10:33 Me iss-user drwxrwxr-x inode/directory
src 2 items Folder Sep 26 10:33 Me iss-user drwxrwxr-x inode/directory
main 2 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
java 1 item Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
resources 2 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
org 1 item Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
optaplanner 1 item Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
examples 24 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
app 1 item Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
cheaptime 3 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
cloudbalancing 3 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
optional 1 item Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
solver 2 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
cloudBalancingScoreRules.drl 3.0 kB Text Sep 26 10:31 Me iss-user -rw-rw-r-- text/x-csrc
cloudBalancingSolverConfig.xml 2.7 kB Markup Sep 26 10:31 Me iss-user -rw-rw-r-- application/xml
swingui 14 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
coachshuttlegathering 2 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
common 1 item Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
conferencescheduling 2 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
curriculumcourse 3 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
dinnerparty 2 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
examination 3 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
flightcrewscheduling 2 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
investment 2 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
machinereassignment 3 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
meetingscheduling 2 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
nqueens 3 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
nurserostering 3 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
pas 3 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
projectjobscheduling 3 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/directory
rocktour 2 items Folder Sep 26 Nov 26 Me iss-user drwxrwxr-x inode/dire
"cloudBalancingSolverConfig.xml" selected (2.7 kB)

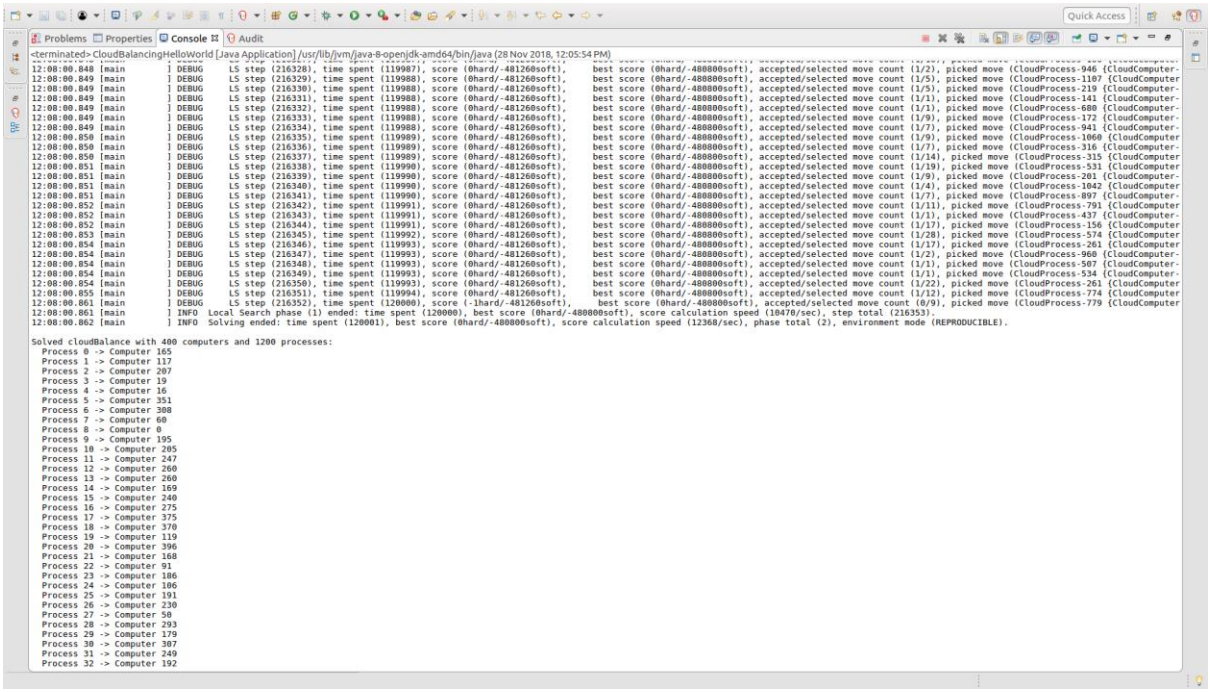

europe40.xml
wedding01.xml
cloudBalancingSolverConfig.xml
<?xml version="1.0" encoding="UTF-8"?>
<solver>
<!-- <environmentMode>FULL_ASSERT</environmentMode> --><!-- To slowly prove there are no bugs in this code -->
<!-- <moveThreadCount>AUTO</moveThreadCount> --><!-- To solve faster by saturating multiple CPU cores -->
<!-- Domain model configuration -->
<scanAnnotatedClasses>
<packageInclude>org.optaplanner.examples.cloudbalancing</packageInclude>
</scanAnnotatedClasses>
<!-- Score configuration -->
<scoreDirectorFactory>
<!-- <easyScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancingEasyScoreCalculator</easyScoreCalculatorClass> -->
<!-- <easyScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancingMapBasedEasyScoreCalculator</easyScoreCalculatorClass> -->
<!-- <incrementalScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancingIncrementalScoreCalculator</incrementalScoreCalculatorClass> -->
<scoreDrl>org/optaplanner/examples/cloudbalancing/solver/cloudBalancingScoreRules.drl</scoreDrl>
<initializingScoreTrend>ONLY_DOWN</initializingScoreTrend>
<!-- <assertionScoreDirectorFactory> -->
<!-- <easyScoreCalculatorClass>org.optaplanner.examples.cloudbalancing.optional.score.CloudBalancingMapBasedEasyScoreCalculator</easyScoreCalculatorClass> -->
<!-- </assertionScoreDirectorFactory> -->
</scoreDirectorFactory>
<!-- Optimization algorithms configuration -->
<termination>
<minutesSpentLimit>2</minutesSpentLimit>
</termination>
<!-- Power tweaked optimization algorithms configuration -->
<!-- <constructionHeuristic> -->
<!-- <constructionHeuristicType>FIRST_FIT_DECREASING</constructionHeuristicType> -->
<!-- </constructionHeuristic> -->
<!-- <localSearch> -->
<!-- <unionMoveSelector> -->
<!-- <changeMoveSelector/> -->
<!-- <swapMoveSelector/> -->
<!-- <pillarChangeMoveSelector/> -->
<!-- <pillarSwapMoveSelector/> -->
<!-- </unionMoveSelector> -->
<!-- <acceptor> -->
<!-- <entityTabuSize>7</entityTabuSize> -->
<!-- </acceptor> -->
<!-- <forager> -->
<!-- <acceptedCountLimit>1000</acceptedCountLimit> -->
<!-- </forager> -->
<!-- </localSearch> -->
<!-- Alternative power tweaked optimization algorithms configuration -->
<!-- <partitionedSearch> -->
<!-- <solutionPartitionerClass>org.optaplanner.examples.cloudbalancing.optional.partitioner.CloudBalancePartitioner</solutionPartitionerClass> -->
<!-- <solutionPartitionerCustomProperties> -->
<!-- <partCount>4</partCount> -->
<!-- <minimumProcessListSize>300</minimumProcessListSize> -->
<!-- </solutionPartitionerCustomProperties> -->
<!-- </partitionedSearch> -->
</solver>
Current Element
Insert Element
Insert Sibling
Insert Entity
amp
apos
gt
lt
quot
Ln 1 Col 1



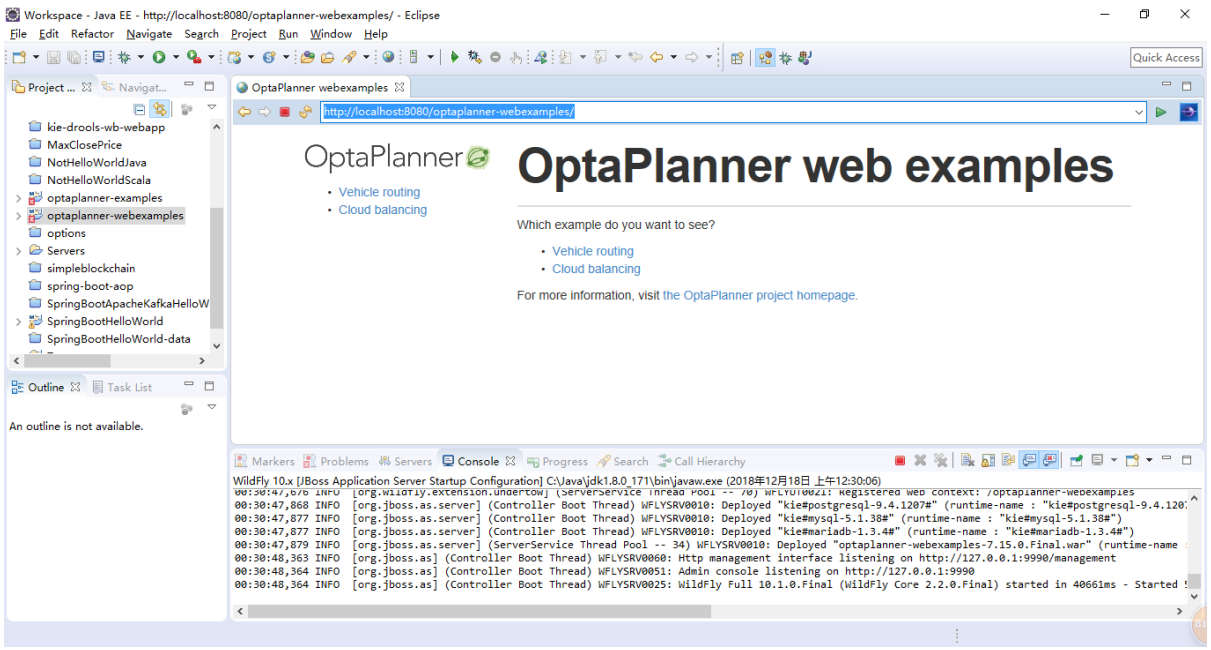



Package Explorer
Navigator
OptaPlannerExamplesApp.java
CloudComputer.java
CloudProcess.java
CloudBalance.java
CloudBalancingHelloWorld.java
Quick Access
New
Open
Open With
Open Type Hierarchy
Show In
Copy
Copy Qualified Name
Paste
Delete
Remove From Context
Build Path
Source
Refactor
Import...
Export...
References
Declarations
Refresh
Assign Working Sets...
Coverage As
Run As
Debug As
Profile As
Validate
Restore from Local History...
Web Services
Team
Compare With
Replace With
Properties
1 Run on Server
2 Java Application
Run Configurations...

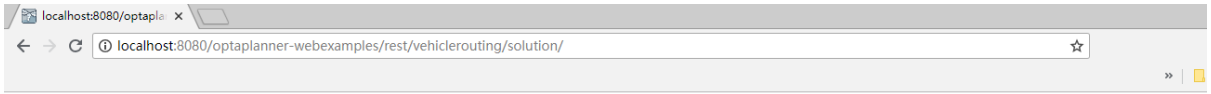

Package Explorer
Navigator
Outline
Problems
Properties
Console
Audit
CloudBalancingHelloWorld [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (28 Nov 2018, 12:05:54 PM)


Problems
Properties
Console
Audit
<terminated> CloudBalancingHelloWorld [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (28 Nov 2018, 12:05:54 PM)
12:08:00.848 [main] DEBUG LS step (216328), time spent (119987), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/2), picked move (CloudProcess-946 {CloudComputer-
12:08:00.849 [main] DEBUG LS step (216329), time spent (119988), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/5), picked move (CloudProcess-1107 {CloudComputer
12:08:00.849 [main] DEBUG LS step (216330), time spent (119988), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/5), picked move (CloudProcess-219 {CloudComputer-
12:08:00.849 [main] DEBUG LS step (216331), time spent (119988), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/1), picked move (CloudProcess-141 {CloudComputer-
12:08:00.849 [main] DEBUG LS step (216332), time spent (119988), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/1), picked move (CloudProcess-680 {CloudComputer-
12:08:00.849 [main] DEBUG LS step (216333), time spent (119988), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/9), picked move (CloudProcess-172 {CloudComputer-
12:08:00.849 [main] DEBUG LS step (216334), time spent (119988), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/7), picked move (CloudProcess-941 {CloudComputer-
12:08:00.850 [main] DEBUG LS step (216335), time spent (119989), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/9), picked move (CloudProcess-1060 {CloudComputer
12:08:00.850 [main] DEBUG LS step (216336), time spent (119989), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/7), picked move (CloudProcess-316 {CloudComputer-
12:08:00.850 [main] DEBUG LS step (216337), time spent (119989), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/14), picked move (CloudProcess-315 {CloudComputer
12:08:00.851 [main] DEBUG LS step (216338), time spent (119990), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/19), picked move (CloudProcess-531 {CloudComputer
12:08:00.851 [main] DEBUG LS step (216339), time spent (119990), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/9), picked move (CloudProcess-201 {CloudComputer-
12:08:00.851 [main] DEBUG LS step (216340), time spent (119990), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/4), picked move (CloudProcess-1042 {CloudComputer
12:08:00.851 [main] DEBUG LS step (216341), time spent (119990), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/7), picked move (CloudProcess-897 {CloudComputer-
12:08:00.852 [main] DEBUG LS step (216342), time spent (119991), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/11), picked move (CloudProcess-791 {CloudComputer
12:08:00.852 [main] DEBUG LS step (216343), time spent (119991), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/1), picked move (CloudProcess-437 {CloudComputer-
12:08:00.852 [main] DEBUG LS step (216344), time spent (119991), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/17), picked move (CloudProcess-156 {CloudComputer
12:08:00.853 [main] DEBUG LS step (216345), time spent (119992), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/28), picked move (CloudProcess-574 {CloudComputer
12:08:00.854 [main] DEBUG LS step (216346), time spent (119993), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/17), picked move (CloudProcess-261 {CloudComputer
12:08:00.854 [main] DEBUG LS step (216347), time spent (119993), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/2), picked move (CloudProcess-960 {CloudComputer-
12:08:00.854 [main] DEBUG LS step (216348), time spent (119993), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/1), picked move (CloudProcess-507 {CloudComputer-
12:08:00.854 [main] DEBUG LS step (216349), time spent (119993), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/1), picked move (CloudProcess-534 {CloudComputer-
12:08:00.854 [main] DEBUG LS step (216350), time spent (119993), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/22), picked move (CloudProcess-261 {CloudComputer
12:08:00.855 [main] DEBUG LS step (216351), time spent (119994), score (0hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (1/12), picked move (CloudProcess-774 {CloudComputer
12:08:00.861 [main] DEBUG LS step (216352), time spent (120000), score (-1hard/-481260soft), best score (0hard/-480800soft), accepted/selected move count (0/9), picked move (CloudProcess-779 {CloudComputer
12:08:00.861 [main] INFO Local Search phase (1) ended: time spent (120000), best score (0hard/-480800soft), score calculation speed (10470/sec), step total (216353).
12:08:00.862 [main] INFO Solving ended: time spent (120001), best score (0hard/-480800soft), score calculation speed (12368/sec), phase total (2), environment mode (REPRODUCIBLE).
Solved cloudBalance with 400 computers and 1200 processes:
Process 0 -> Computer 165
Process 1 -> Computer 117
Process 2 -> Computer 207
Process 3 -> Computer 19
Process 4 -> Computer 16
Process 5 -> Computer 351
Process 6 -> Computer 308
Process 7 -> Computer 60
Process 8 -> Computer 0
Process 9 -> Computer 195
Process 10 -> Computer 205
Process 11 -> Computer 247
Process 12 -> Computer 260
Process 13 -> Computer 260
Process 14 -> Computer 169
Process 15 -> Computer 240
Process 16 -> Computer 275
Process 17 -> Computer 375
Process 18 -> Computer 370
Process 19 -> Computer 119
Process 20 -> Computer 396
Process 21 -> Computer 168
Process 22 -> Computer 91
Process 23 -> Computer 186
Process 24 -> Computer 106
Process 25 -> Computer 191
Process 26 -> Computer 230
Process 27 -> Computer 50
Process 28 -> Computer 293
Process 29 -> Computer 179
Process 30 -> Computer 307
Process 31 -> Computer 249
Process 32 -> Computer 192

Solver in Java / Eclipse (Web Application on KIE Server)



Above: example web applications



{"name":"belgium-road-time-n50-k10","customerList":[{"locationName":"ANTHISNES","latitude":50.4812987,"longitude":5.5198048,"demand":15},{"locationName":"AVE-ET-AUFFE","latitude":50.1085676,"longitude":5.1429583,"demand":16},{"locationName":"BAVIKHOVE","latitude":50.8753581,"longitude":3.3113744,"demand":31},{"locationName":"BERSILLIES-L'ABBAYE","latitude":50.2624949,"longitude":4.1523037,"demand":14},{"locationName":"BLERET","latitude":50.688785,"longitude":5.2859129,"demand":33},
{"locationName":"BOURLERS","latitude":50.025208,"longitude":4.3409814,"demand":7},{"locationName":"BULSKAMP","latitude":51.0432467,"longitude":2.6501345,"demand":9},
{"locationName":"CHIEVRES","latitude":50.5661351,"longitude":3.7853169,"demand":17},{"locationName":"DAMME","latitude":51.2365829,"longitude":3.3400765,"demand":33},
{"locationName":"DONSTIENNES","latitude":50.2849958,"longitude":4.3106966,"demand":26},{"locationName":"ELVERDINGE","latitude":50.8848669,"longitude":2.8162074,"demand":18},
{"locationName":"EVELETTE","latitude":50.411922,"longitude":5.173705,"demand":10},{"locationName":"FONTAINE-L'EVEQUE","latitude":50.4100558,"longitude":4.3249526,"demand":32},
{"locationName":"GELINDEN","latitude":50.7670559,"longitude":5.2628934,"demand":26},{"locationName":"GONDREGNIES","latitude":50.6271045,"longitude":3.9115699,"demand":14},
{"locationName":"GUIGNIES","latitude":50.5499808,"longitude":3.3727285,"demand":32},{"locationName":"HAREN_BRUSSEL","latitude":50.8919578,"longitude":4.4182942,"demand":33},
{"locationName":"HEPPIGNIES","latitude":50.4814121,"longitude":4.4932598,"demand":30},{"locationName":"HOFSTADE_BT.","latitude":50.9912912,"longitude":4.492735,"demand":25},
{"locationName":"HUMBEEK","latitude":50.9667658,"longitude":4.3834362,"demand":18},{"locationName":"JURBISE","latitude":50.5208191,"longitude":3.9017078,"demand":26},
{"locationName":"KOMEN","latitude":50.77614,"longitude":3.007366,"demand":2},{"locationName":"LANDEGEM","latitude":51.0566021,"longitude":3.573045,"demand":25},
{"locationName":"LES_BONS_VILLERS","latitude":50.5427462,"longitude":4.4476802,"demand":10},{"locationName":"LISSEWEGE","latitude":51.2948381,"longitude":3.1994146,"demand":3},
{"locationName":"MAFFLE","latitude":50.6200815,"longitude":3.8018102,"demand":28},{"locationName":"MAULDE","latitude":50.6165736,"longitude":3.5470846,"demand":9},
{"locationName":"MERKSPLAS","latitude":51.361787,"longitude":4.861625,"demand":21},{"locationName":"MONT_NAM_","latitude":50.3534706,"longitude":4.9014191,"demand":7},
{"locationName":"NAAST","latitude":50.5504316,"longitude":4.0964803,"demand":24},{"locationName":"NIEUWKERKEN-WAAS","latitude":51.1936164,"longitude":4.1787815,"demand":1},
{"locationName":"OLLIGNIES","latitude":50.6876172,"longitude":3.8574228,"demand":5},{"locationName":"ORET","latitude":50.2998711,"longitude":4.6155651,"demand":19},
{"locationName":"PEPINGEN","latitude":50.7360287,"longitude":4.1360863,"demand":4},{"locationName":"POUPEHAN","latitude":49.8122243,"longitude":5.0029,"demand":9},
{"locationName":"RENINGE","latitude":50.9477884,"longitude":2.7891722,"demand":7},{"locationName":"ROSMEER","latitude":50.8460854,"longitude":5.5750235,"demand":16},{"locationName":"SAINT-MARTIN","latitude":50.5009515,"longitude":4.6473657,"demand":27},{"locationName":"SELANGE","latitude":49.6096472,"longitude":5.8560275,"demand":15},{"locationName":"SINT-KWINTENS-LENNIK","latitude":50.8068661,"longitude":4.1538281,"demand":17},{"locationName":"SOMZEE","latitude":50.2950667,"longitude":4.4831638,"demand":20},
{"locationName":"TAILLES","latitude":50.2273877,"longitude":5.7437507,"demand":27},{"locationName":"TILFF","latitude":50.5695196,"longitude":5.5844061,"demand":33},{"locationName":"VAUX-LEZ-ROSIERES","latitude":49.9102794,"longitude":5.5659053,"demand":18},{"locationName":"VILLERS-L'EVEQUE","latitude":50.7044254,"longitude":5.4399729,"demand":31},
{"locationName":"WOSSEM","latitude":50.8346073,"longitude":4.557627,"demand":19},{"locationName":"WARNANT-DREYE","latitude":50.5951652,"longitude":5.2276287,"demand":24},
{"locationName":"WETTEREN","latitude":51.0005742,"longitude":3.8693544,"demand":16},{"locationName":"XHENDELESSE","latitude":50.6061517,"longitude":5.7830075,"demand":32}],"vehicleRouteList":
[{"depotLocationName":"BRUSSEL","depotLatitude":50.8427501,"depotLongitude":4.3515499,"hexColor":"#8ae234","capacity":125,"demandTotal":123,"customerList":
[{"locationName":"HAREN_BRUSSEL","latitude":50.8919578,"longitude":4.4182942,"demand":33},{"locationName":"BLERET","latitude":50.688785,"longitude":5.2859129,"demand":33},{"locationName":"VILLERS-L'EVEQUE","latitude":50.7044254,"longitude":5.4399729,"demand":31},{"locationName":"GELINDEN","latitude":50.7670559,"longitude":5.2628934,"demand":26}]},
{"depotLocationName":"BRUSSEL","depotLatitude":50.8427501,"depotLongitude":4.3515499,"hexColor":"#fce94f","capacity":125,"demandTotal":125,"customerList":
[{"locationName":"LES_BONS_VILLERS","latitude":50.5427462,"longitude":4.4476802,"demand":10},{"locationName":"HEPPIGNIES","latitude":50.4814121,"longitude":4.4932598,"demand":30},
{"locationName":"SOMZEE","latitude":50.2950667,"longitude":4.4831638,"demand":20},{"locationName":"ORET","latitude":50.2998711,"longitude":4.6155651,"demand":19},{"locationName":"SAINT-MARTIN","latitude":50.5009515,"longitude":4.6473657,"demand":27},{"locationName":"WOSSEM","latitude":50.8346073,"longitude":4.557627,"demand":19}]},
{"depotLocationName":"BRUSSEL","depotLatitude":50.8427501,"depotLongitude":4.3515499,"hexColor":"#729fcf","capacity":125,"demandTotal":113,"customerList":
[{"locationName":"OLLIGNIES","latitude":50.6876172,"longitude":3.8574228,"demand":5},{"locationName":"MAULDE","latitude":50.6165736,"longitude":3.5470846,"demand":9},
{"locationName":"GUIGNIES","latitude":50.5499808,"longitude":3.3727285,"demand":32},{"locationName":"BAVIKHOVE","latitude":50.8753581,"longitude":3.3113744,"demand":31},

Above: example solution enquiry via RESTful API

index.jsp | leaflet.jsp | VehicleRoutingSolver... | DefaultVehicleRouting... | VehicleRoutingRestSe...

```
/**
 * @see DefaultVehicleRoutingRestService
 */
@Path("/vehiclerouting")
public interface VehicleRoutingRestService {

    @GET
    @Path("/solution")
    @Produces("application/json")
    JsonVehicleRoutingSolution getSolution();

    @POST
    @Path("/solution/solve")
    @Produces("application/json")
    JsonMessage solve();

    @POST
    @Path("/solution/terminateEarly")
    @Produces("application/json")
    JsonMessage terminateEarly();
```

Above: example KIE server interaction via RESTful API

index.jsp | leaflet.jsp | VehicleRoutingSolver... | DefaultVehicleRouting... | VehicleRoutingRestSe... | CloudB

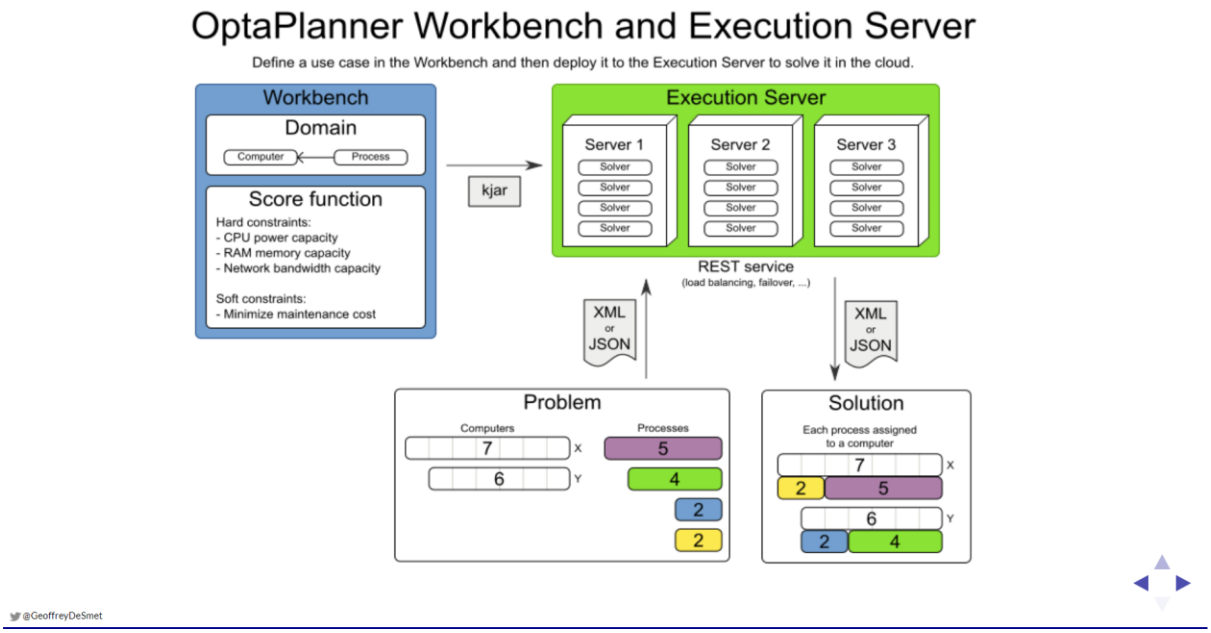
```
};

loadSolution = function() {
  $.ajax({
    url: "<%=application.getContextPath()%>/rest/vehiclerouting/solution",
    type: "GET",
    dataType : "json",
    success: function(solution) {
      var markers = [];
      $.each(solution.customerList, function(index, customer) {
        var customerIcon = L.divIcon({
          iconSize: new L.Point(20, 20),
          className: "vehicleRoutingCustomerMarker",
          html: "<span>" + customer.demand + "</span>"
        });
        var marker = L.marker([customer.latitude, customer.longitude], {icon: customerIcon});
        marker.addTo(map).bindPopup(customer.locationName + "</br>Deliver " + customer.demand + " items.");
        markers.push(marker);
      });
      map.fitBounds(L.featureGroup(markers).getBounds());
    }, error : function(jqXHR, textStatus, errorThrown) {ajaxError(jqXHR, textStatus, errorThrown)}
  });
};
```

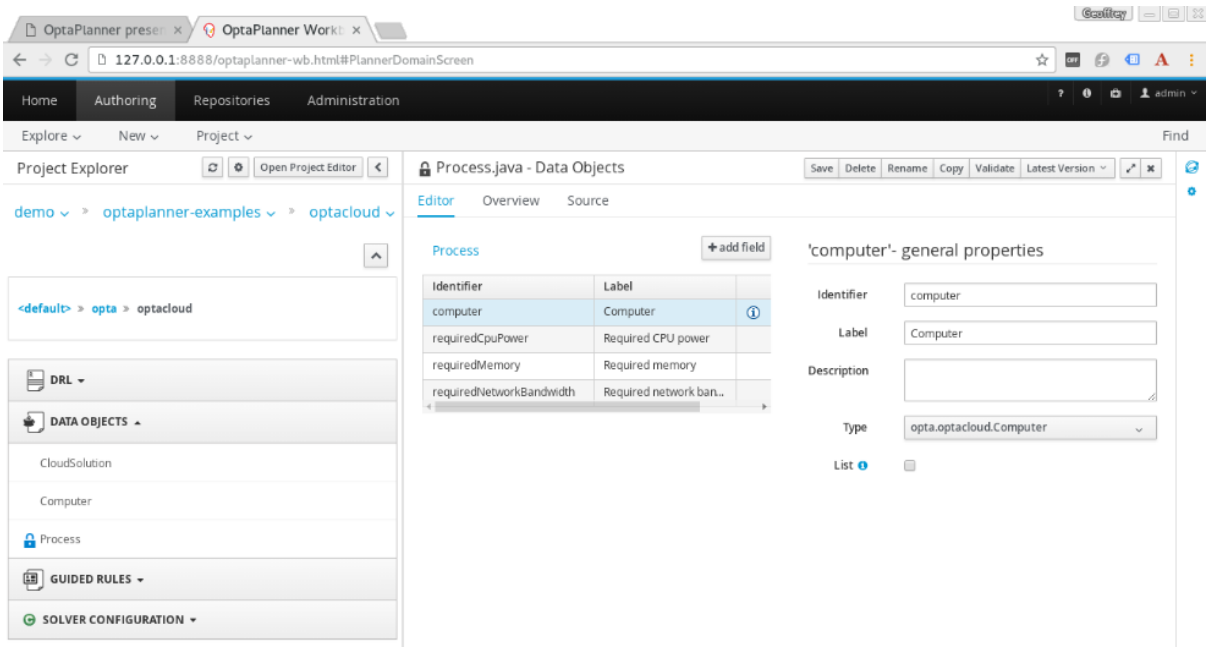
Above: example server end point

Reference

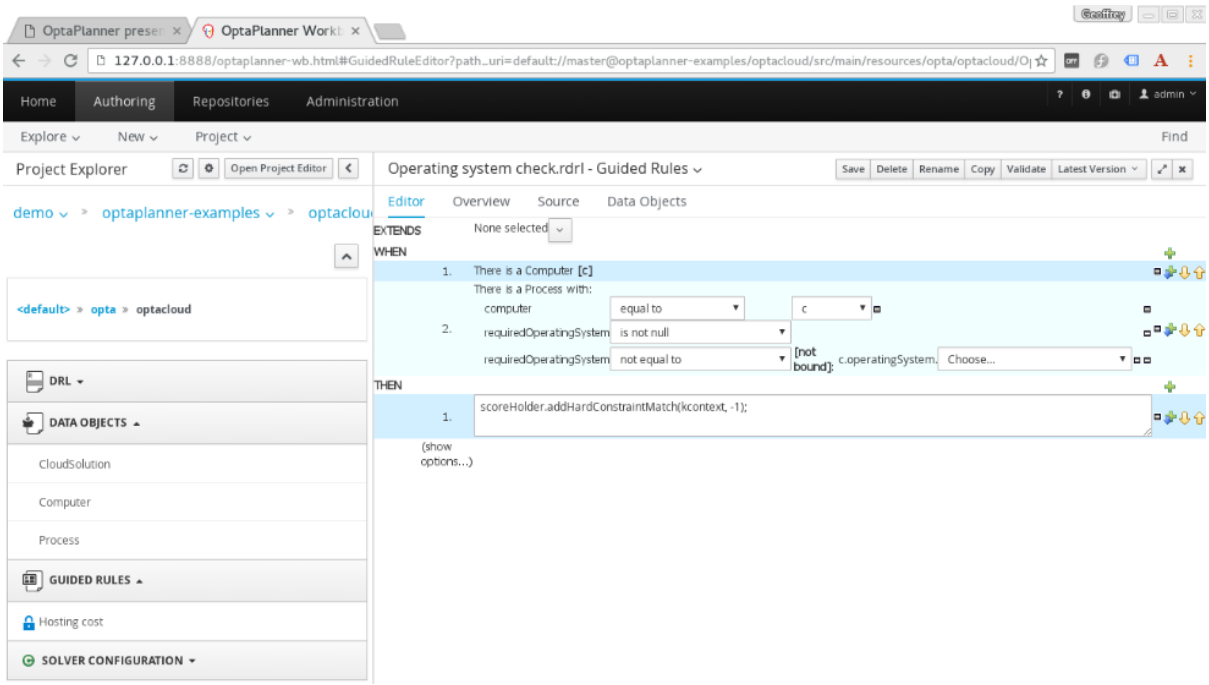
- https://github.com/kiegroup/optaplanner/tree/master/optaplanner-webexamples
- https://docs.optaplanner.org/latest/optaplanner-docs/html_single/index.html#quickStart

Solver in KIE Workbench & Server

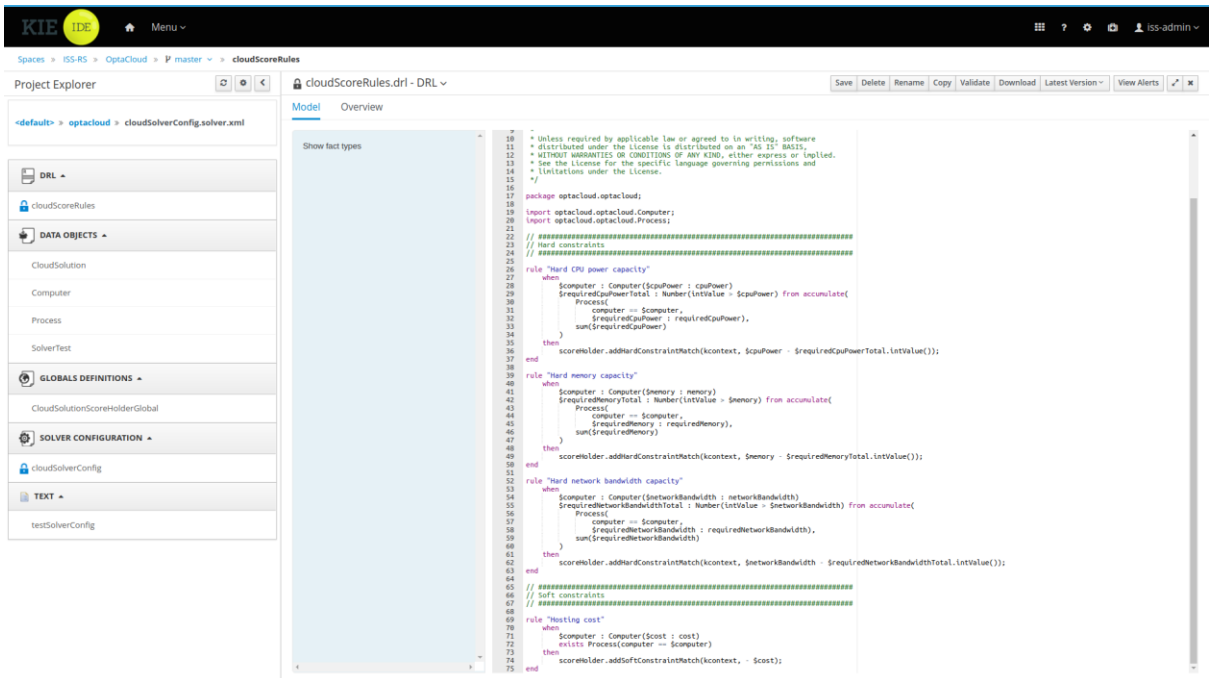




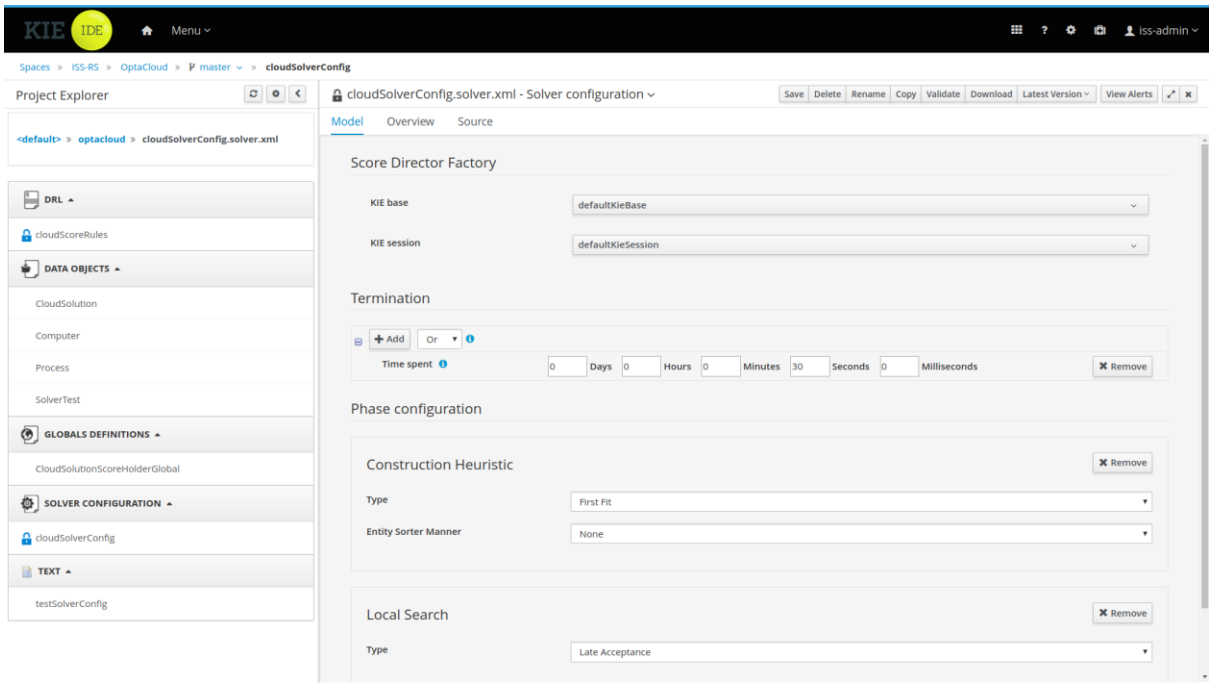
Above: domain objects based on class diagram



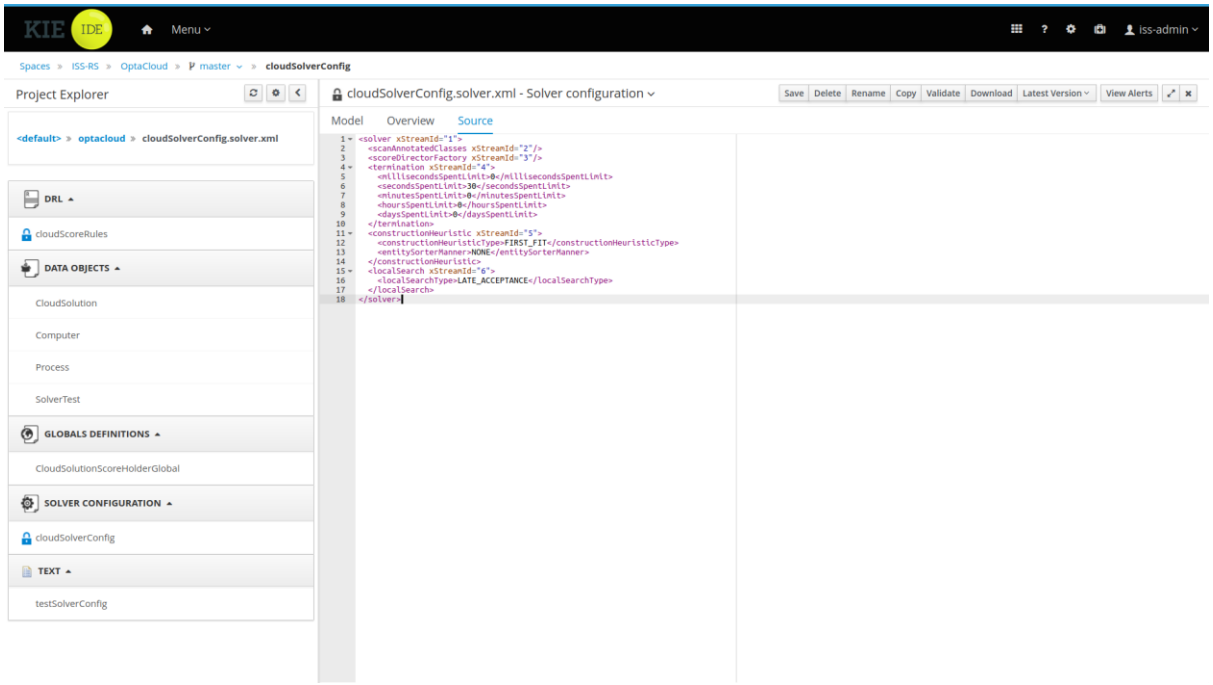
Above: constraints using Drools guided rule



Above: constraints using Drools guided rule



Above: solver configuration



Above: solver configuration (xml)

Reference

- http://www.optaplanner.org/learn/useCases/cloudOptimization.html
- http://www.optaplanner.org/learn/slides/optaplanner-presentation/training.html#/4/26
- https://docs.jboss.org/optaplanner/release/latestFinal/optaplanner-wb-es-docs/html_single/

Workshop 2.2 [Individual]

Choose one use case; propose relevant enhancements; then import, analyze, adapt, enhance, and solve/execute.

Enhancement example 1:
Enhance the Cloud Balancing system by introducing a comprehensive user interface for end user to

1. [Input] Key in problem configuration:
    a. Computer instances with: cpuPower, memory, networkBandwidth, cost.
    b. Process instances with: requiredCpuPower, requiredMemory, and requiredNetworkBandwidth.
2. [Output] Display graphical final solution of cloud computer assignments:
    a. Hint: reusable Java code from https://github.com/kiegroup/optaplanner/tree/master/optaplanner-examples/src/main/java/org/optaplanner/examples/cloudbalancing

Enhancement example 2:
Enrich the Cloud Balancing domain model and add extra constraints such as these:

- Each Process belongs to a Service. A computer might crash, so processes running the same service must be assigned to different computers.
- Each Computer is located in a Building. A building might burn down, so processes of the same services should (or must) be assigned to computers in different buildings.

Reference https://docs.optaplanner.org/latest/optaplanner-docs/html_single/index.html#cloudBalancingBeyondThisTutorial

Enhancement example 3:
Propose and implement/enhance your own relevant business use case.

Workshop 3 Guide

Refer to workshop by co-lecturer

Workshop 4 Guide

Refer to workshop by co-lecturer

Workshop 5 Guide

WORKSHOP CREATING HYBRID REASONING SYSTEM
EEP & MTech Stackable

- **KIE Minimum Viable Product (MVP) – Individual Work**
  - Determine a new business problem suitable for advanced reasoning techniques **OR** significantly enhance a previous workshop project
  - Plan, design, develop, integrate, test, and deploy bespoke hybrid reasoning system using components, e.g. KIE tools, OptaPlanner, Genetic algorithms, Orange, Data mining.
  - Prepare project report and user guide.
  - Submit project deliverables. Refer to Project Submission Template

☺ **Candidate Project: Hybrid Airport Gate Assignment System (HAGAS)**

http://ousar.lib.okayama-u.ac.jp/files/public/4/48534/20160528091554614463/K0004584_honbun.pdf

WORKSHOP CREATING HYBRID REASONING SYSTEM
MTech Thru-Train

- **KIE Minimum Viable Product (MVP) – Group Work**
  - Form a project team of 4-6 members, choose a team name, appoint a team leader.
  - Plan, design, develop, integrate, test, and deploy bespoke hybrid reasoning system using components, e.g. KIE tools, OptaPlanner, Genetic algorithms, Orange, Data mining.
  - Prepare project report and user guide.
  - Prepare system demo for video presentation.
  - Submit project deliverables. Refer to Project Submission Template

☺ **Candidate Project: Hybrid Airport Gate Assignment System (HAGAS)**

http://ousar.lib.okayama-u.ac.jp/files/public/4/48534/20160528091554614463/K0004584_honbun.pdf

Workshop 5.1 [EEP Individual / MTech Group]

Identify a relevant business scenario/problem. Propose and create a hybrid reasoning system.
The proposed group workshop project must develop, integrate, and demonstrate at least two out of following three technique groups:

1. Business rule **OR** Business process based reasoning techniques
2. Business resource optimization techniques: Heuristic search **OR** Constraint satisfaction **OR** Evolutional computing
3. Knowledge Discovery **OR** Data Mining techniques

The submitted runnable system should have a graphical user interface for end user to input or update data to execute different business scenarios, e.g. to enable initial planning and frequent re-planning/re-optimization. And to display system output results in a user friendly manner. (Input/output using xml file or console log is not considered as user friendly.)

Candidate/Example Project:
ANNEX 1 WORKSHOP PROJECT CANDIDATE

Project Submission Guideline:
ANNEX 2 PROJECT CODE EXPORT & IMPORT USING KIE WORKBENCH
ANNEX 3 WORKSHOP PROJECT SUBMISSION

Domain Modelling Reference

- https://docs.optaplanner.org/latest/optaplanner-docs/html_single/index.html#designPatterns

ANNEX 1 WORKSHOP PROJECT CANDIDATE

Workshop Project Candidate One

Hybrid Airport Gate Assignment System (HAGAS)

The Airport Gate Assignment Problem: Scheduling Algorithms and Simulation Approach
Ahmed Thanyan AL-Sultan

The rapid development of airlines has made airports busier and more complicated. The assignment of schedule to available gates is a major issue for daily airline operations. We consider the over-constrained airport gate assignment problem (AGAP) where the number of flights exceeds the number of available gates, and where the objectives are to minimize the number of ungated flights and the total walking distance or connection times. The procedures used in this project are to create a mathematical model formulation to identify decision variables to identify, constraints and objective functions. In addition, we will consider in the AGAP the size of each gate in the terminal and also the towing process for the aircraft. We will use a greedy algorithm and a Tabu search meta-heuristic to solve the problem and compare it with other scheduling methods. Actual and forecasted data will be simulated in the experiment. The greedy algorithm minimizes ungated flights while providing initial feasible solutions that allow flexibility in seeking good solutions, especially in case when flight schedules are dense in time. Experiments conducts give good results. The distance a passenger has to walk in any airport to reach various key areas, including departure gates, baggage belts and connecting flights provide for an important performance measure for the quality of any airport. While certain walking distances are fixed, others are dynamic. In particular, the distances traversed by passengers from check-in counters to gates and from gate to gate, in the case of transfer or connecting passengers, change according to how scheduled flights are assigned to gates. This allows for the ground handling agents and airlines, together with airport authorities, to dynamically assign airport gates to scheduled flights so as to minimize walking distances while, consequently, minimizing connection times. Which flight to gate assignment policy to be used so as to achieve such minimum times can be derived at the start of such planning day based on published flights schedules and booked passenger loads. The airport gate assignment problem (AGAP) seeks to find feasible flight to gate assignments so that total passenger connection times and walking distances is minimized. Distances that are taken into account are those from check-in to gates in the case of embarking or originating passengers, from gates to baggage claim areas (check-out) in the case of disembarking or destination passengers and from gate to gate in the case of transfer or connecting passengers. In the over-constrained case, where the number of aircraft exceeds the number of available gates, we include the distance from the apron or tarmac area to the terminal for aircraft assigned to these areas.

…

Reference

The Airport Gate Assignment Problem: Scheduling Algorithms and Simulation Approach, Ahmed Thanyan AL-Sultan, Graduate School of environmental science, March 2012
http://ousar.lib.okayama-u.ac.jp/files/public/4/48534/20160528091554614463/K0004584_honbun.pdf

Workshop Project Candidate Two

OptaPlanner Application Implementation in KIE Workbench & Server

Objective:
Construct a useful business reasoning system using KIE product suite, incorporating an OptaPlanner solver as an embedded optimization engine/task for automated machine reasoning.

Choose one OptaPlanner example application from below **ONLY**:

- Cloud balancing
- Course timetabling
- Vehicle routing with time windows
- Project job scheduling
- Exam timetabling
- Nurse rostering
- Cheap time scheduling
- Flight crew scheduling

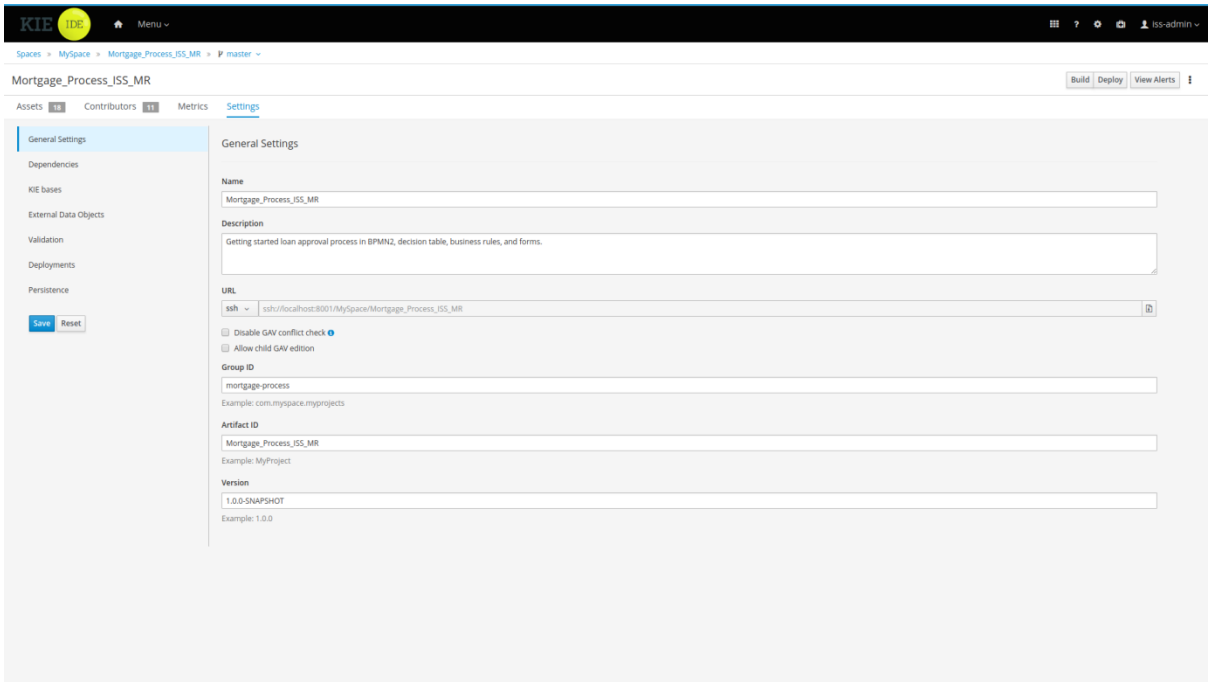
System Requirements:

- Make use of KIE BRMS (Drools) & BPMS (jBPM) capability;
- Define at least one practical business enhancement/requirement based on OptaPlanner example;
- Convert/migrate original Maven/Eclipse OptaPlanner project with enhancements into KIE (jBPM) Workbench project;
- Develop a (web-based) User Interface;
- Deploy the developed system onto KIE Server for use;

ANNEX 2 PROJECT CODE EXPORT & IMPORT USING KIE WORKBENCH

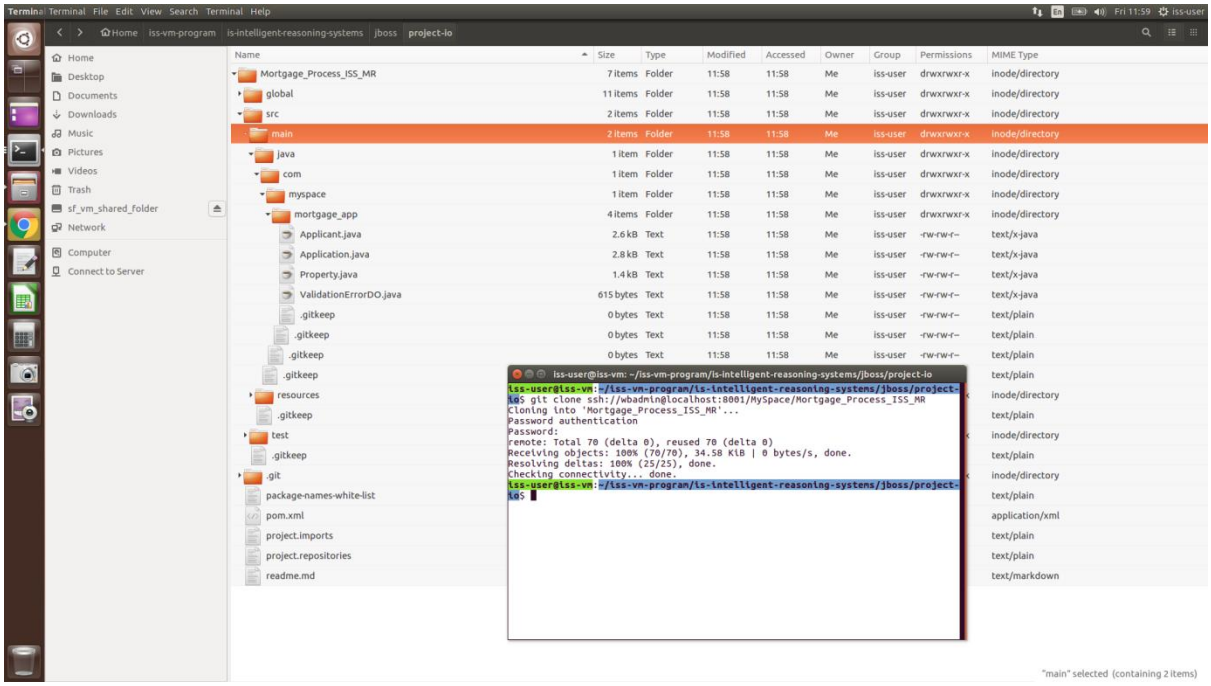
Example: export KIE project ***Mortgage_Process_ISS_MR*** from work space ***MySpace***

Review project settings to obtain project URL



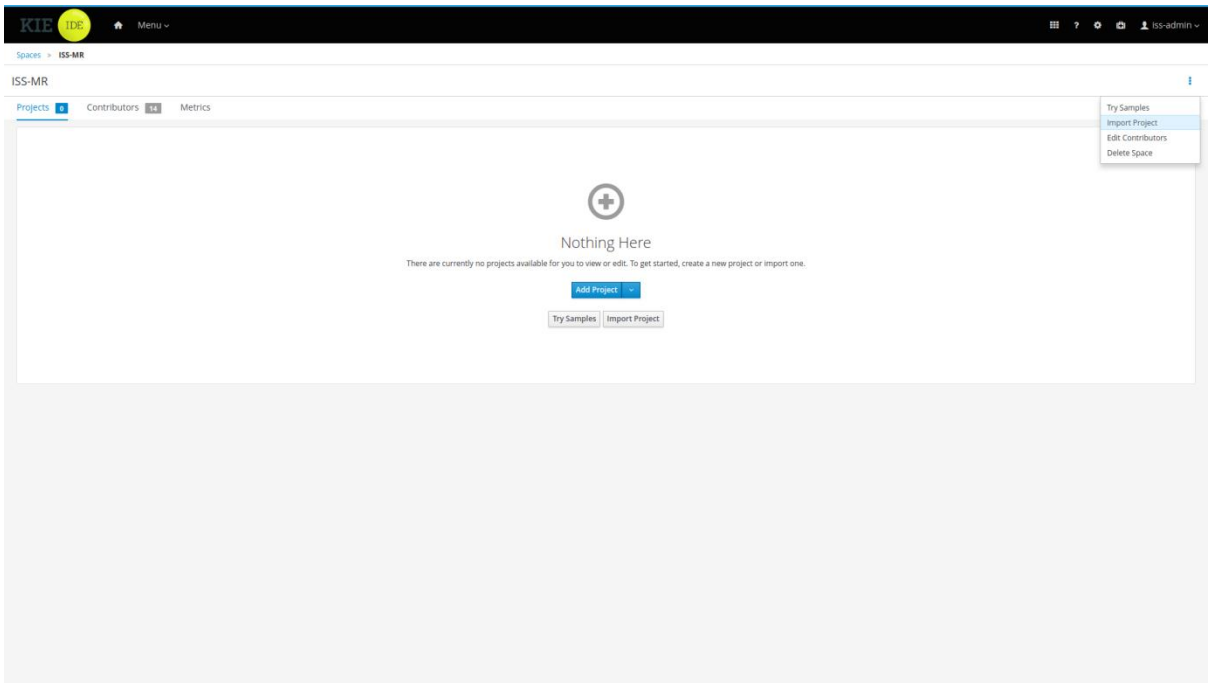
Export project from KIE Workbench

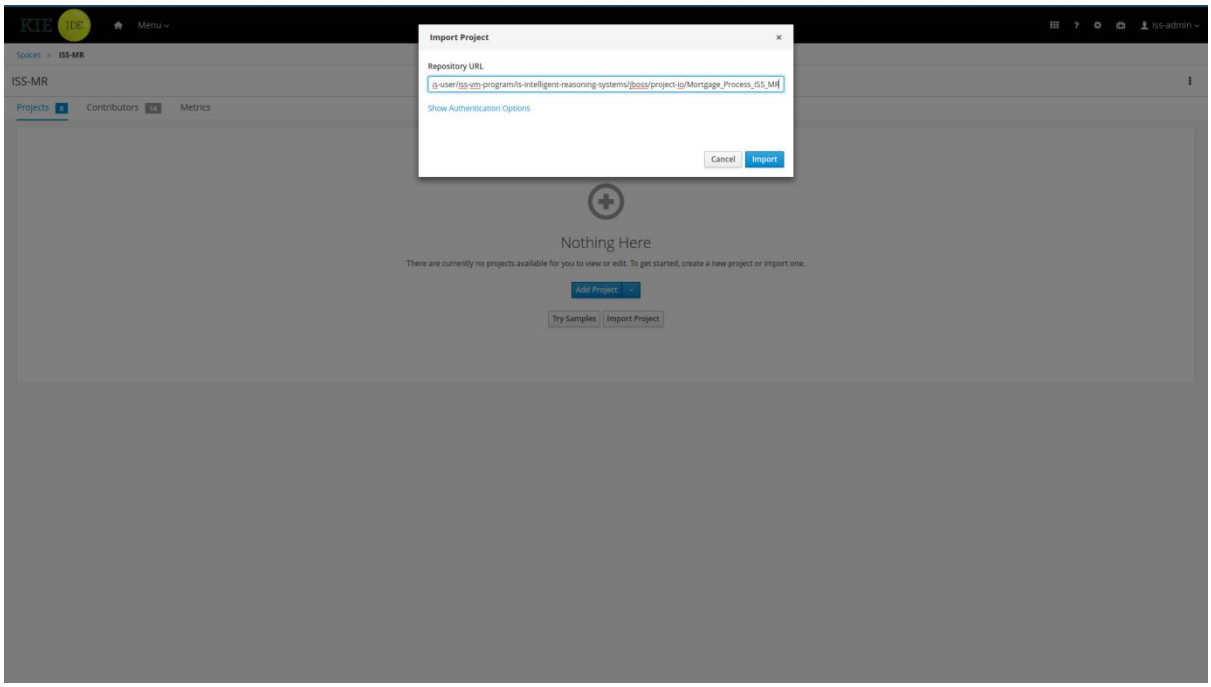
1. Select a folder for exporting, example here uses ***/home/iss-user/iss-vm-program/is-intelligent-reasoning-systems/jboss/project-io***
2. Start a Terminal there, key in command ***git clone ssh://wbadmin@localhost:8001/MySpace/Mortgage_Process_ISS_MR***
3. Key in password '***wbadmin***' for user wbadmin

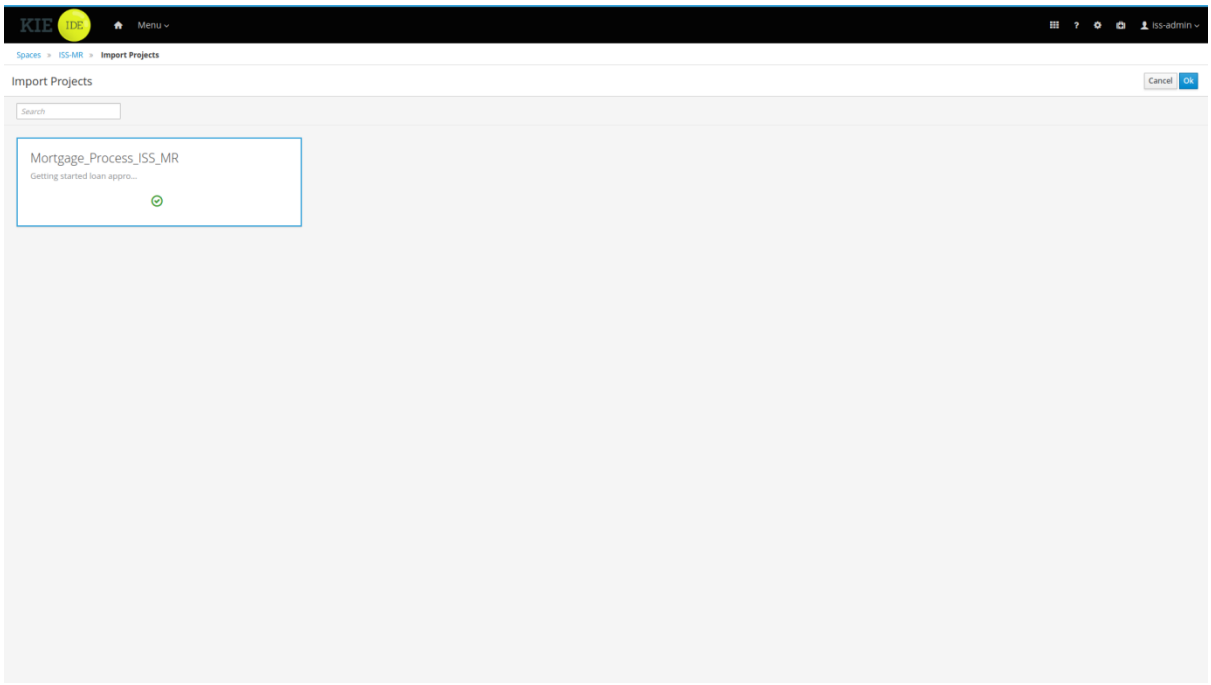


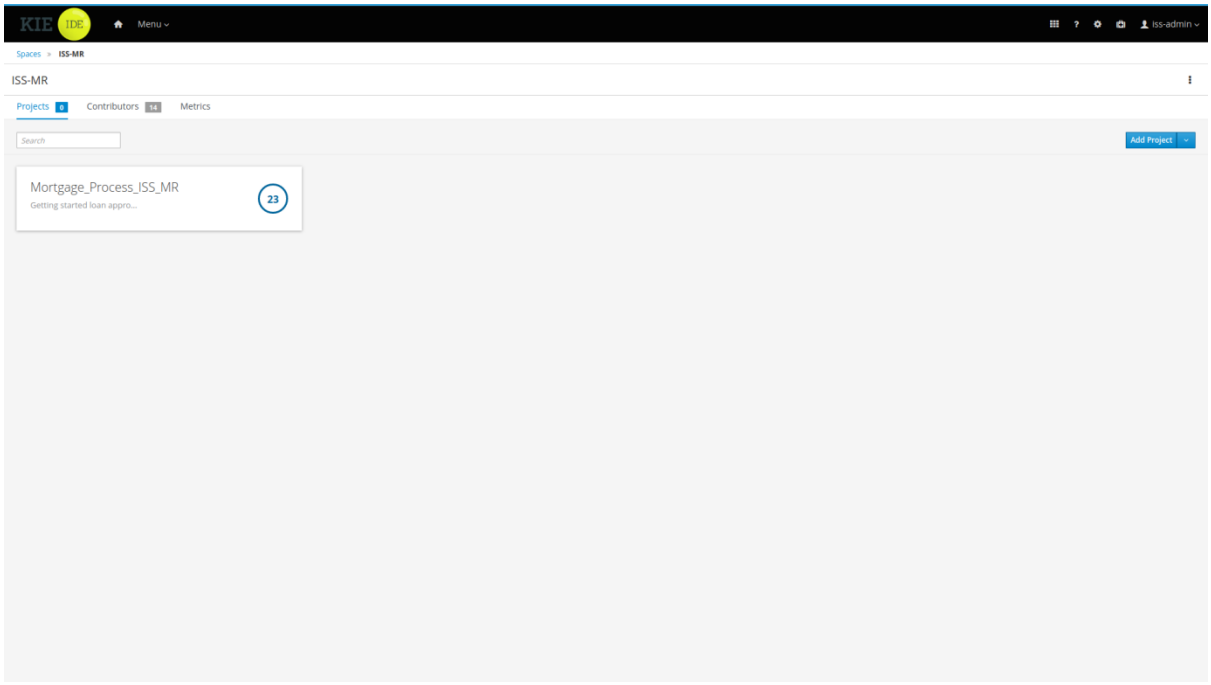
Import project into KIE Workbench

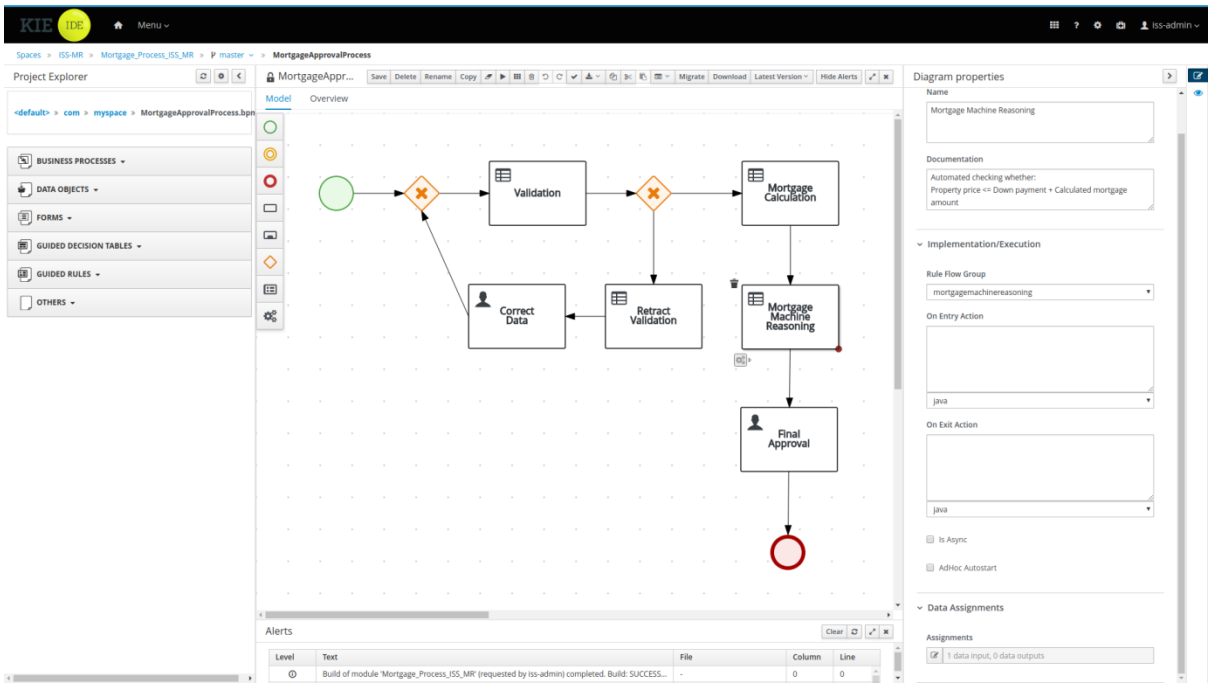
1. In KIE workbench, select/create a project Space, example here uses ***ISS-MR***
2. Click menu function '***Import Project***'
3. For Repository URL, key in ***file:///home/iss-user/iss-vm-program/is-intelligent-reasoning-systems/jboss/project-io/Mortgage_Process_ISS_MR***






KIE IDE
Menu
iss-admin
Spaces » ISS-MR » Import Projects
Import Projects
Cancel
Ok
Search
Mortgage_Process_ISS_MR
Getting started loan appro...


KIE IDE
Menu
iss-admin
Spaces » ISS-MR
ISS-MR
Projects 0
Contributors 14
Metrics
Search
Add Project
Mortgage_Process_ISS_MR
Getting started loan appro...
23



Reference

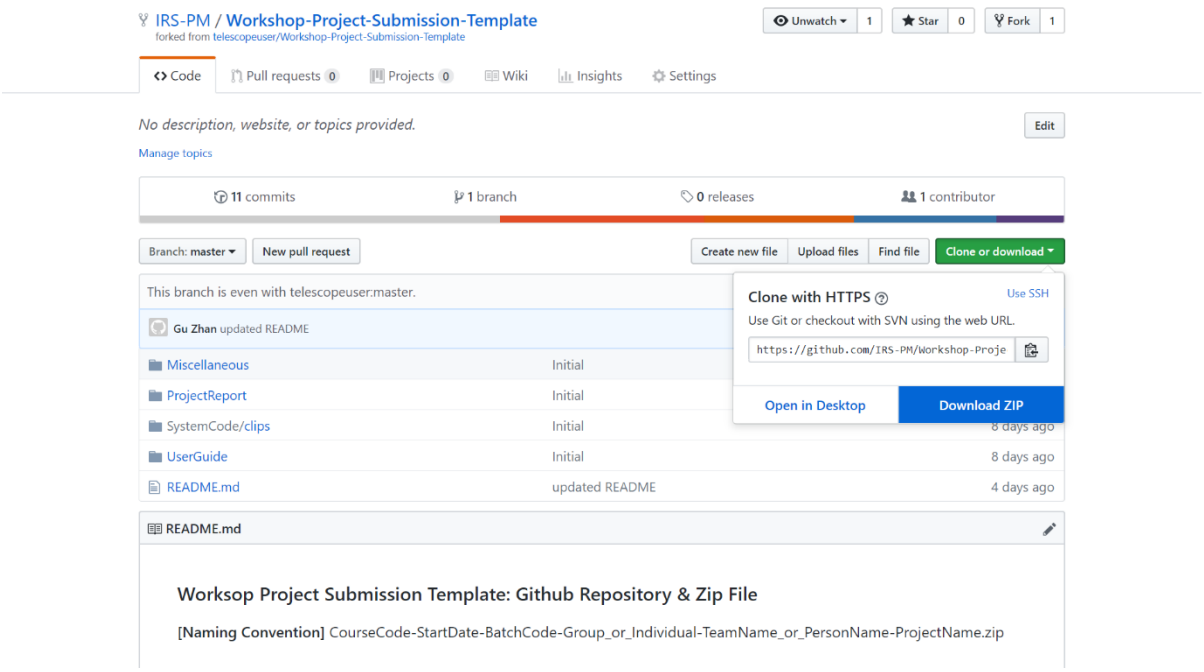
- https://developer.jboss.org/thread/269991
- https://developer.jboss.org/thread/237411
- https://developer.jboss.org/thread/252588

ANNEX 3 WORKSHOP PROJECT SUBMISSION

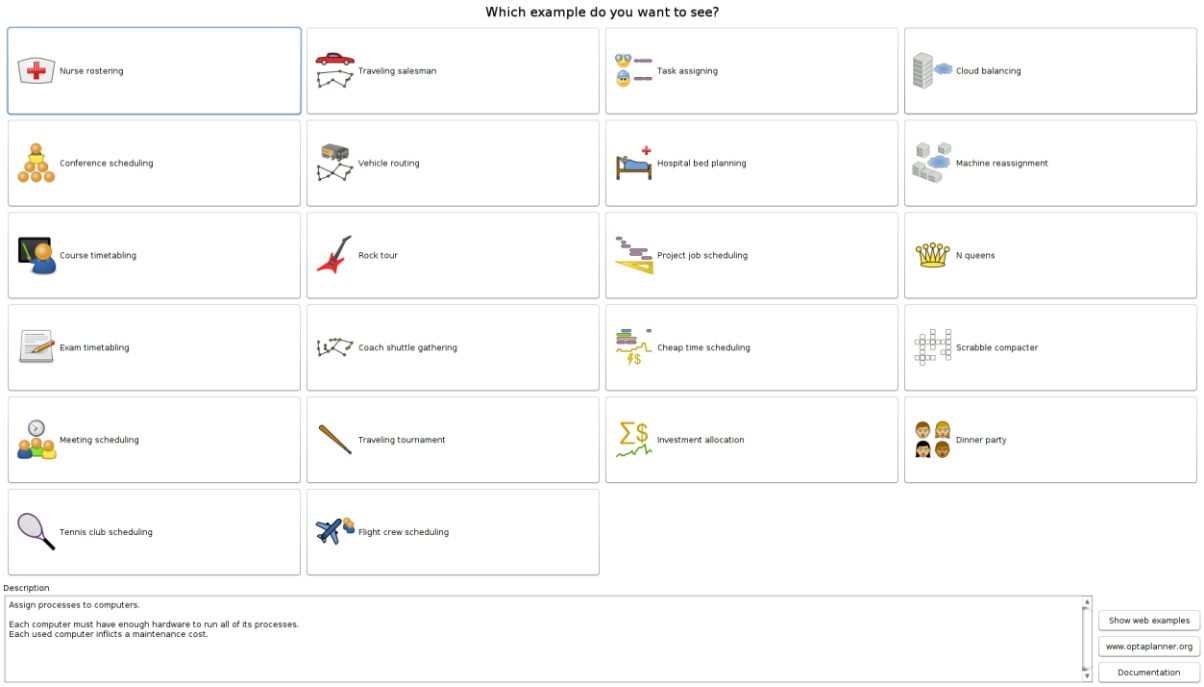
Submission due by 23:59 on last lecture date (+ 14)
One delayed day = 10 marks deduction

1. [MTech & EEP] Create Github repository for project submission
2. [MTech] Download Github repository as a ZIP file, then upload to NUS LumiNUS / IVLE

Reference https://github.com/IRS-PM/Workshop-Project-Submission-Template



ANNEX 4 KIE OptaPlanner Examples



Reference

- https://docs.optaplanner.org/latest/optaplanner-docs/html_single/index.html#useCasesAndExamples

ANNEX 4 : KIE OPTAPLANNER EXAMPLES

KIE OptaPlanner Deep Dive – Dinner Party

- **Business Scenario / Problem Description**
- **Miss Manners is throwing another dinner party.**
  - This time she invited 144 guests and prepared 12 round tables with 12 seats each.
  - Every guest should sit next to someone (left and right) of the opposite gender.
  - And that neighbour should have at least one hobby in common with the guest.
  - At every table, there should be two politicians, two doctors, two socialites, two coaches, two teachers and two programmers.
  - And the two politicians, two doctors, two coaches and two programmers should not be the same kind at a table.

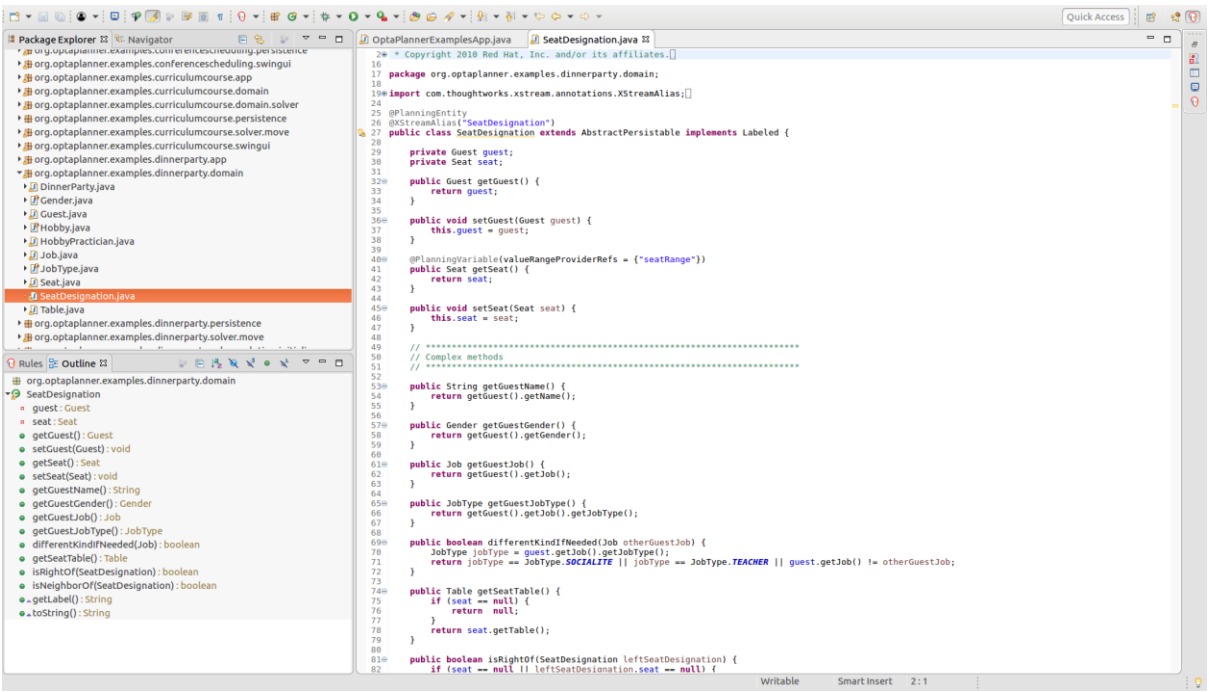
 271

ANNEX 4 : KIE OPTAPLANNER EXAMPLES

Exercise: Draw Class Diagram

273


Unsolved dataset shortcuts
wedding01
Solved dataset shortcuts
wedding01-score20
Import...
Open...
Save as...
Export as...
Solve
Table 0
Table 1
Table 2
Table 4
Table 5
Table 6
Table 8
Table 9
Table 10
Constraint matches
Latest best score: 20


Package Explorer
Navigator
OptaPlannerExamplesApp.java
SeatDesignation.java
Rules
Outline
org.optaplanner.examples.dinnerparty.domain
SeatDesignation
guest : Guest
seat : Seat
getGuest() : Guest
setGuest(Guest) : void
getSeat() : Seat
setSeat(Seat) : void
getGuestName() : String
getGuestGender() : Gender
getGuestJob() : Job
getGuestJobType() : JobType
differentKindIfNeeded(Job) : boolean
getSeatTable() : Table
isRightOf(SeatDesignation) : boolean
isNeighborOf(SeatDesignation) : boolean
getLabel() : String
toString() : String
Writable
Smart Insert
2 : 1

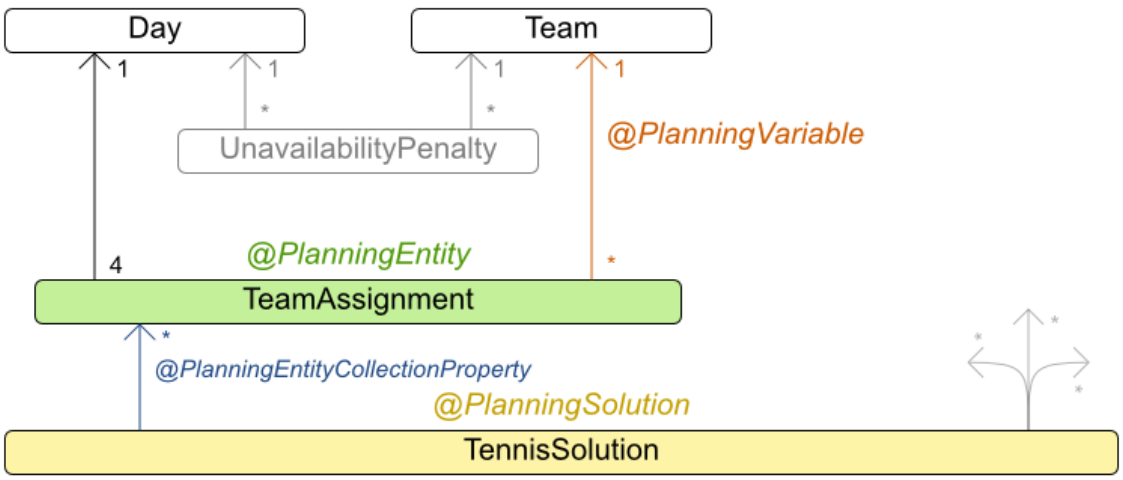
ANNEX 4 : KIE OPTAPLANNER EXAMPLES

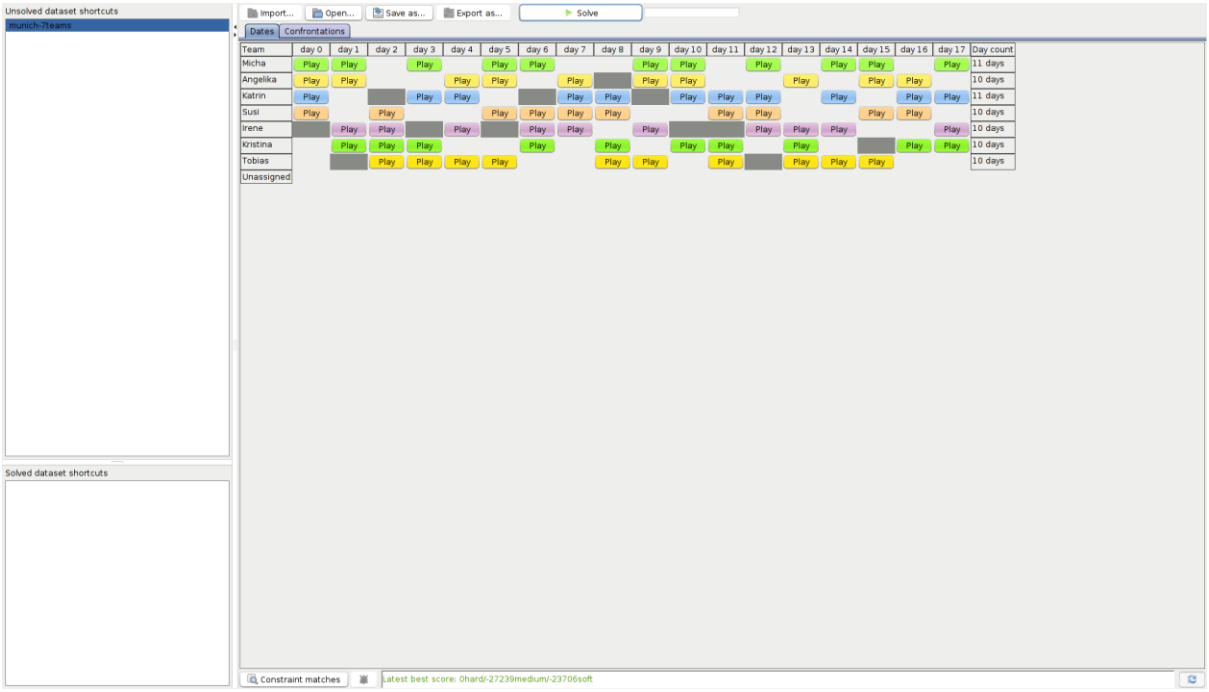
KIE OptaPlanner Deep Dive – Tennis Club Scheduling

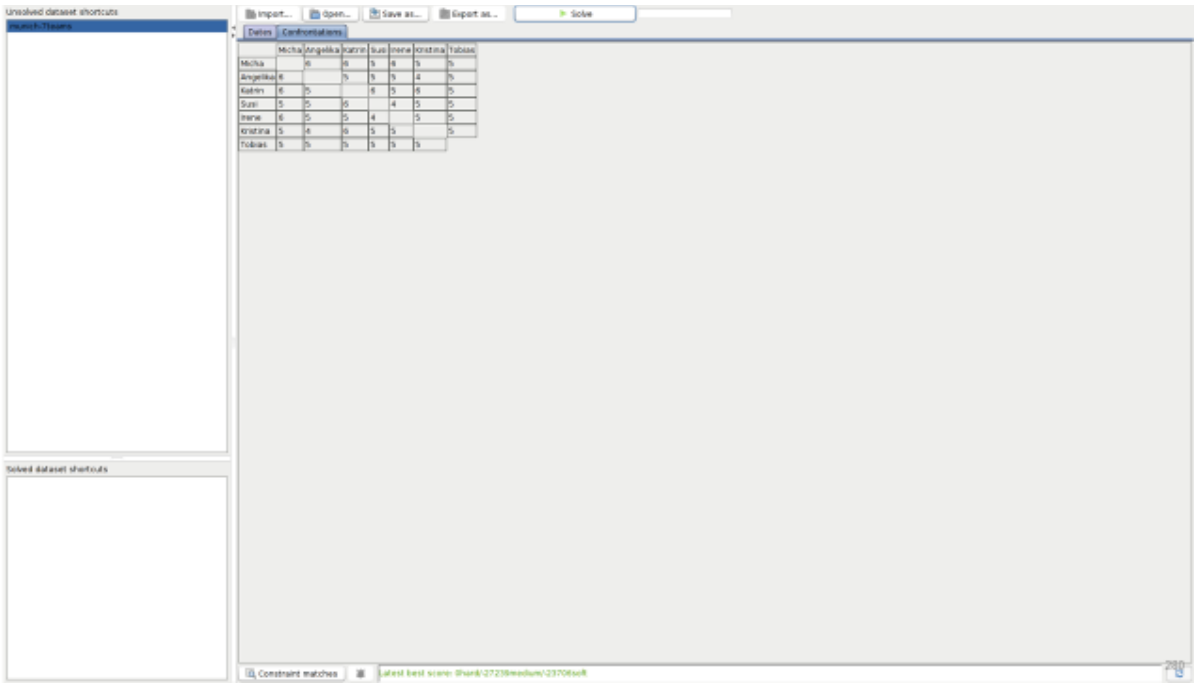
- **Business Scenario / Problem Description**
- **Every week the tennis club has four teams playing round robin against each other. Assign those four spots to the teams fairly.**
- **Hard constraints:**
  - Conflict: A team can only play once per day.
  - Unavailability: Some teams are unavailable on some dates.
- **Medium constraints:**
  - Fair assignment: All teams should play an (almost) equal number of times.
- **Soft constraints:**
  - Evenly confrontation: Each team should play against every other team an equal number of times.

 278

Tennis class diagram




Unsolved dataset shortcuts
munich-7teams
Solved dataset shortcuts
Import...
Open...
Save as...
Export as...
Solve
Dates
Confrontations
Team
day 0
day 1
day 2
day 3
day 4
day 5
day 6
day 7
day 8
day 9
day 10
day 11
day 12
day 13
day 14
day 15
day 16
day 17
Day count
Micha
Angelika
Katrin
Susi
Irene
Kristina
Tobias
Unassigned
Play
11 days
10 days
Constraint matches
Latest best score: 0hard/-27239medium/-23706soft


Unsolved dataset shortcuts
Solved dataset shortcuts
Import...
Open...
Save as...
Export as...
Solve
Dates
Confrontations
Micha
Angelika
Katrin
Susi
Irene
Kristina
Tobias
Constraint matches

ANNEX 4 : KIE OPTAPLANNER EXAMPLES

KIE OptaPlanner Deep Dive – Meeting Scheduling

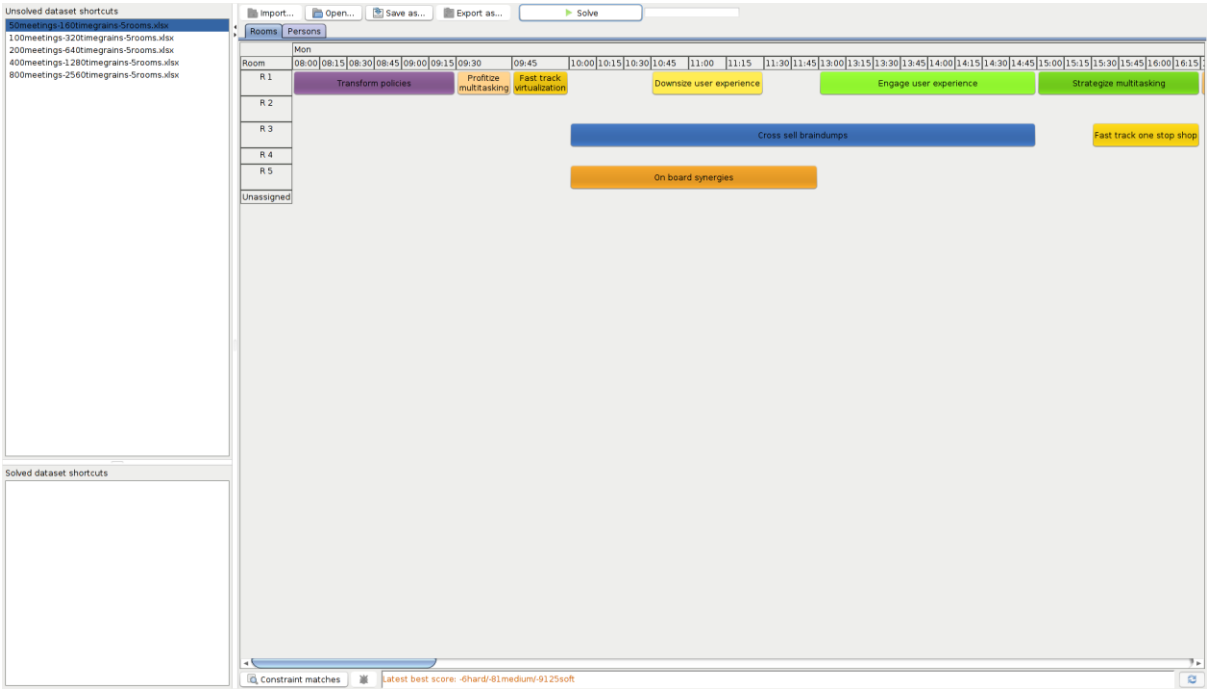
- **Business Scenario / Problem Description**
- **Assign each meeting to a starting time and a room. Meetings have different durations.**
- **Hard constraints:**
  - Room conflict: two meetings must not use the same room at the same time.
  - Required attendance: A person cannot have two required meetings at the same time.
  - Required room capacity: A meeting must not be in a room that doesn't fit all of the meeting's attendees.
  - Start and end on same day: A meeting shouldn't be scheduled over multiple days.
- **Medium constraints:**
  - Preferred attendance: A person cannot have two preferred meetings at the same time, nor a preferred and a required meeting at the same time.
- **Soft constraints:**
  - Sooner rather than later: Schedule all meetings as soon as possible.
  - A break between meetings: Any two meetings should have at least one time grain break between them.
  - Overlapping meetings: To minimize the number of meetings in parallel so people don't have to choose one meeting over the other.
  - Assign larger rooms first: If a larger room is available any meeting should be assigned to that room in order to accommodate as many people as possible even if they haven't signed up to that meeting.
  - Room stability: If a person has two consecutive meetings with two or less time grains break between them they better be in the same room.

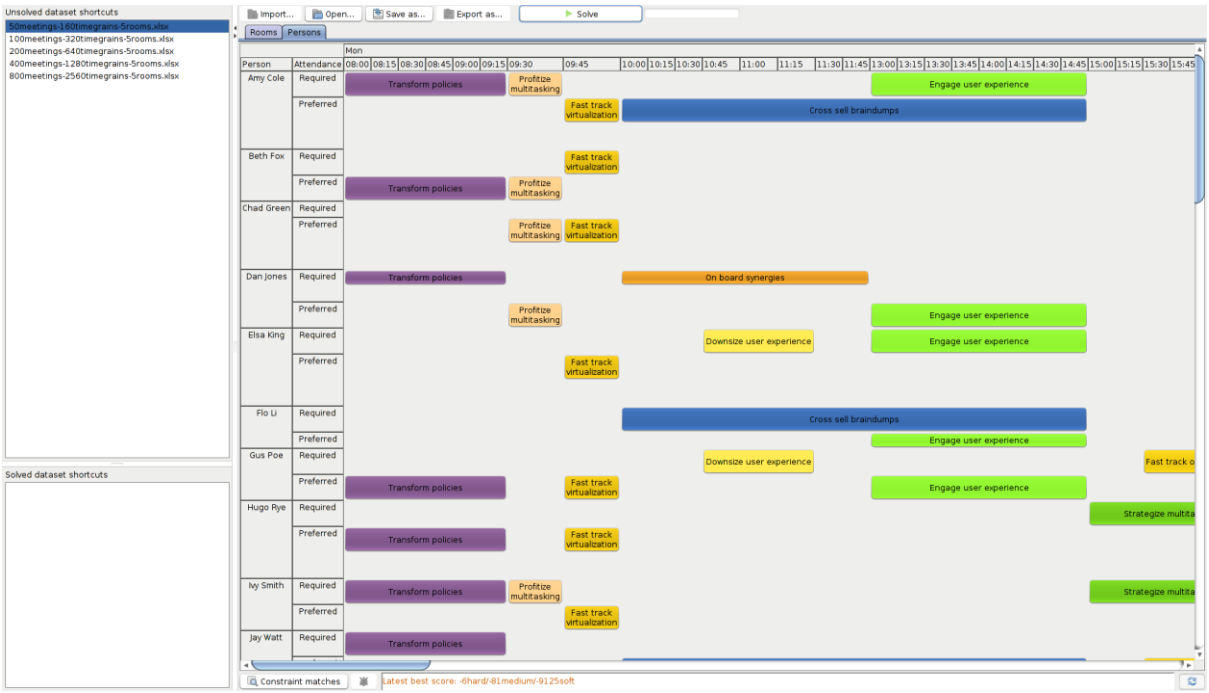
 286

ANNEX 4 : KIE OPTAPLANNER EXAMPLES

Exercise: Draw Class Diagram

289


Unsolved dataset shortcuts
50meetings-160timegrains-5rooms.xlsx
100meetings-320timegrains-5rooms.xlsx
200meetings-640timegrains-5rooms.xlsx
400meetings-1280timegrains-5rooms.xlsx
800meetings-2560timegrains-5rooms.xlsx
Solved dataset shortcuts
Import...
Open...
Save as...
Export as...
Solve
Rooms
Persons
Mon
Room
R 1
R 2
R 3
R 4
R 5
Unassigned
Transform policies
Profitize multitasking
Fast track virtualization
Downsize user experience
Engage user experience
Strategize multitasking
Cross sell braindumps
Fast track one stop shop
On board synergies
Constraint matches
Latest best score: -6hard/-81medium/-9125soft


Unsolved dataset shortcuts
50meetings-160timegrains-5rooms.xlsx
100meetings-320timegrains-5rooms.xlsx
200meetings-640timegrains-5rooms.xlsx
400meetings-1280timegrains-5rooms.xlsx
800meetings-2560timegrains-5rooms.xlsx
Solved dataset shortcuts
Import...
Open...
Save as...
Export as...
Solve
Rooms
Persons
Mon
Person
Attendance
Amy Cole
Beth Fox
Chad Green
Dan Jones
Elsa King
Flo Li
Gus Poe
Hugo Rye
Ivy Smith
Jay Watt
Required
Preferred
Transform policies
Profitize multitasking
Fast track virtualization
Engage user experience
Cross sell braindumps
On board synergies
Downsize user experience
Constraint matches
Latest best score: -6hard/-81medium/-9125soft

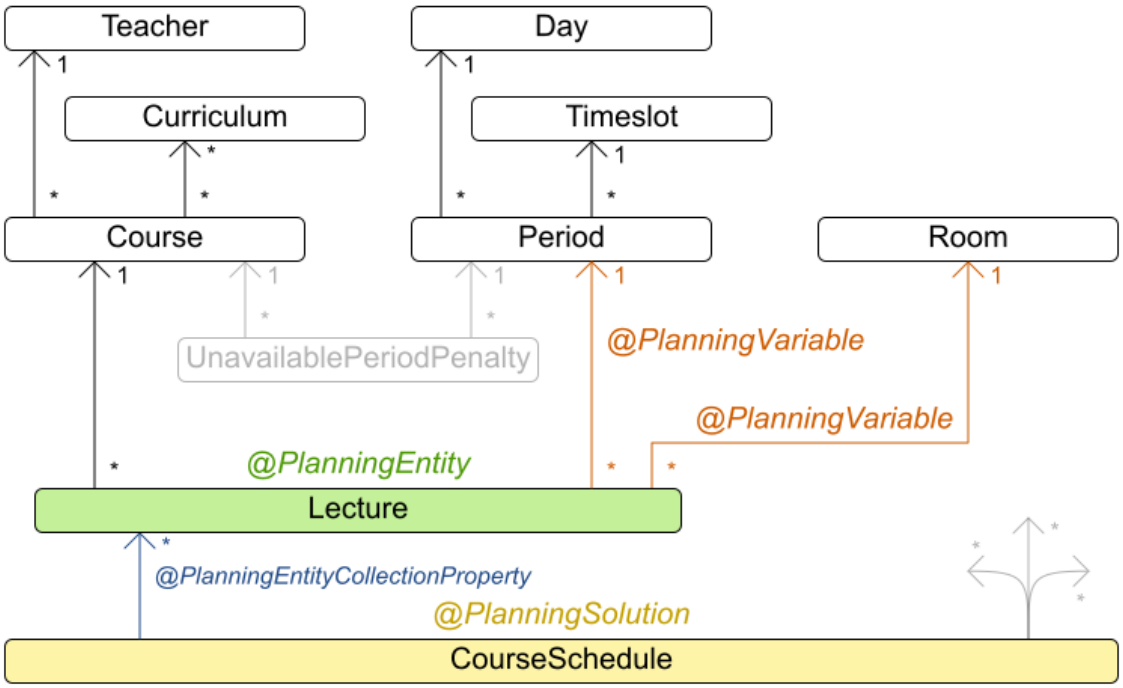
ANNEX 4 : KIE OPTAPLANNER EXAMPLES

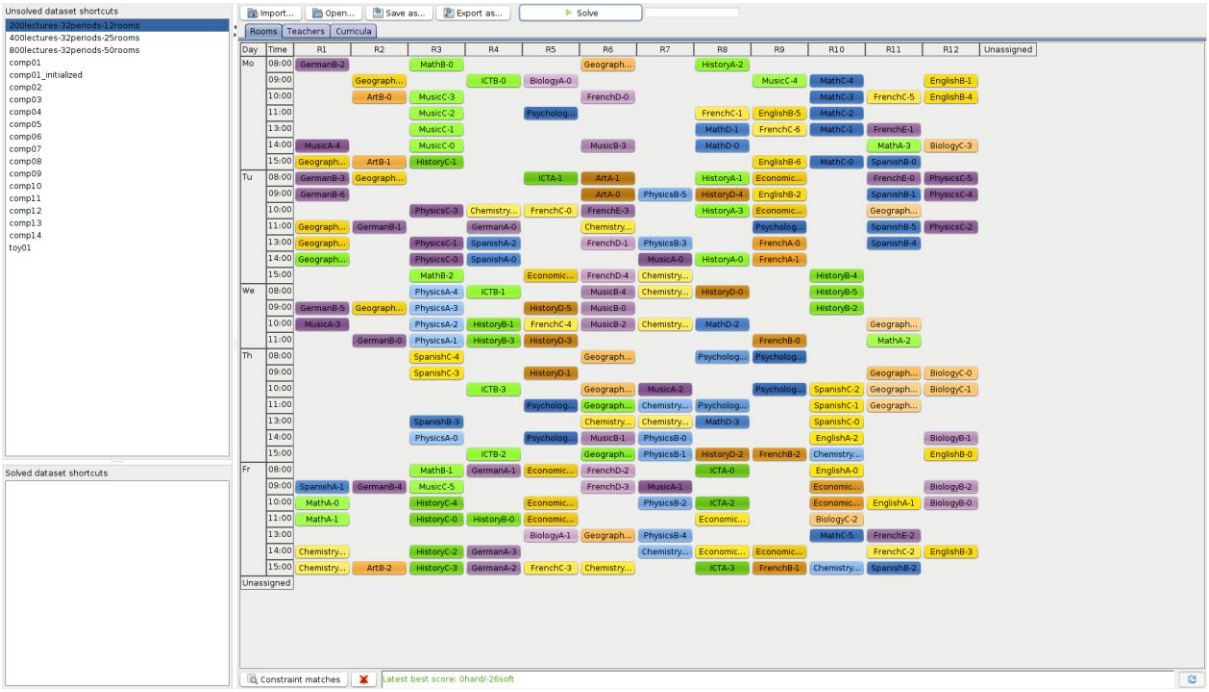
KIE OptaPlanner Deep Dive – Curriculum Course Scheduling

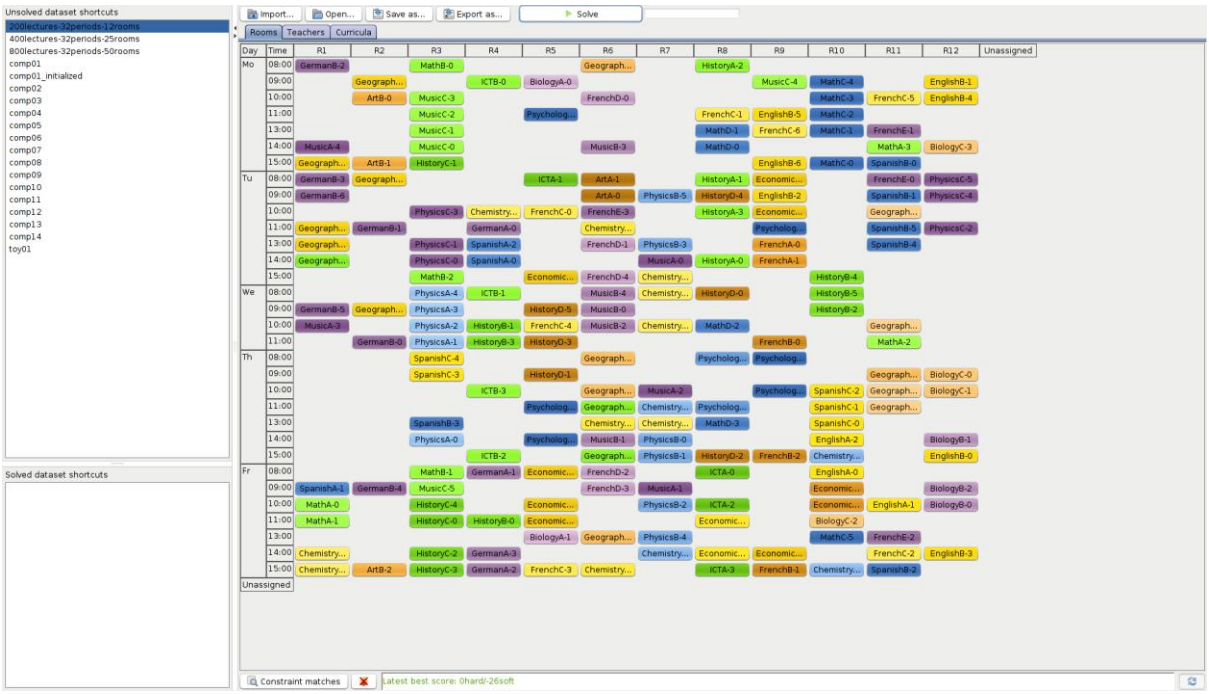
- **Business Scenario / Problem Description**
- **Schedule each lecture into a timeslot and into a room.**
- **Hard constraints:**
  - Teacher conflict: A teacher must not have two lectures in the same period.
  - Curriculum conflict: A curriculum must not have two lectures in the same period.
  - Room occupancy: two lectures must not be in the same room in the same period.
  - Unavailable period (specified per dataset): A specific lecture must not be assigned to a specific period.
- **Soft constraints:**
  - Room capacity: A room's capacity should not be less than the number of students in its lecture.
  - Minimum working days: Lectures of the same course should be spread out into a minimum number of days.
  - Curriculum compactness: Lectures belonging to the same curriculum should be adjacent to each other (so in consecutive periods).
  - Room stability: Lectures of the same course should be assigned to the same room.
- **The problem is defined by the International Timetabling Competition 2007 track 3.**
  http://www.cs.qub.ac.uk/itc2007/curriculmcourse/course_curriculm_index.htm

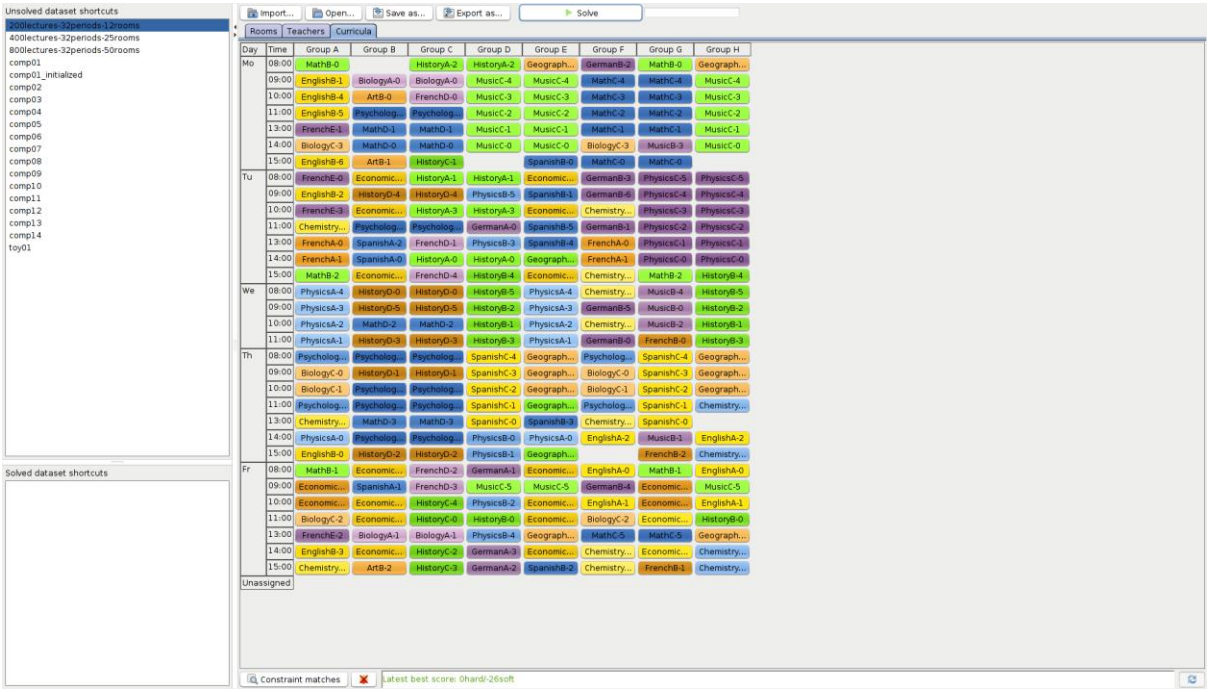
 294

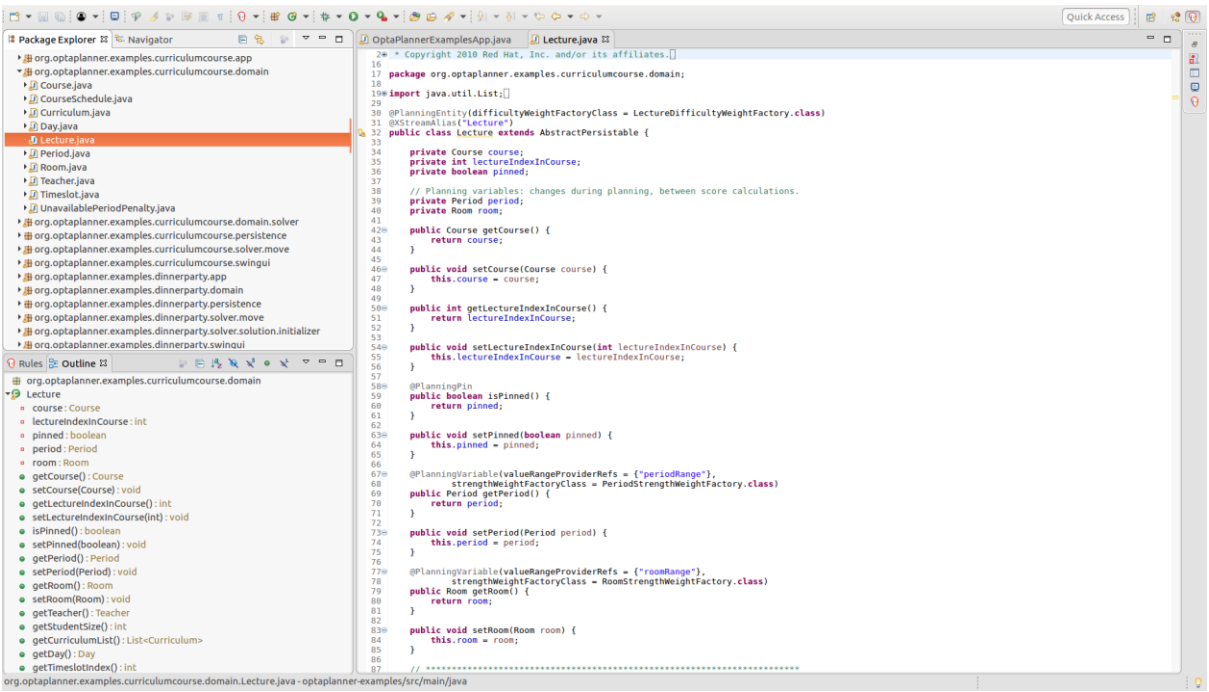
Curriculum course class diagram




Unsolved dataset shortcuts
200lectures-32periods-12rooms
400lectures-32periods-25rooms
800lectures-32periods-50rooms
comp01
comp01_initialized
comp02
comp03
comp04
comp05
comp06
comp07
comp08
comp09
comp10
comp11
comp12
comp13
comp14
toy01
Solved dataset shortcuts
Import...
Open...
Save as...
Export as...
Solve
Rooms
Teachers
Curricula
Day
Time
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
Unassigned
Mo
Tu
We
Th
Fr
Unassigned
Constraint matches
Latest best score: 0hard/-26soft


Unsolved dataset shortcuts
200lectures-32periods-12rooms
400lectures-32periods-25rooms
800lectures-32periods-50rooms
comp01
comp01_initialized
comp02
comp03
comp04
comp05
comp06
comp07
comp08
comp09
comp10
comp11
comp12
comp13
comp14
toy01
Solved dataset shortcuts
Import...
Open...
Save as...
Export as...
Solve
Rooms
Teachers
Curricula
Day
Time
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
Unassigned
Mo
Tu
We
Th
Fr
Unassigned
Constraint matches
Latest best score: 0hard/-26soft


Unsolved dataset shortcuts
200lectures-32periods-12rooms
400lectures-32periods-25rooms
800lectures-32periods-50rooms
comp01
comp01_initialized
comp02
comp03
comp04
comp05
comp06
comp07
comp08
comp09
comp10
comp11
comp12
comp13
comp14
toy01
Solved dataset shortcuts
Import...
Open...
Save as...
Export as...
Solve
Rooms
Teachers
Curricula
Constraint matches
Latest best score: 0hard/-26soft


Package Explorer
Navigator
OptaPlannerExamplesApp.java
Lecture.java
Rules
Outline
org.optaplanner.examples.curriculumcourse.domain.Lecture.java - optaplanner-examples/src/main/java

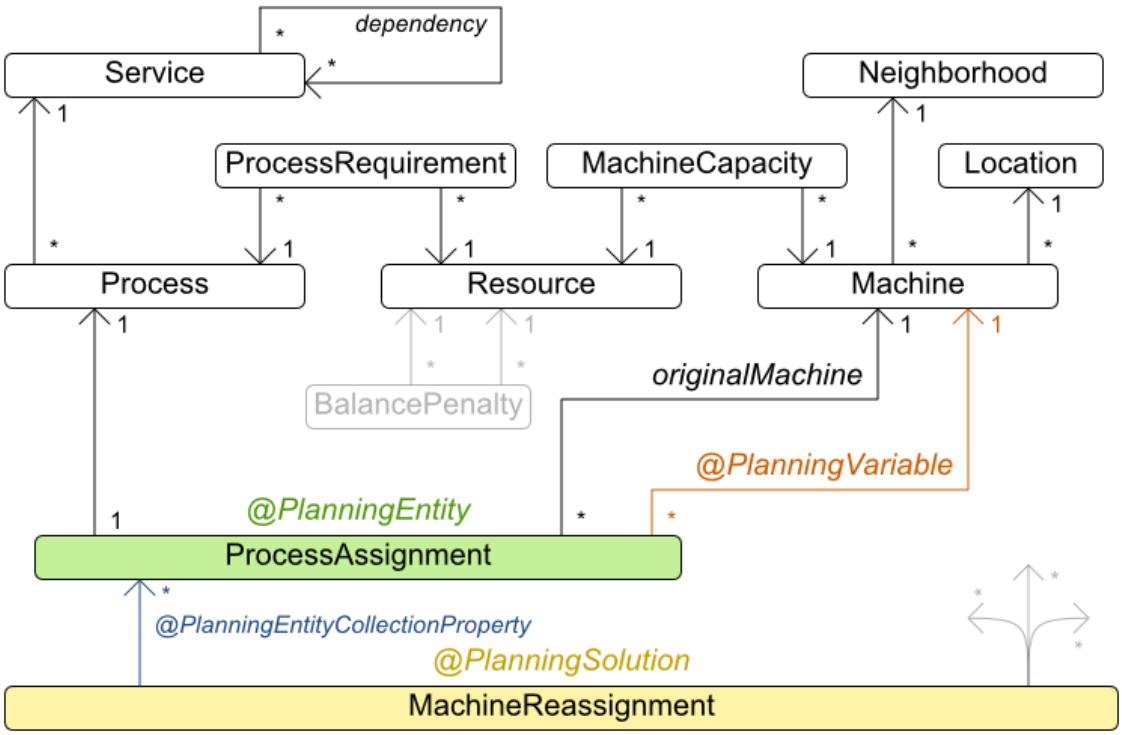
ANNEX 4 : KIE OPTAPLANNER EXAMPLES

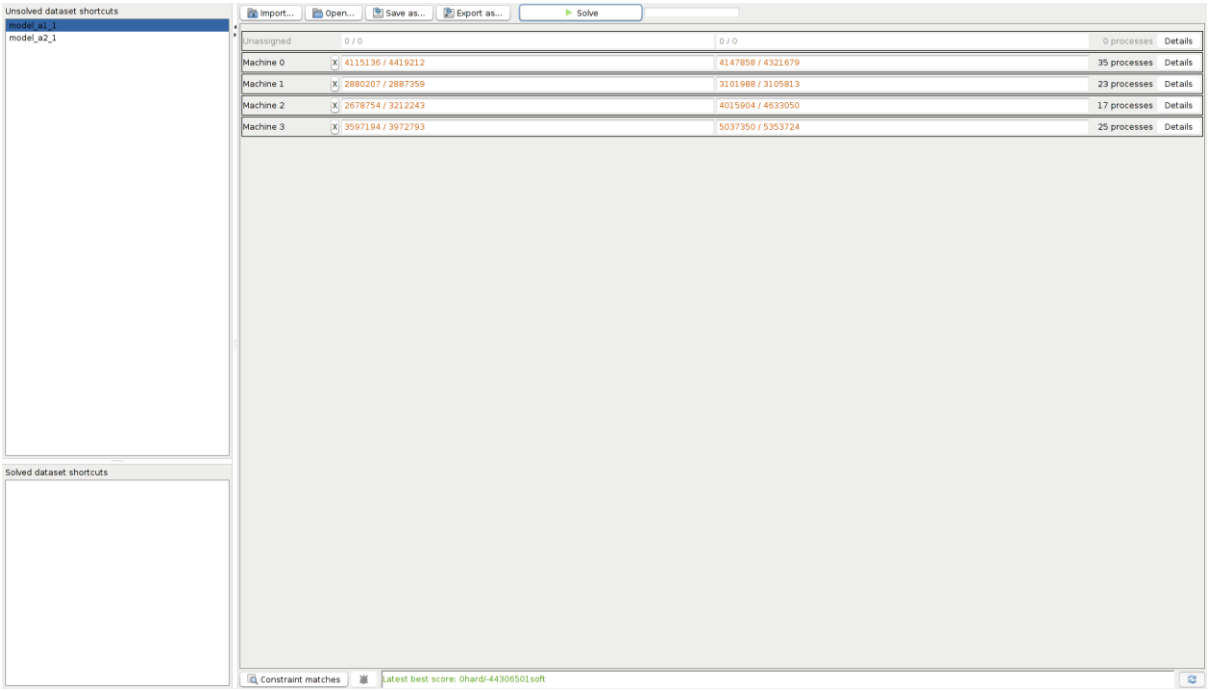
KIE OptaPlanner Deep Dive – Machine Reassignment

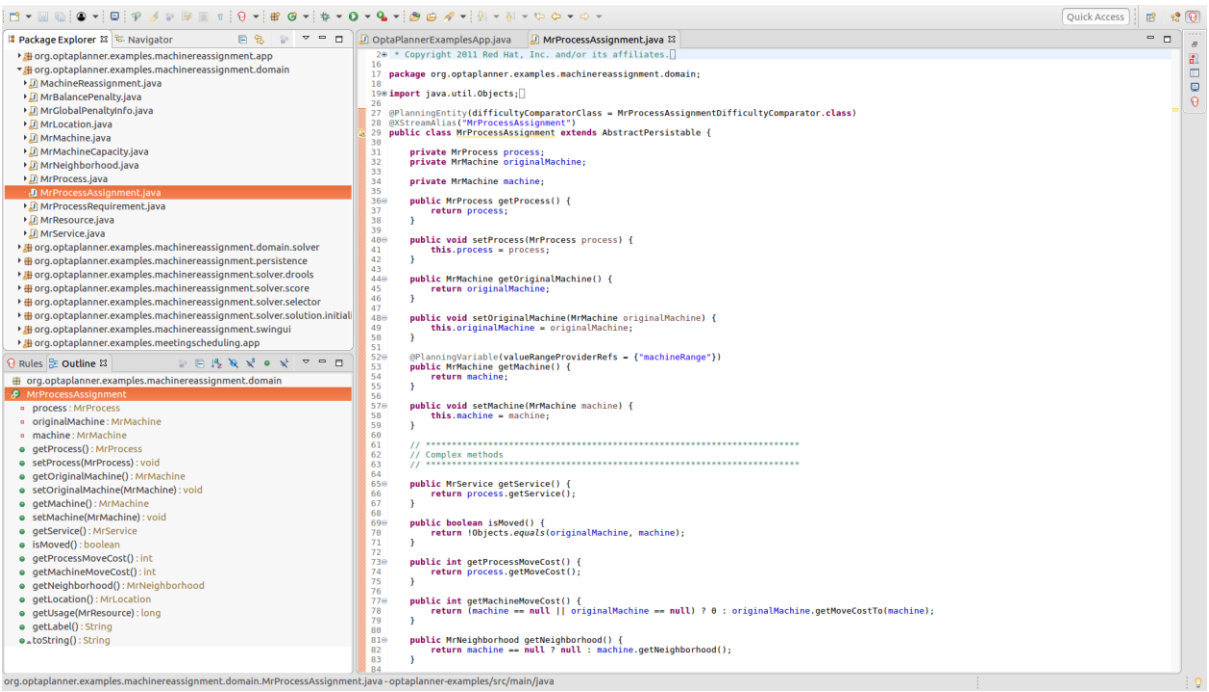
- **Business Scenario / Problem Description**
- **Assign each process to a machine. All processes already have an original (unoptimized) assignment. Each process requires an amount of each resource (such as CPU, RAM, ...). This is a more complex version of the Cloud Balancing example.**
- **Hard constraints:**
  - Maximum capacity: The maximum capacity for each resource for each machine must not be exceeded.
  - Conflict: Processes of the same service must run on distinct machines.
  - Spread: Processes of the same service must be spread out across locations.
  - Dependency: The processes of a service depending on another service must run in the neighborhood of a process of the other service.
  - Transient usage: Some resources are transient and count towards the maximum capacity of both the original machine as the newly assigned machine.
- **Soft constraints:**
  - Load: The safety capacity for each resource for each machine should not be exceeded.
  - Balance: Leave room for future assignments by balancing the available resources on each machine.
  - Process move cost: A process has a move cost.
  - Service move cost: A service has a move cost.
  - Machine move cost: Moving a process from machine A to machine B has another A-B specific move cost.
- **The problem is defined by the Google ROADEF/EURO Challenge 2012.**
  http://challenge.roadef.org/2012/en/

 303

Machine reassignment class diagram




Unsolved dataset shortcuts
model_a1_1
model_a2_1
Solved dataset shortcuts
Import...
Open...
Save as...
Export as...
Solve
Unassigned
Machine 0
Machine 1
Machine 2
Machine 3
Details
Constraint matches
Latest best score: 0hard/-44306501soft


Package Explorer
Navigator
OptaPlannerExamplesApp.java
MrProcessAssignment.java
Rules
Outline
org.optaplanner.examples.machinereassignment.domain.MrProcessAssignment.java - optaplanner-examples/src/main/java

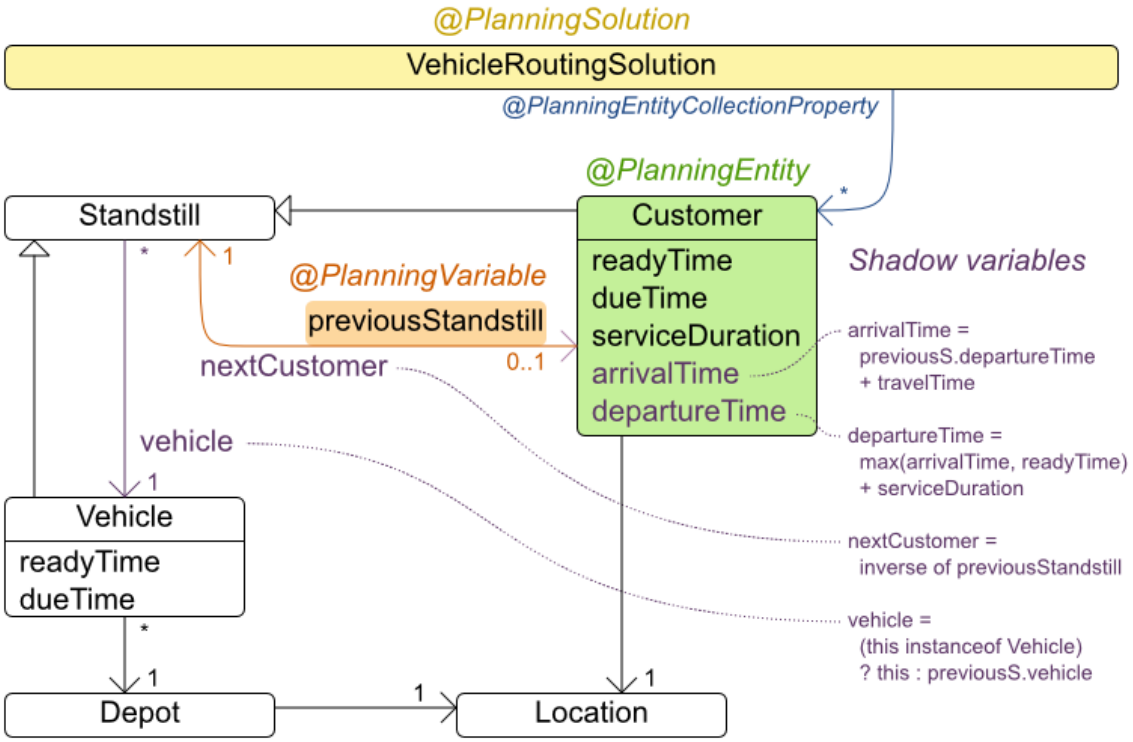
ANNEX 4 : KIE OPTAPLANNER EXAMPLES

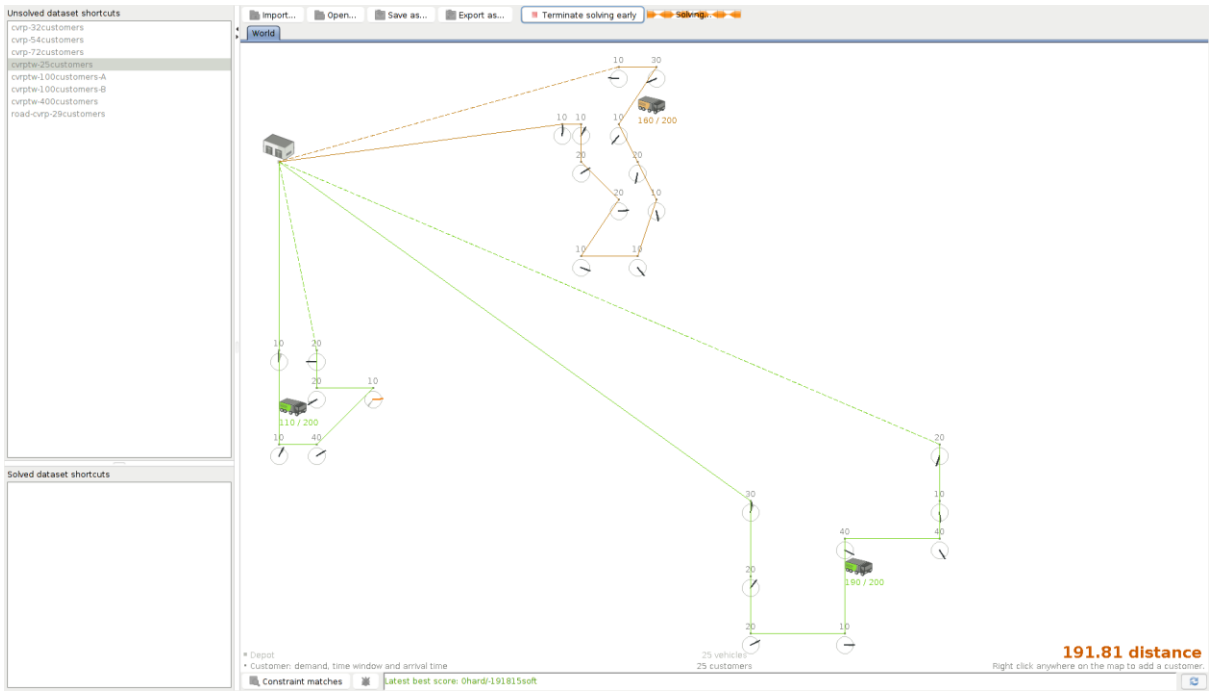
KIE OptaPlanner Deep Dive – Vehicle Routing

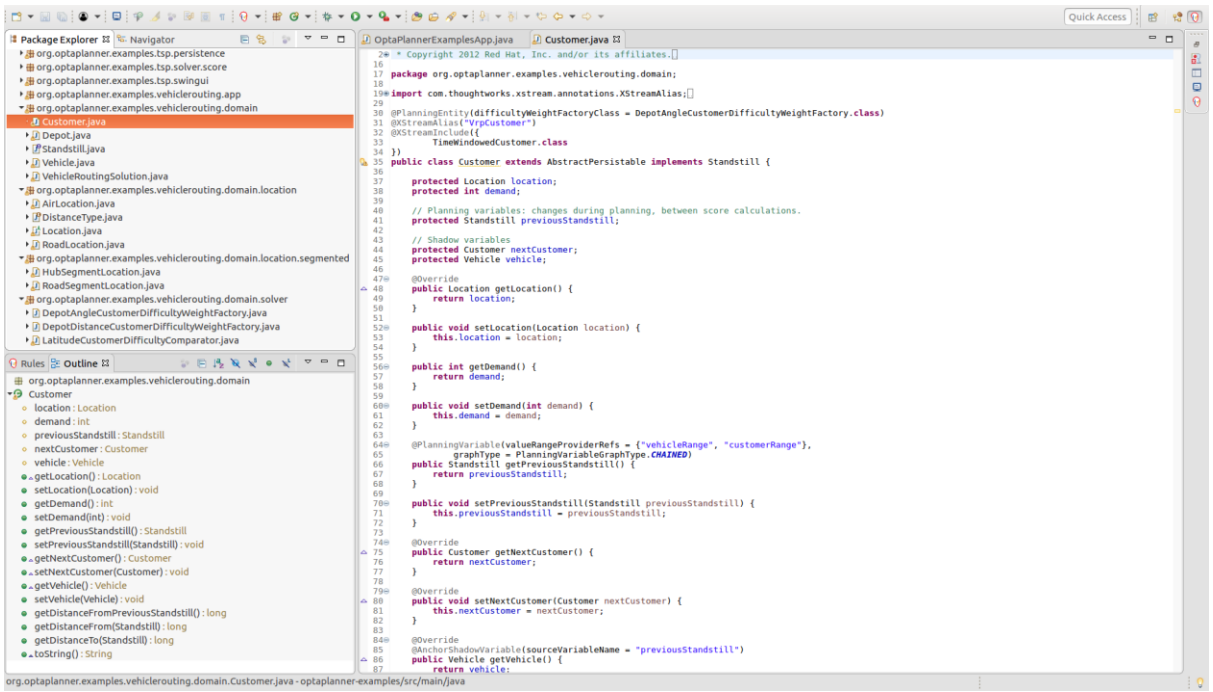
- **Business Scenario / Problem Description**
- **Using a fleet of vehicles, pick up the objects of each customer and bring them to the depot. Each vehicle can service multiple customers, but it has a limited capacity.**
- **Besides the basic case (CVRP), there is also a variant with time windows (CVRPTW).**
- **Hard constraints:**
  - Vehicle capacity: a vehicle cannot carry more items then its capacity.
  - Time windows (only in CVRPTW):
  - Travel time: Traveling from one location to another takes time.
  - Customer service duration: a vehicle must stay at the customer for the length of the service duration.
  - Customer ready time: a vehicle may arrive before the customer's ready time, but it must wait until the ready time before servicing.
  - Customer due time: a vehicle must arrive on time, before the customer's due time.
- **Soft constraints:**
  - Total distance: minimize the total distance driven (fuel consumption) of all vehicles.
- **The capacitated vehicle routing problem (CVRP) and its time-windowed variant (CVRPTW) are defined by the VRP web.** http://neo.lcc.uma.es/vrp/

 311

Vehicle routing class diagram




Unsolved dataset shortcuts
cvrp-32customers
cvrp-54customers
cvrp-72customers
cvrptw-25customers
cvrptw-100customers-A
cvrptw-100customers-B
cvrptw-400customers
road-cvrp-29customers
Solved dataset shortcuts
Import...
Open...
Save as...
Export as...
Terminate solving early
Solving...
World
160 / 200
110 / 200
190 / 200
191.81 distance
Depot
Customer: demand, time window and arrival time
25 vehicles
25 customers
Right click anywhere on the map to add a customer.
Constraint matches
Latest best score: 0hard/-191815soft


Quick Access
Package Explorer
Navigator
org.optaplanner.examples.tsp.persistence
org.optaplanner.examples.tsp.solver.score
org.optaplanner.examples.tsp.swingui
org.optaplanner.examples.vehiclerouting.app
org.optaplanner.examples.vehiclerouting.domain
Customer.java
Depot.java
Standstill.java
Vehicle.java
VehicleRoutingSolution.java
org.optaplanner.examples.vehiclerouting.domain.location
AirLocation.java
DistanceType.java
Location.java
RoadLocation.java
org.optaplanner.examples.vehiclerouting.domain.location.segmented
HubSegmentLocation.java
RoadSegmentLocation.java
org.optaplanner.examples.vehiclerouting.domain.solver
DepotAngleCustomerDifficultyWeightFactory.java
DepotDistanceCustomerDifficultyWeightFactory.java
LatitudeCustomerDifficultyComparator.java
Rules
Outline
org.optaplanner.examples.vehiclerouting.domain
Customer
location : Location
demand : int
previousStandstill : Standstill
nextCustomer : Customer
vehicle : Vehicle
getLocation() : Location
setLocation(Location) : void
getDemand() : int
setDemand(int) : void
getPreviousStandstill() : Standstill
setPreviousStandstill(Standstill) : void
getNextCustomer() : Customer
setNextCustomer(Customer) : void
getVehicle() : Vehicle
setVehicle(Vehicle) : void
getDistanceFromPreviousStandstill() : long
getDistanceFrom(Standstill) : long
getDistanceTo(Standstill) : long
toString() : String
OptaPlannerExamplesApp.java
Customer.java
org.optaplanner.examples.vehiclerouting.domain.Customer.java - optaplanner-examples/src/main/java

ANNEX 4 : KIE OPTAPLANNER EXAMPLES

KIE OptaPlanner Deep Dive – Project Job Scheduling

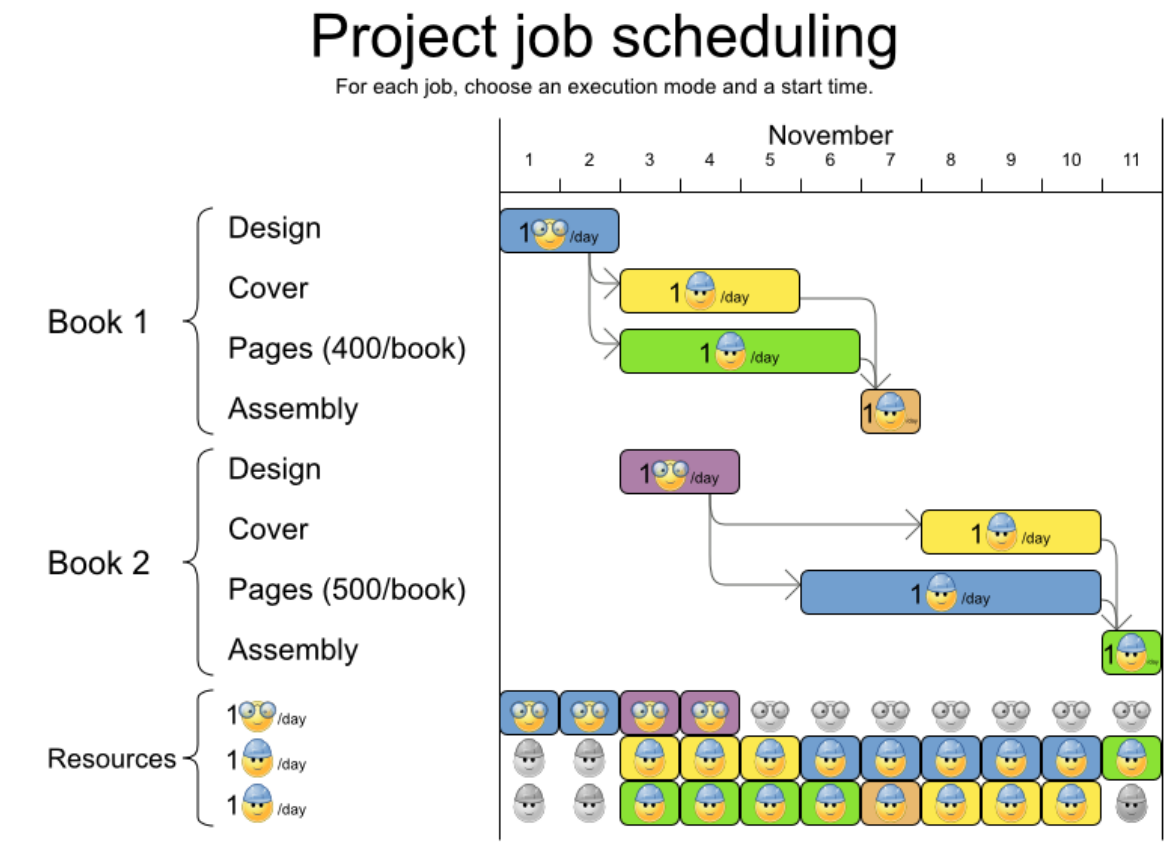
- **Business Scenario / Problem Description**
- **Schedule all jobs in time and execution mode to minimize project delays. Each job is part of a project. A job can be executed in different ways: each way is an execution mode that implies a different duration but also different resource usages. This is a form of flexible job shop scheduling.**
- **Hard constraints:**
  - Job precedence: a job can only start when all its predecessor jobs are finished.
  - Resource capacity: do not use more resources than available.
  - Resources are local (shared between jobs of the same project) or global (shared between all jobs)
  - Resource are renewable (capacity available per day) or nonrenewable (capacity available for all days)
- **Medium constraints:**
  - Total project delay: minimize the duration (makespan) of each project.
- **Soft constraints:**
  - Total makespan: minimize the duration of the whole multi-project schedule.
  - The problem is defined by the MISTA 2013 challenge.
- **The problem is defined by the MISTA 2013 challenge.** http://gent.cs.kuleuven.be/mista2013challenge/

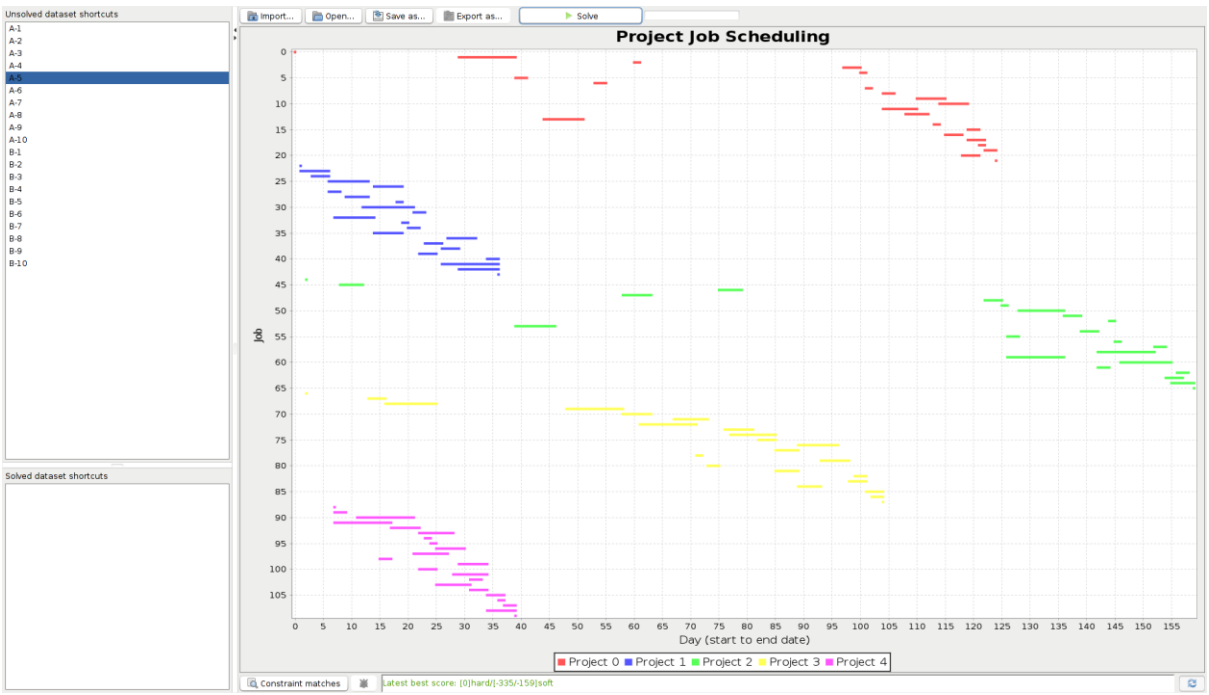
 320

ANNEX 4 : KIE OPTAPLANNER EXAMPLES

Exercise: Draw Class Diagram

323


Project job scheduling
For each job, choose an execution mode and a start time.
November
1 2 3 4 5 6 7 8 9 10 11
Book 1
Design
Cover
Pages (400/book)
Assembly
Book 2
Design
Cover
Pages (500/book)
Assembly
Resources
1 /day
1 /day
1 /day


Unsolved dataset shortcuts
Solved dataset shortcuts
Import...
Open...
Save as...
Export as...
Solve
Project Job Scheduling
Job
Day (start to end date)
Project 0
Project 1
Project 2
Project 3
Project 4
Constraint matches

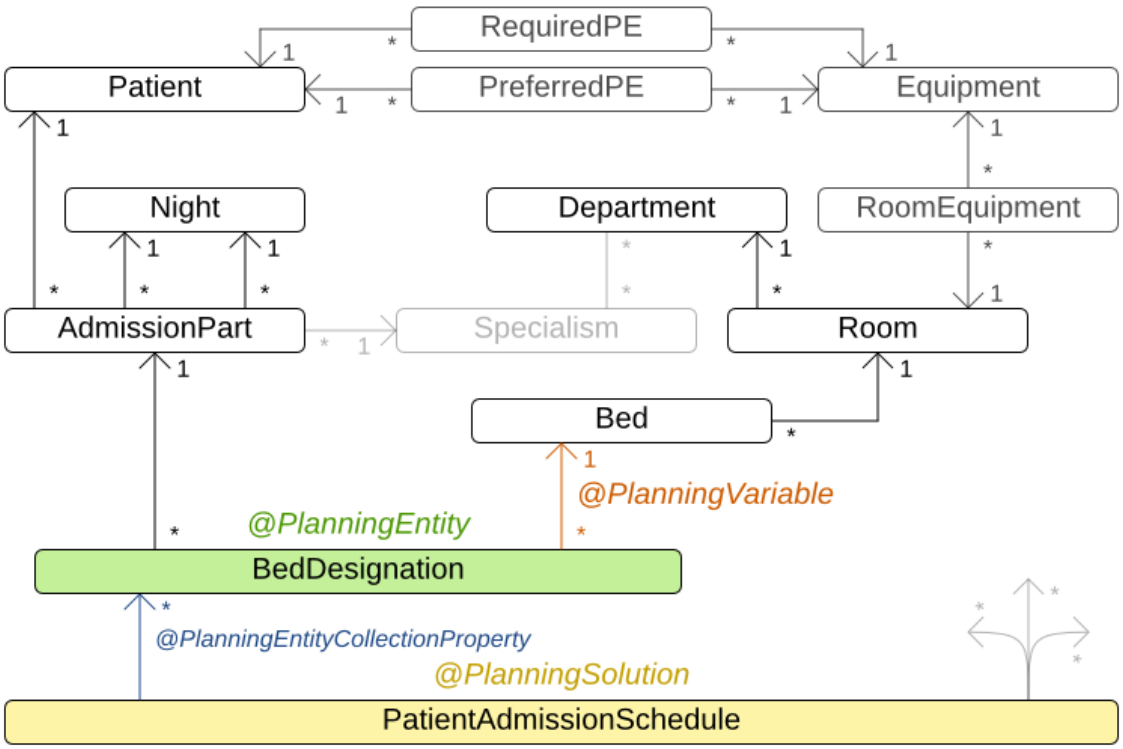
ANNEX 4 : KIE OPTAPLANNER EXAMPLES

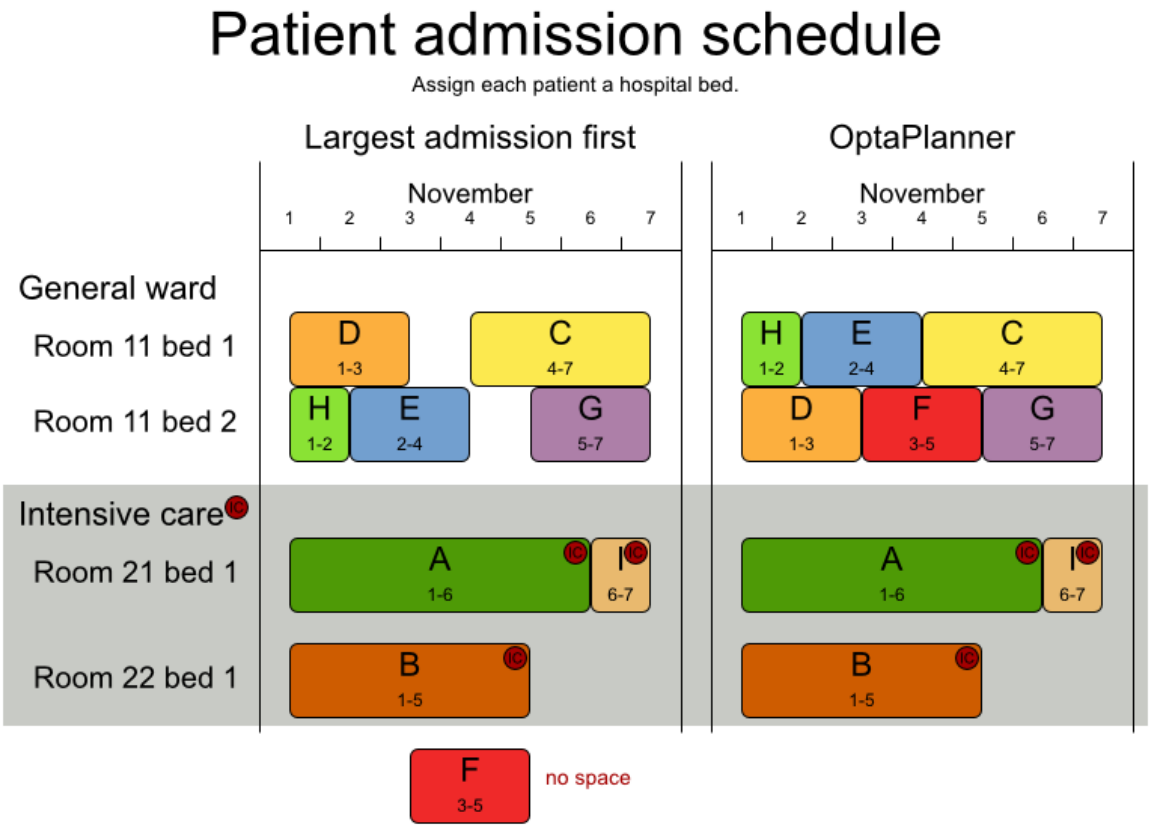
KIE OptaPlanner Deep Dive – Hospital Bed Planning (PAS)

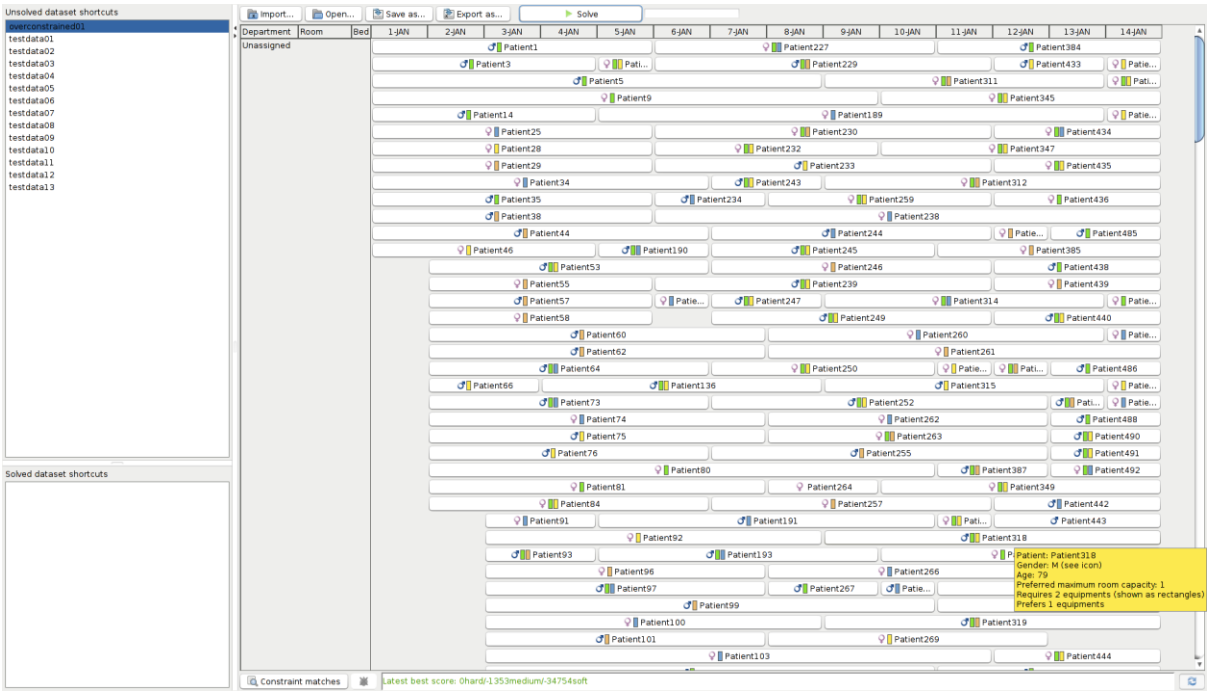
- **Business Scenario / Problem Description**
- Assign each patient (that will come to the hospital) into a bed for each night that the patient will stay in the hospital. Each bed belongs to a room and each room belongs to a department. The arrival and departure dates of the patients is fixed: only a bed needs to be assigned for each night. This problem features overconstrained datasets.
- **Hard constraints:**
  - Two patients must not be assigned to the same bed in the same night. Weight: -1000hard * conflictNightCount.
  - A room can have a gender limitation: only females, only males, the same gender in the same night or no gender limitation at all. Weight: -50hard * nightCount.
  - A department can have a minimum or maximum age. Weight: -100hard * nightCount.
  - A patient can require a room with specific equipment(s). Weight: -50hard * nightCount.
- **Medium constraints:**
  - Assign every patient to a bed, unless the dataset is over-constrained. Weight: -1medium * nightCount.
- **Soft constraints:**
  - A patient can prefer a maximum room size, for example if he/she wants a single room. Weight: -8soft * nightCount.
  - A patient is best assigned to a department that specializes in his/her problem. Weight: -10soft * nightCount.
  - A patient is best assigned to a room that specializes in his/her problem. Weight: -20soft * nightCount.
  - That room speciality should be priority 1. Weight: -10soft * (priority - 1) * nightCount.
  - A patient can prefer a room with specific equipment(s). Weight: -20soft * nightCount.
- The problem is a variant on **Kaho's Patient Scheduling**. https://people.cs.kuleuven.be/~wim.vancroonenburg/pas/

328

Hospital bed allocation class diagram




Patient admission schedule
Assign each patient a hospital bed.
Largest admission first
OptaPlanner
November
1 2 3 4 5 6 7
General ward
Room 11 bed 1
Room 11 bed 2
Intensive care
Room 21 bed 1
Room 22 bed 1
D 1-3
C 4-7
H 1-2
E 2-4
G 5-7
A 1-6
I 6-7
B 1-5
F 3-5
no space



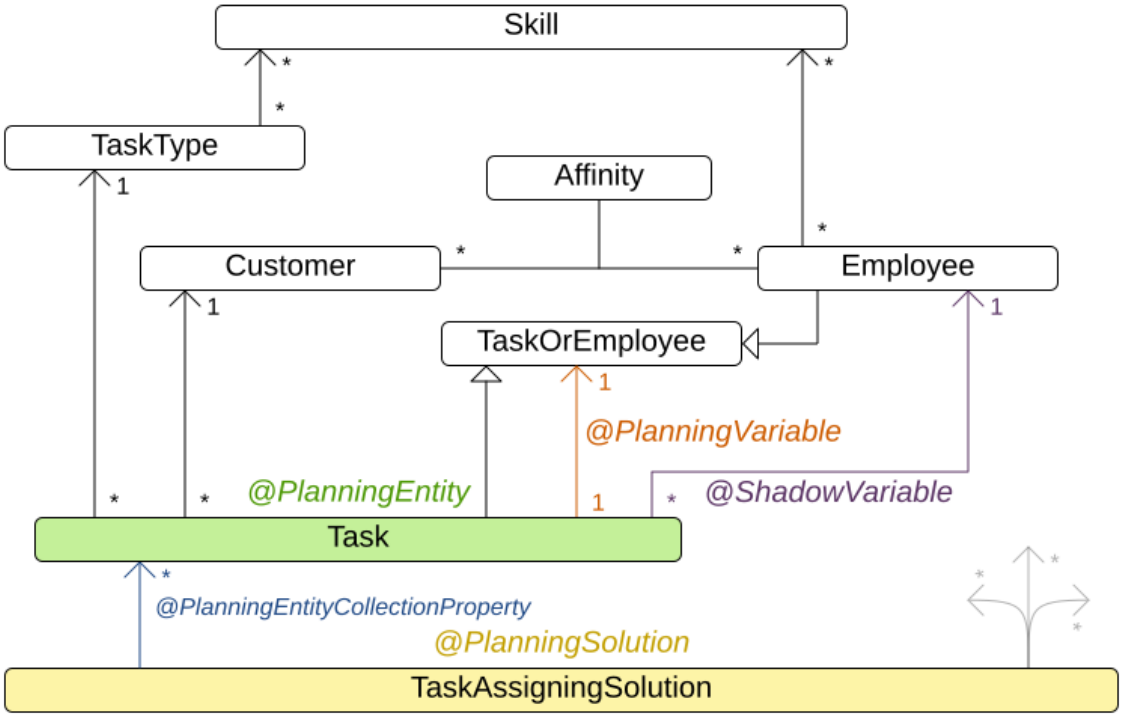
ANNEX 4 : KIE OPTAPLANNER EXAMPLES

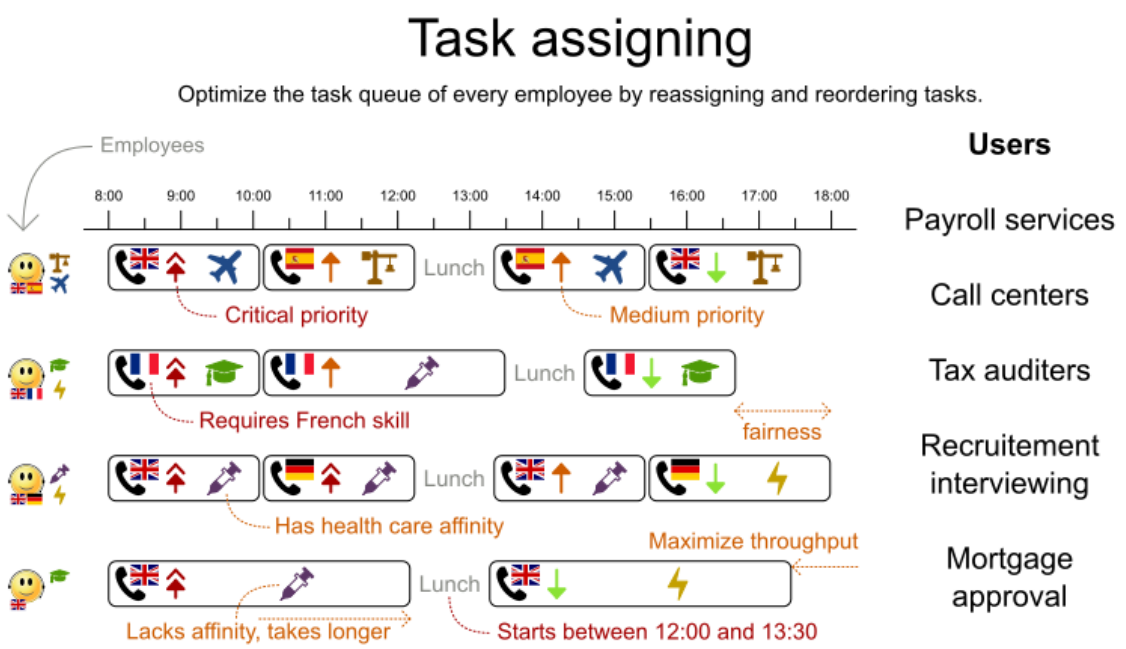
KIE OptaPlanner Deep Dive – Task assigning

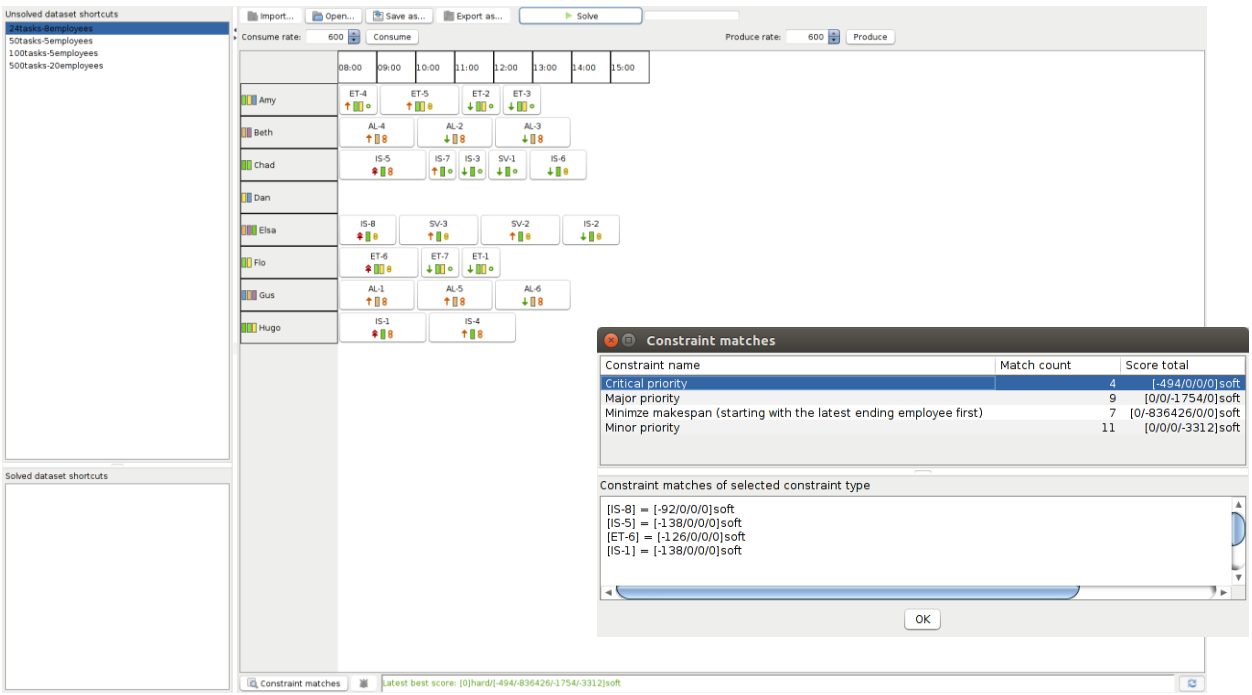
- **Business Scenario / Problem Description**
- **Assign each task to a spot in an employee's queue. Each task has a duration which is affected by the employee's affinity level with the task's customer.**
- **Hard constraints:**
  - Skill: Each task requires one or more skills. The employee must posses all these skills.
- **Soft level 0 constraints:**
  - Critical tasks: Complete critical tasks first, sooner than major and minor tasks.
- **Soft level 1 constraints:**
  - Minimize makespan: Reduce the time to complete all tasks.
  - Start with the longest working employee first, then the second longest working employee and so forth, to creates fairness and load balancing.
- **Soft level 2 constraints:**
  - Major tasks: Complete major tasks as soon as possible, sooner than minor tasks.
- **Soft level 3 constraints:**
  - Minor tasks: Complete minor tasks as soon as possible.

337

Task assigning class diagram




Task assigning
Optimize the task queue of every employee by reassigning and reordering tasks.
Employees
8:00
9:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
Lunch
Critical priority
Medium priority
Requires French skill
fairness
Has health care affinity
Maximize throughput
Lacks affinity, takes longer
Starts between 12:00 and 13:30
Users
Payroll services
Call centers
Tax auditers
Recruitement interviewing
Mortgage approval

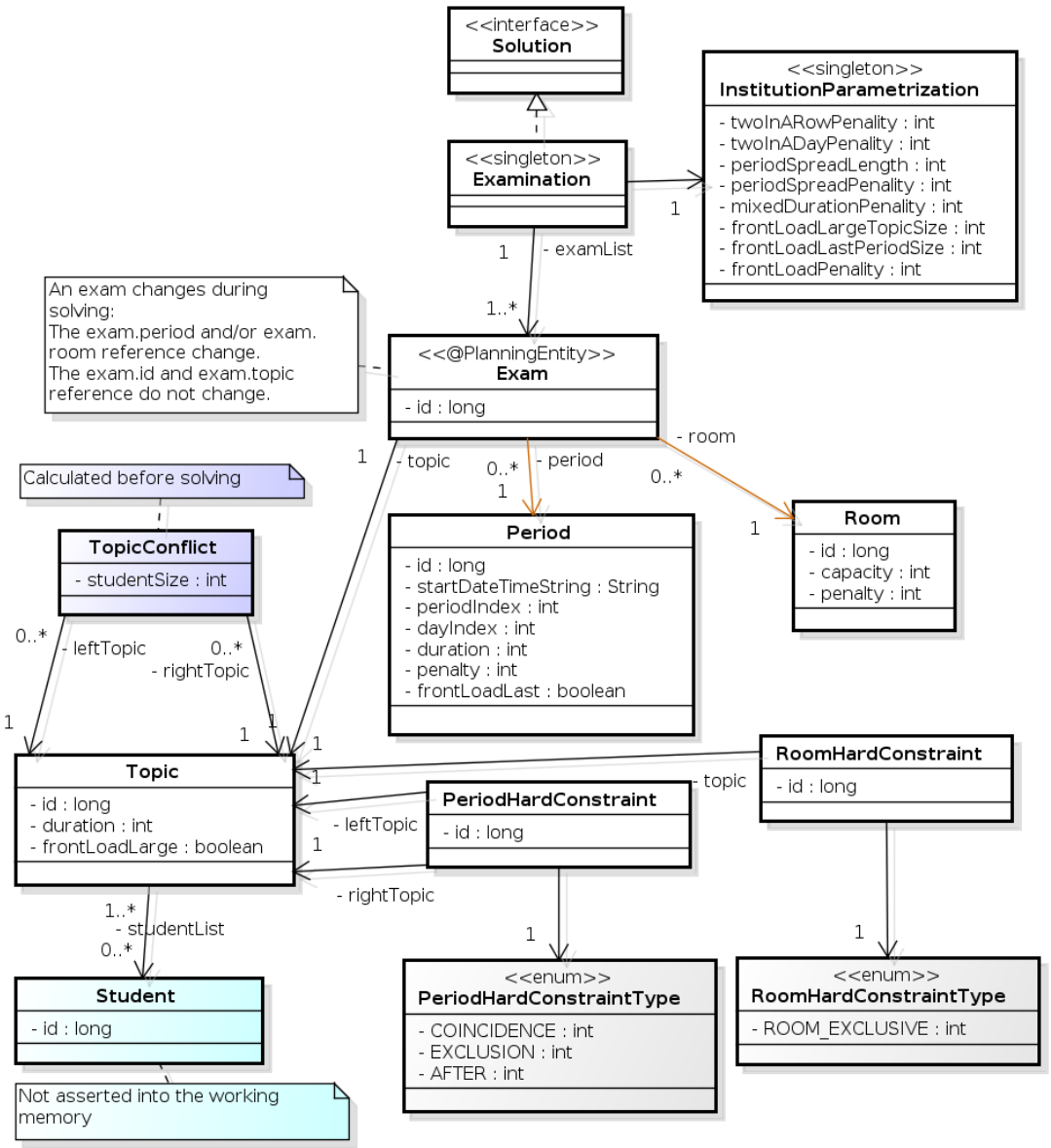

Unsolved dataset shortcuts
24tasks-8employees
50tasks-5employees
100tasks-5employees
500tasks-20employees
Solved dataset shortcuts
Import...
Open...
Save as...
Export as...
Solve
Consume rate:
Consume
Produce rate:
Produce
Amy
Beth
Chad
Dan
Elsa
Flo
Gus
Hugo
Constraint matches
Constraint name
Match count
Score total
Critical priority 4 [-494/0/0/0]soft
Major priority 9 [0/0/-1754/0]soft
Minimze makespan (starting with the latest ending employee first) 7 [0/-836426/0/0]soft
Minor priority 11 [0/0/0/-3312]soft
Constraint matches of selected constraint type
[IS-8] = [-92/0/0/0]soft
[IS-5] = [-138/0/0/0]soft
[ET-6] = [-126/0/0/0]soft
[IS-1] = [-138/0/0/0]soft
OK
Constraint matches

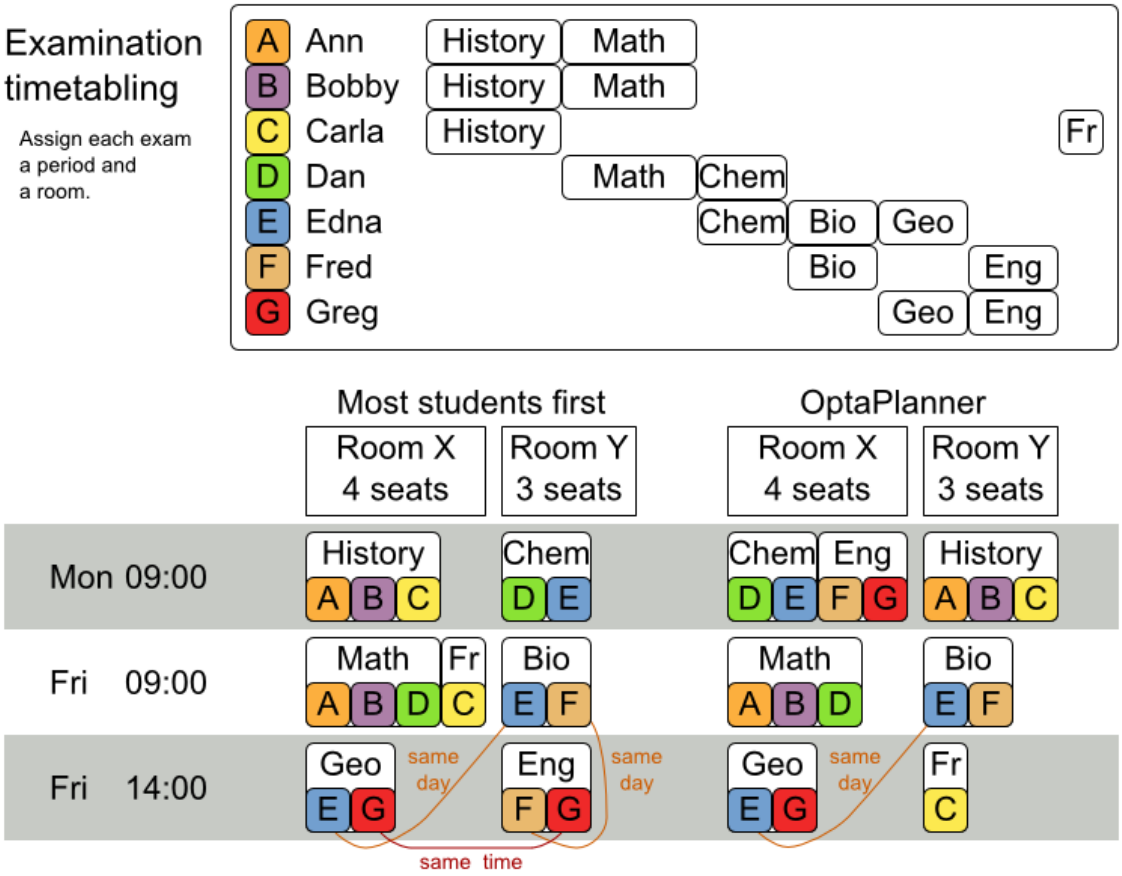
ANNEX 4 : KIE OPTAPLANNER EXAMPLES

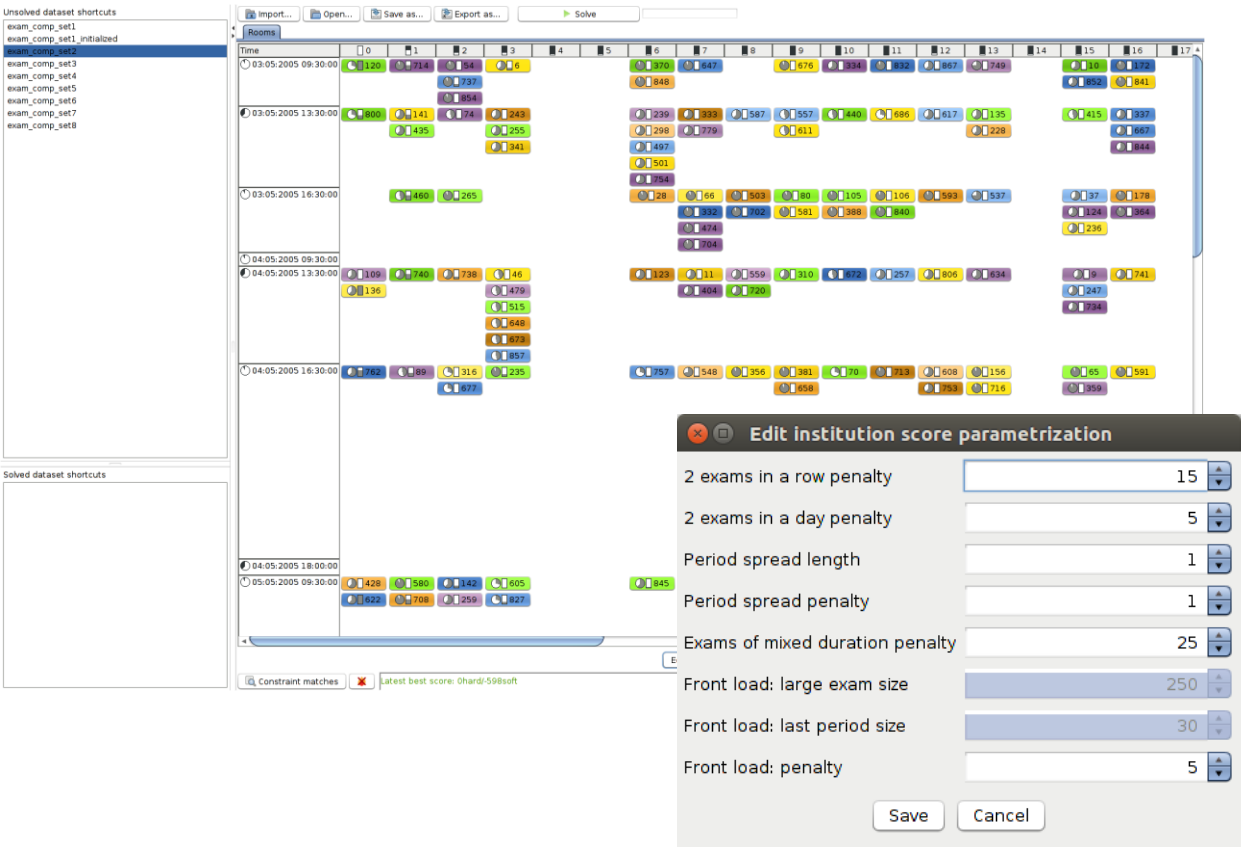
KIE OptaPlanner Deep Dive – Exam Timetabling

- **Business Scenario / Problem Description**
- **Schedule each exam into a period and into a room. Multiple exams can share the same room during the same period.**
- **Hard constraints:**
  - Exam conflict: two exams that share students must not occur in the same period.
  - Room capacity: A room's seating capacity must suffice at all times.
  - Period duration: A period's duration must suffice for all of its exams.
  - Period related hard constraints (specified per dataset):
    - Coincidence: two specified exams must use the same period (but possibly another room).
    - Exclusion: two specified exams must not use the same period.
    - After: A specified exam must occur in a period after another specified exam's period.
  - Room related hard constraints (specified per dataset):
    - Exclusive: one specified exam should not have to share its room with any other exam.
- **Soft constraints (each of which has a parametrized penalty):**
  - The same student should not have two exams in a row.
  - The same student should not have two exams on the same day.
  - Period spread: two exams that share students should be a number of periods apart.
  - Mixed durations: two exams that share a room should not have different durations.
  - Front load: Large exams should be scheduled earlier in the schedule.
  - Period penalty (specified per dataset): Some periods have a penalty when used.
  - Room penalty (specified per dataset): Some rooms have a penalty when used.
  - It uses large test data sets of real-life universities.
- **The problem is defined by the International Timetabling Competition 2007 track 1.**
  http://www.cs.qub.ac.uk/itc2007/examtrack/exam_track_index.htm

346




Examination timetabling
Assign each exam a period and a room.
A Ann History Math
B Bobby History Math
C Carla History Fr
D Dan Math Chem
E Edna Chem Bio Geo
F Fred Bio Eng
G Greg Geo Eng
Most students first
OptaPlanner
Room X 4 seats
Room Y 3 seats
Mon 09:00
Fri 09:00
Fri 14:00
same day
same time


Edit institution score parametrization
2 exams in a row penalty 15
2 exams in a day penalty 5
Period spread length 1
Period spread penalty 1
Exams of mixed duration penalty 25
Front load: large exam size 250
Front load: last period size 30
Front load: penalty 5
Save
Cancel

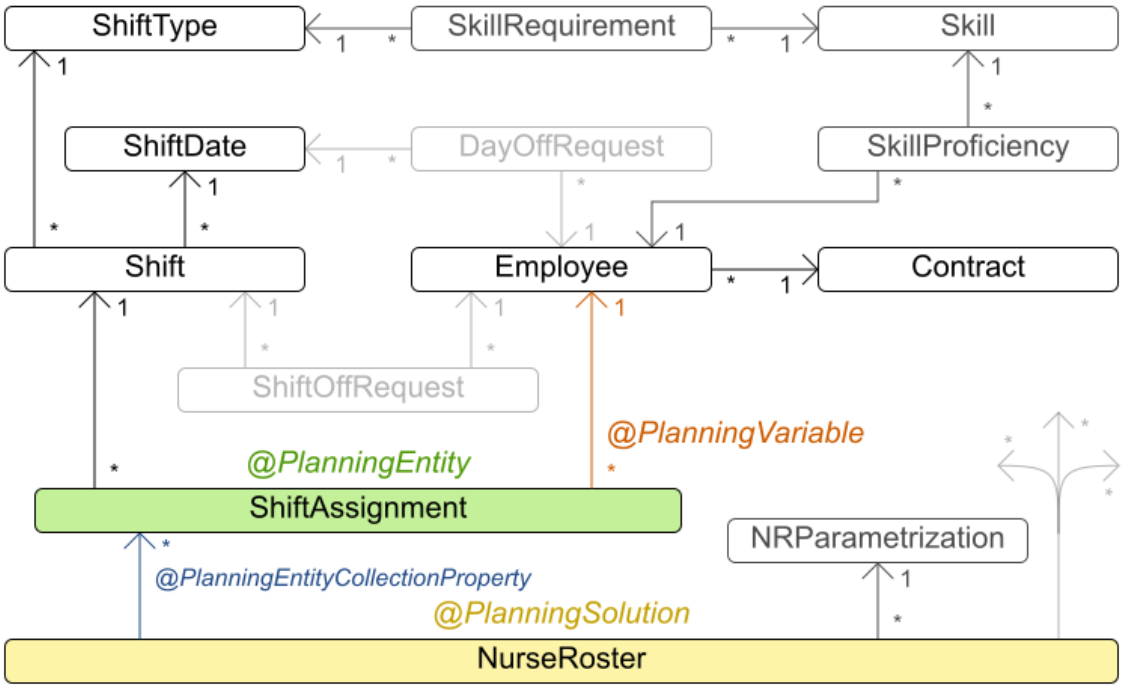
ANNEX 4 : KIE OPTAPLANNER EXAMPLES

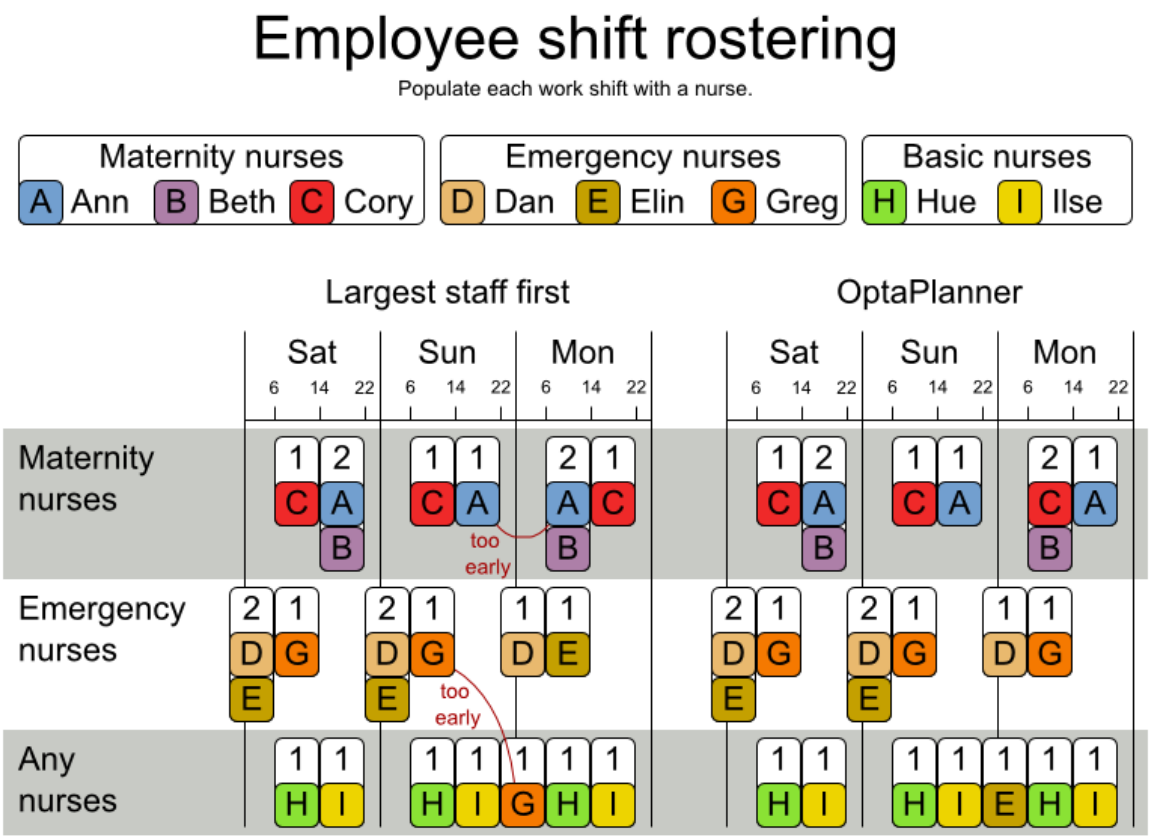
KIE OptaPlanner Deep Dive – Employee Rostering

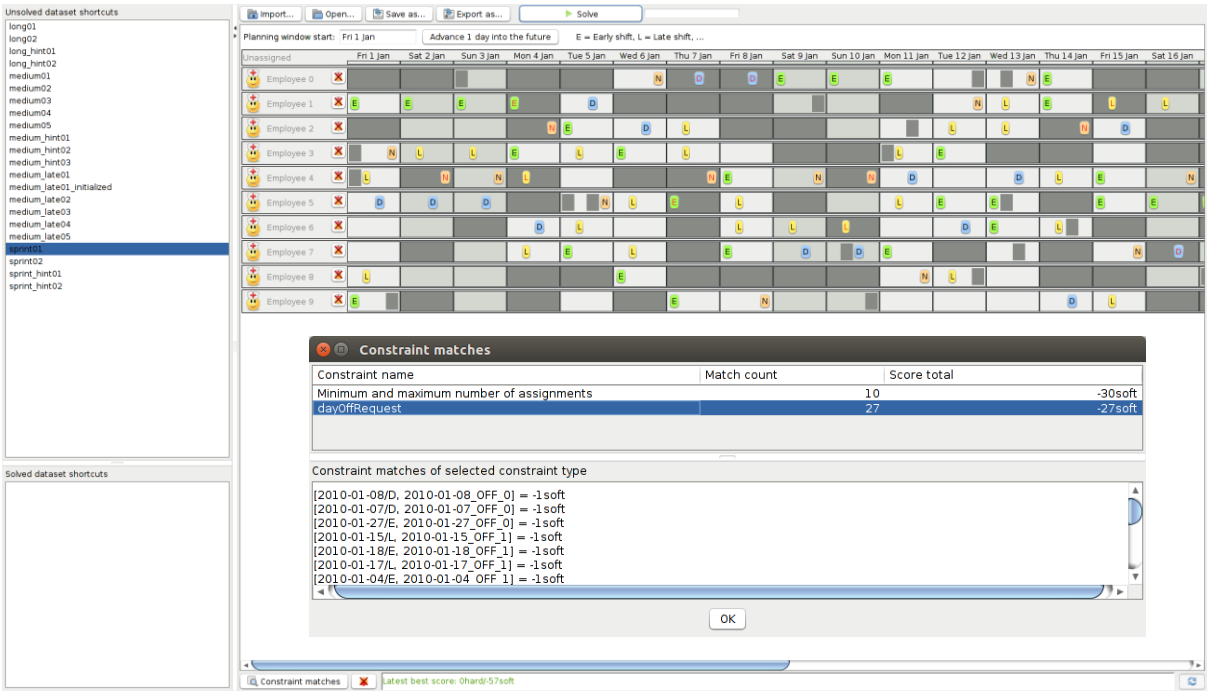
- **Business Scenario / Problem Description**
- **For each shift, assign a nurse to work that shift.**
- **Hard constraints:**
  - No unassigned shifts (built-in): Every shift need to be assigned to an employee.
  - Shift conflict: An employee can have only one shift per day.
- **Soft constraints:**
  - Contract obligations. The business frequently violates these, so they decided to define these as soft constraints instead of hard constraints.
    - Minimum and maximum assignments: Each employee needs to work more than x shifts and less than y shifts (depending on their contract).
    - Minimum and maximum consecutive working days: Each employee needs to work between x and y days in a row (depending on their contract).
    - Minimum and maximum consecutive free days: Each employee needs to be free between x and y days in a row (depending on their contract).
    - Minimum and maximum consecutive working weekends: Each employee needs to work between x and y weekends in a row (depending on their contract).
    - Complete weekends: Each employee needs to work every day in a weekend or not at all.
    - Identical shift types during weekend: Each weekend shift for the same weekend of the same employee must be the same shift type.
    - Unwanted patterns: A combination of unwanted shift types in a row. For example: a late shift followed by an early shift followed by a late shift.
- **Employee wishes:**
  - Day on request: An employee wants to work on a specific day.
  - Day off request: An employee does not want to work on a specific day.
  - Shift on request: An employee wants to be assigned to a specific shift.
  - Shift off request: An employee does not want to be assigned to a specific shift.
- **Alternative skill: An employee assigned to a skill should have a proficiency in every skill required by that shift.**
- **The problem is defined by** the International Nurse Rostering Competition 2010, http://www.kuleuven-kortrijk.be/nrpcompetition

356

Nurse rostering class diagram




Employee shift rostering
Populate each work shift with a nurse.
Maternity nurses
A Ann
B Beth
C Cory
Emergency nurses
D Dan
E Elin
G Greg
Basic nurses
H Hue
I Ilse
Largest staff first
OptaPlanner
Sat
Sun
Mon
6 14 22
Maternity nurses
Emergency nurses
Any nurses
too early
too early


Unsolved dataset shortcuts
Solved dataset shortcuts
Constraint matches
Constraint name
Match count
Score total
Minimum and maximum number of assignments
10
-30soft
dayOffRequest
27
-27soft
Constraint matches of selected constraint type
OK

ANNEX 4 : KIE OPTAPLANNER EXAMPLES

KIE OptaPlanner Deep Dive – Traveling Tournament Problem (TTP)

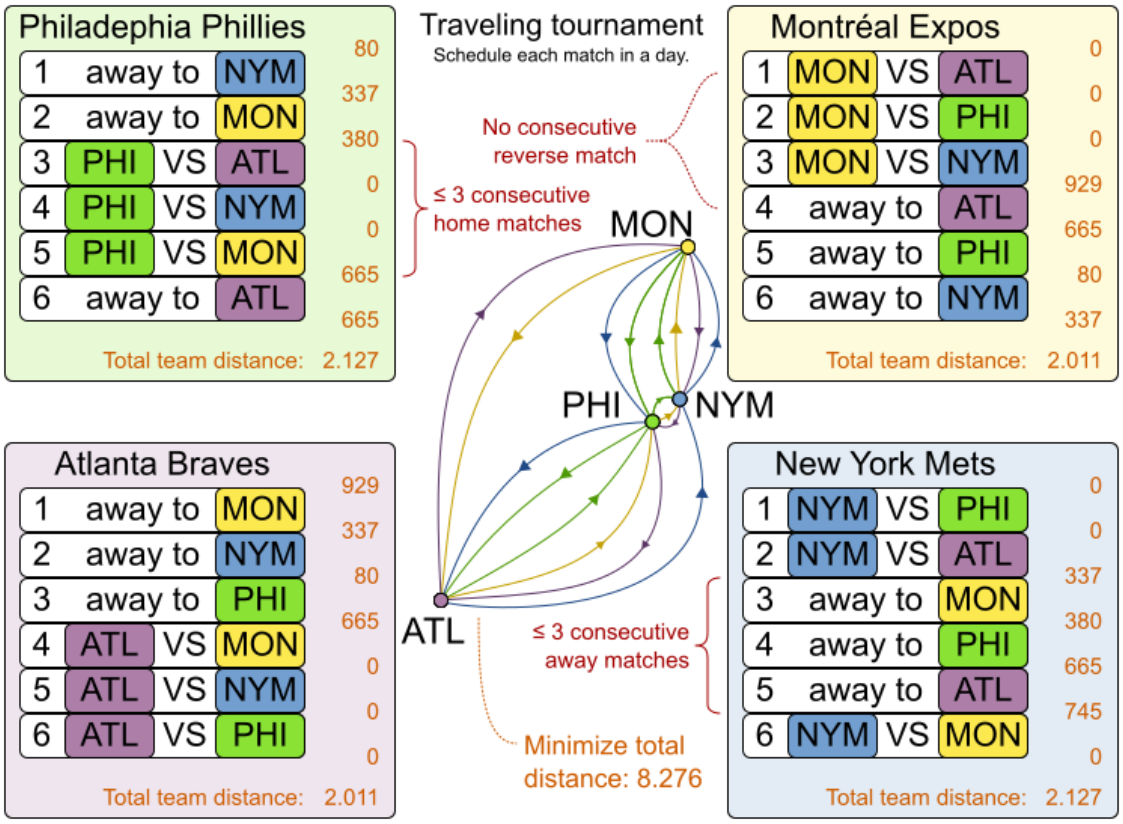
- **Business Scenario / Problem Description**
- **Schedule matches between n teams.**
- **Hard constraints:**
  - Each team plays twice against every other team: once home and once away.
  - Each team has exactly one match on each timeslot.
  - No team must have more than three consecutive home or three consecutive away matches.
  - No repeaters: no two consecutive matches of the same two opposing teams.
- **Soft constraints:**
  - Minimize the total distance traveled by all teams.
- **The problem is defined on Michael Trick's website.**
  http://mat.tepper.cmu.edu/TOURN/

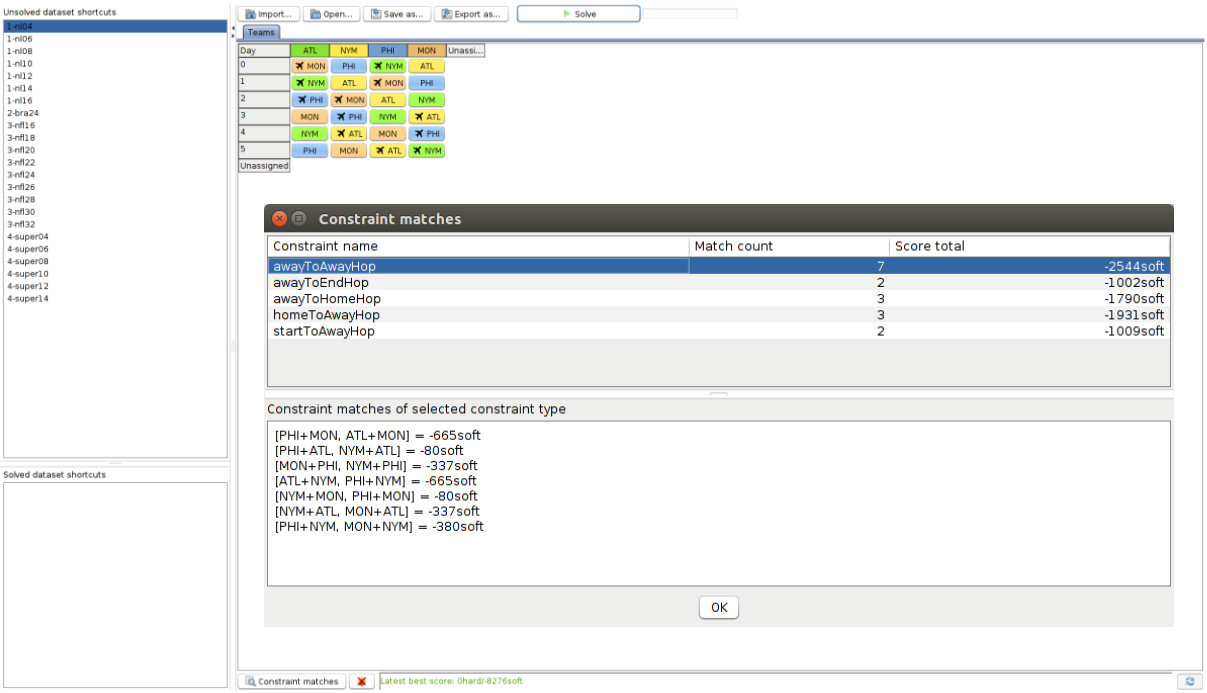
366

ANNEX 4 : KIE OPTAPLANNER EXAMPLES

Exercise: Draw Class Diagram

369


Philadephia Phillies
1 away to NYM 80
2 away to MON 337
3 PHI VS ATL 380
4 PHI VS NYM 0
5 PHI VS MON 0
6 away to ATL 665
665
Total team distance: 2.127
Traveling tournament
Schedule each match in a day.
No consecutive reverse match
≤ 3 consecutive home matches
MON
PHI
NYM
ATL
≤ 3 consecutive away matches
Minimize total distance: 8.276
Montréal Expos
1 MON VS ATL 0
2 MON VS PHI 0
3 MON VS NYM 0
4 away to ATL 929
5 away to PHI 665
6 away to NYM 80
337
Total team distance: 2.011
Atlanta Braves
1 away to MON 929
2 away to NYM 337
3 away to PHI 80
4 ATL VS MON 665
5 ATL VS NYM 0
6 ATL VS PHI 0
0
Total team distance: 2.011
New York Mets
1 NYM VS PHI 0
2 NYM VS ATL 0
3 away to MON 337
4 away to PHI 380
5 away to ATL 665
6 NYM VS MON 745
0
Total team distance: 2.127


Constraint matches
Constraint name
Match count
Score total
awayToAwayHop 7 -2544soft
awayToEndHop 2 -1002soft
awayToHomeHop 3 -1790soft
homeToAwayHop 3 -1931soft
startToAwayHop 2 -1009soft
Constraint matches of selected constraint type
[PHI+MON, ATL+MON] = -665soft
[PHI+ATL, NYM+ATL] = -80soft
[MON+PHI, NYM+PHI] = -337soft
[ATL+NYM, PHI+NYM] = -665soft
[NYM+MON, PHI+MON] = -80soft
[NYM+ATL, MON+ATL] = -337soft
[PHI+NYM, MON+NYM] = -380soft
OK

ANNEX 4 : KIE OPTAPLANNER EXAMPLES

KIE OptaPlanner Deep Dive – Cheap Time Scheduling

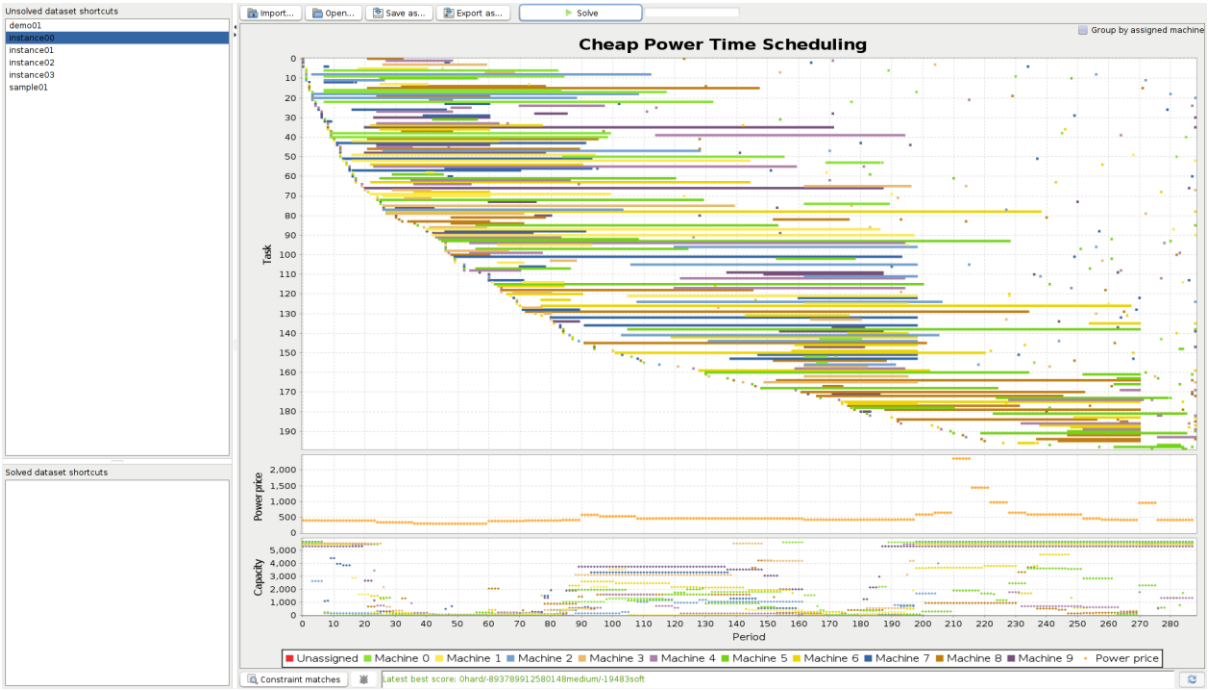
- **Business Scenario / Problem Description**
- **Schedule all tasks in time and on a machine to minimize power cost. Power prices differs in time. This is a form of job shop scheduling.**
- **Hard constraints:**
  - Start time limits: each task must start between its earliest start and latest start limit.
  - Maximum capacity: the maximum capacity for each resource for each machine must not be exceeded.
  - Startup and shutdown: each machine must be active in the periods during which it has assigned tasks. Between tasks it is allowed to be idle to avoid startup and shutdown costs.
- **Medium constraints:**
  - Power cost: minimize the total power cost of the whole schedule.
  - Machine power cost: Each active or idle machine consumes power, which infers a power cost (depending on the power price during that time).
  - Task power cost: Each task consumes power too, which infers a power cost (depending on the power price during its time).
  - Machine startup and shutdown cost: Every time a machine starts up or shuts down, an extra cost is inflicted.
- **Soft constraints (addendum to the original problem definition):**
  - Start early: prefer starting a task sooner rather than later.

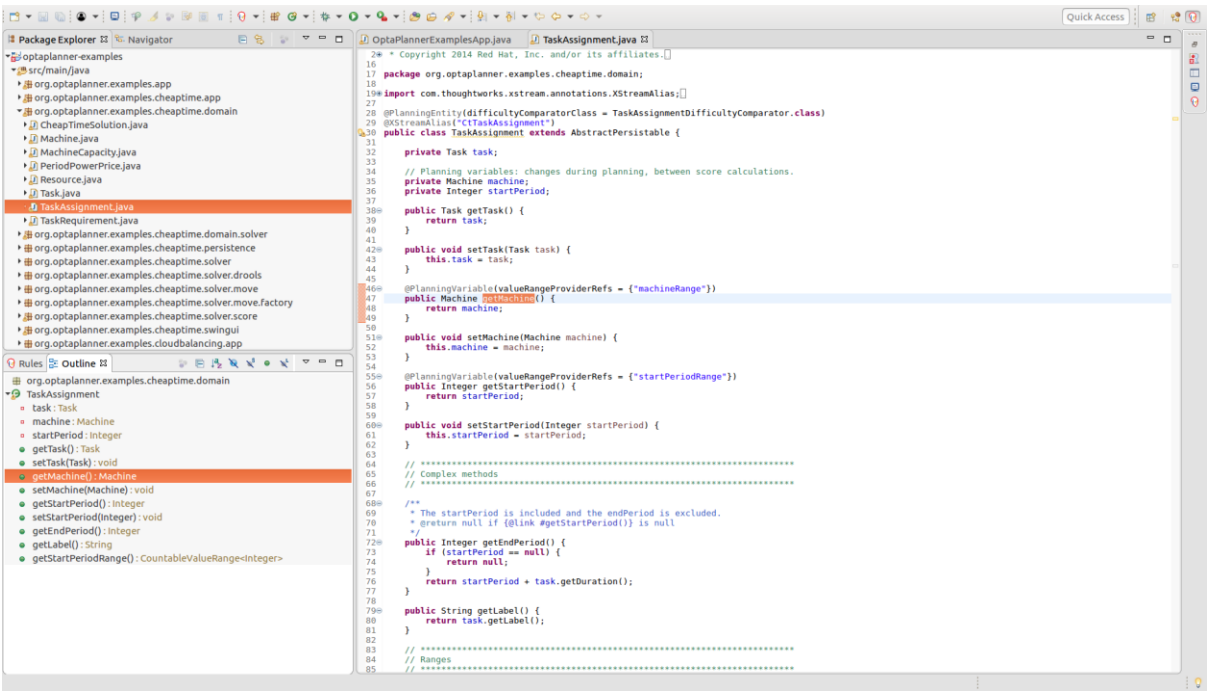
375

ANNEX 4 : KIE OPTAPLANNER EXAMPLES

Exercise: Draw Class Diagram

377


Unsolved dataset shortcuts
demo01
instance00
instance01
instance02
instance03
sample01
Solved dataset shortcuts
Import...
Open...
Save as...
Export as...
Solve
Group by assigned machine
Cheap Power Time Scheduling
Task
Power price
Capacity
Period
Unassigned
Machine 0
Machine 1
Machine 2
Machine 3
Machine 4
Machine 5
Machine 6
Machine 7
Machine 8
Machine 9
Power price
Constraint matches


Package Explorer
Navigator
optaplanner-examples
src/main/java
org.optaplanner.examples.app
org.optaplanner.examples.cheaptime.app
org.optaplanner.examples.cheaptime.domain
CheapTimeSolution.java
Machine.java
MachineCapacity.java
PeriodPowerPrice.java
Resource.java
Task.java
TaskAssignment.java
TaskRequirement.java
org.optaplanner.examples.cheaptime.domain.solver
org.optaplanner.examples.cheaptime.persistence
org.optaplanner.examples.cheaptime.solver
org.optaplanner.examples.cheaptime.solver.drools
org.optaplanner.examples.cheaptime.solver.move
org.optaplanner.examples.cheaptime.solver.move.factory
org.optaplanner.examples.cheaptime.solver.score
org.optaplanner.examples.cheaptime.swingui
org.optaplanner.examples.cloudbalancing.app
Rules
Outline
org.optaplanner.examples.cheaptime.domain
TaskAssignment
task : Task
machine : Machine
startPeriod : Integer
getTask() : Task
setTask(Task) : void
getMachine() : Machine
setMachine(Machine) : void
getStartPeriod() : Integer
setStartPeriod(Integer) : void
getEndPeriod() : Integer
getLabel() : String
getStartPeriodRange() : CountableValueRange<Integer>
Quick Access
OptaPlannerExamplesApp.java
TaskAssignment.java

ANNEX 4 : KIE OPTAPLANNER EXAMPLES

KIE OptaPlanner Deep Dive – Investment Asset Class Allocation

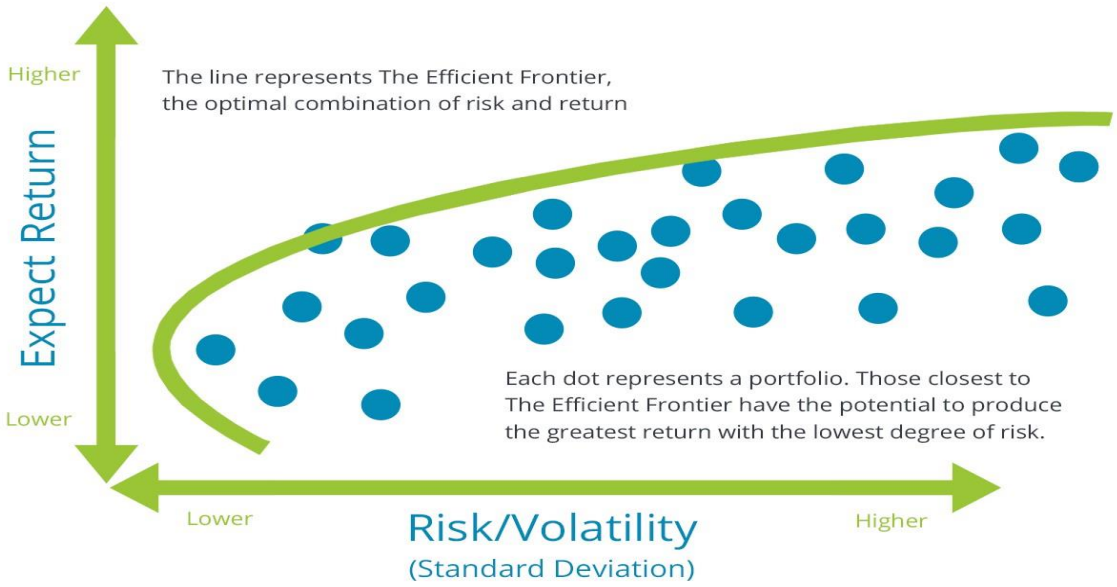
- **Business Scenario / Problem Description**
- **Decide the relative quantity to invest in each asset class.**
- **Hard constraints:**
  - Risk maximum: the total standard deviation must not be higher than the standard deviation maximum.
  - Total standard deviation calculation takes asset class correlations into account by applying Markowitz Portfolio Theory.
  - Region maximum: Each region has a quantity maximum.
  - Sector maximum: Each sector has a quantity maximum.
- **Soft constraints:**
  - Maximize expected return.

384

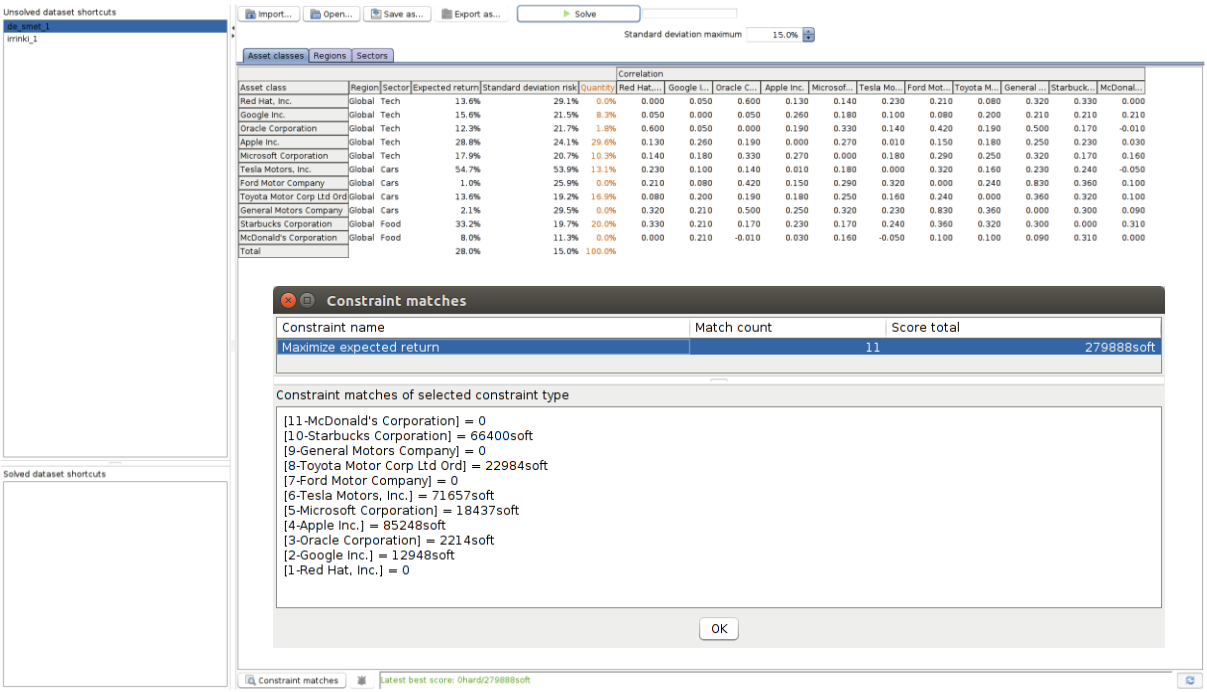
ANNEX 4 : KIE OPTAPLANNER EXAMPLES

Exercise: Draw Class Diagram

388



Link https://www.guidedchoice.com/video/dr-harry-markowitz-father-of-modern-portfolio-theory/



ANNEX 4 : KIE OPTAPLANNER EXAMPLES

KIE OptaPlanner Deep Dive – Conference Scheduling

- **Business Scenario / Problem Description**
- **Assign each conference talk to a timeslot and a room, after the talks have been accepted.**
- **Hard constraints (unless configured otherwise):**
  - Talk type of timeslot: The type of a talk must match the timeslot's talk type.
  - Room unavailable timeslots: A talk's room must be available during the talk's timeslot.
  - Room conflict: Two talks can't use the same room during overlapping timeslots.
  - Speaker unavailable timeslots: Every talk's speaker must be available during the talk's timeslot.
  - Speaker conflict: Two talks can't share a speaker during overlapping timeslots.
  - Generic purpose timeslot and room tags
    - Speaker required timeslot tags: If a speaker has a required timeslot tag, then all his/her talks must be assigned to a timeslot with that tag.
    - Speaker prohibited timeslot tags: If a speaker has a prohibited timeslot tag, then all his/her talks cannot be assigned to a timeslot with that tag.
    - Talk required timeslot tags: If a talk has a required timeslot tag, then it must be assigned to a timeslot with that tag.
    - Talk prohibited timeslot tags: If a talk has a prohibited timeslot tag, then it cannot be assigned to a timeslot with that tag.
    - Speaker required room tags: If a speaker has a required room tag, then all his/her talks must be assigned to a room with that tag.
    - Speaker prohibited room tags: If a speaker has a prohibited room tag, then all his/her talks cannot be assigned to a room with that tag.
    - Talk required room tags: If a talk has a required room tag, then it must be assigned to a room with that tag.
    - Talk prohibited room tags: If a talk has a prohibited room tag, then it cannot be assigned to a room with that tag.
  - Talk prerequisite talks: A talk must be scheduled after all its prerequisite talks.
  - Consecutive talks pause: A speaker who has more than one talk must have a break between them.
  - Talk mutually-exclusive-talks tags: Talks that share such tags must not be scheduled in overlapping timeslots.
- **Medium constraints (unless configured otherwise):**
  - Published timeslot: A published talk must not be scheduled at a different timeslot than currently published. If a hard constraint's input data changes after publishing (such as speaker unavailability), then this medium constraint will be minimally broken to attain a new feasible solution.

394

ANNEX 4 : KIE OPTAPLANNER EXAMPLES

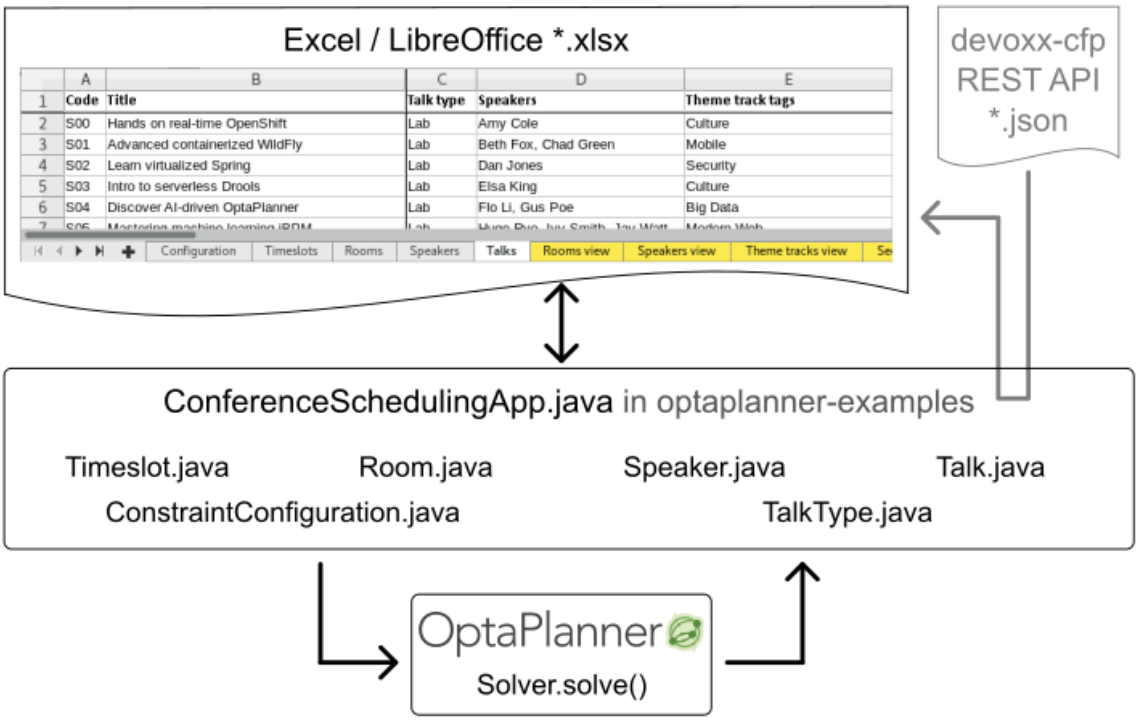
KIE OptaPlanner Deep Dive – Conference Scheduling

- **Business Scenario / Problem Description**
- **Assign each conference talk to a timeslot and a room, after the talks have been accepted.**
- **Hard constraints (unless configured otherwise):**
  - Talk type of timeslot: The type of a talk must match the timeslot's talk type.
  - Room unavailable timeslots: A talk's room must be available during the talk's timeslot.
  - Room conflict: Two talks can't use the same room during overlapping timeslots.
  - Speaker unavailable timeslots: Every talk's speaker must be available during the talk's timeslot.
  - Speaker conflict: Two talks can't share a speaker during overlapping timeslots.
  - Generic purpose timeslot and room tags
    - Speaker required timeslot tags: If a speaker has a required timeslot tag, then all his/her talks must be assigned to a timeslot with that tag.
    - Speaker prohibited timeslot tags: If a speaker has a prohibited timeslot tag, then all his/her talks cannot be assigned to a timeslot with that tag.
    - Talk required timeslot tags: If a talk has a required timeslot tag, then it must be assigned to a timeslot with that tag.
    - Talk prohibited timeslot tags: If a talk has a prohibited timeslot tag, then it cannot be assigned to a timeslot with that tag.
    - Speaker required room tags: If a speaker has a required room tag, then all his/her talks must be assigned to a room with that tag.
    - Speaker prohibited room tags: If a speaker has a prohibited room tag, then all his/her talks cannot be assigned to a room with that tag.
    - Talk required room tags: If a talk has a required room tag, then it must be assigned to a room with that tag.
    - Talk prohibited room tags: If a talk has a prohibited room tag, then it cannot be assigned to a room with that tag.
  - Talk prerequisite talks: A talk must be scheduled after all its prerequisite talks.
  - Consecutive talks pause: A speaker who has more than one talk must have a break between them.
  - Talk mutually-exclusive-talks tags: Talks that share such tags must not be scheduled in overlapping timeslots.
- **Medium constraints (unless configured otherwise):**
  - Published timeslot: A published talk must not be scheduled at a different timeslot than currently published. If a hard constraint's input data changes after publishing (such as speaker unavailability), then this medium constraint will be minimally broken to attain a new feasible solution.

394

Conference scheduling architecture

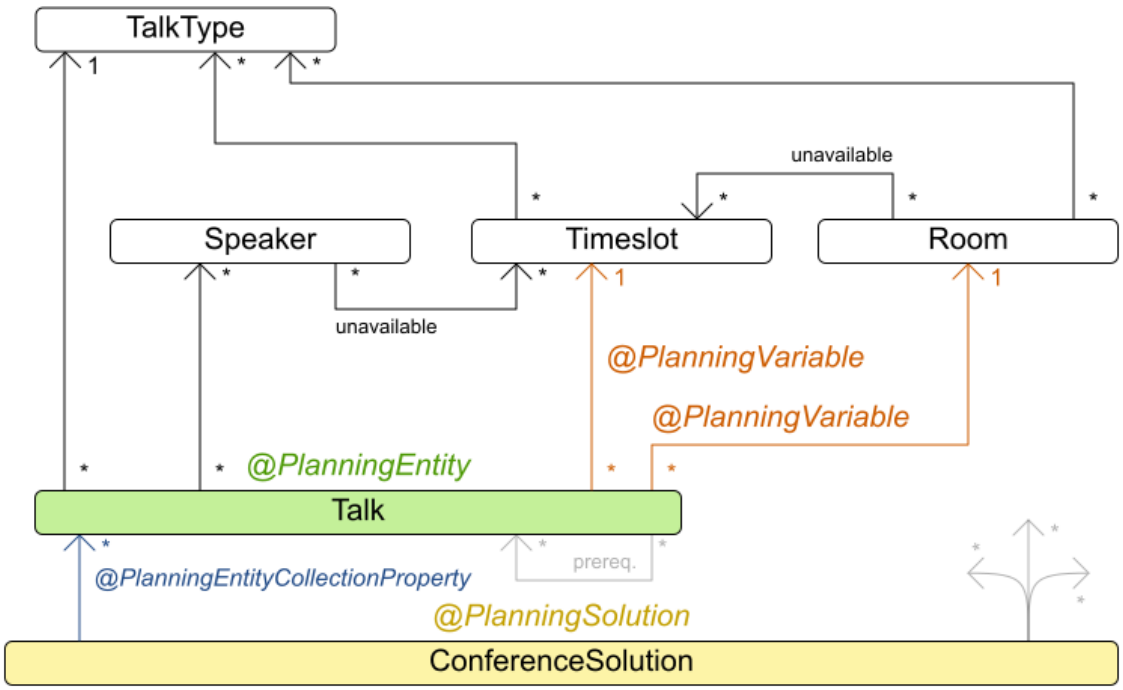
Planner works with plain Java objects that are read/written to *.xlsx

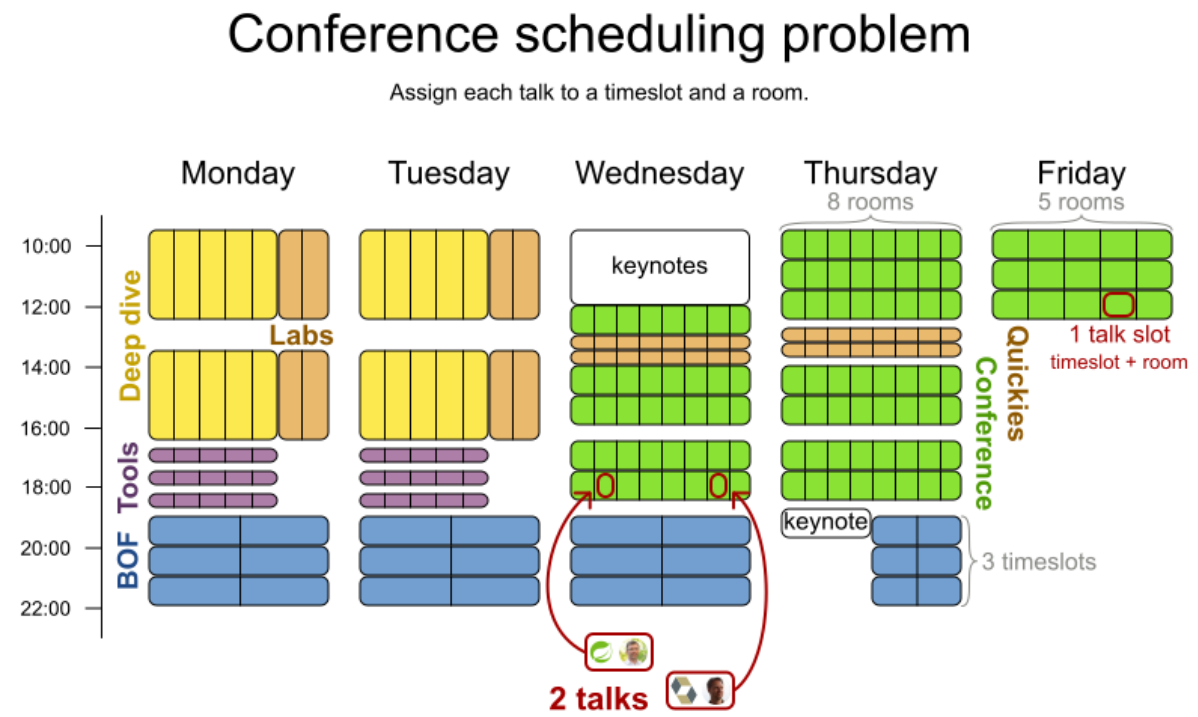


	A	B	C	D	E
1	Code	Title	Talk type	Speakers	Theme track tags
2	S00	Hands on real-time OpenShift	Lab	Amy Cole	Culture
3	S01	Advanced containerized WildFly	Lab	Beth Fox, Chad Green	Mobile
4	S02	Learn virtualized Spring	Lab	Dan Jones	Security
5	S03	Intro to serverless Drools	Lab	Elsa King	Culture
6	S04	Discover AI-driven OptaPlanner	Lab	Flo Li, Gus Poe	Big Data

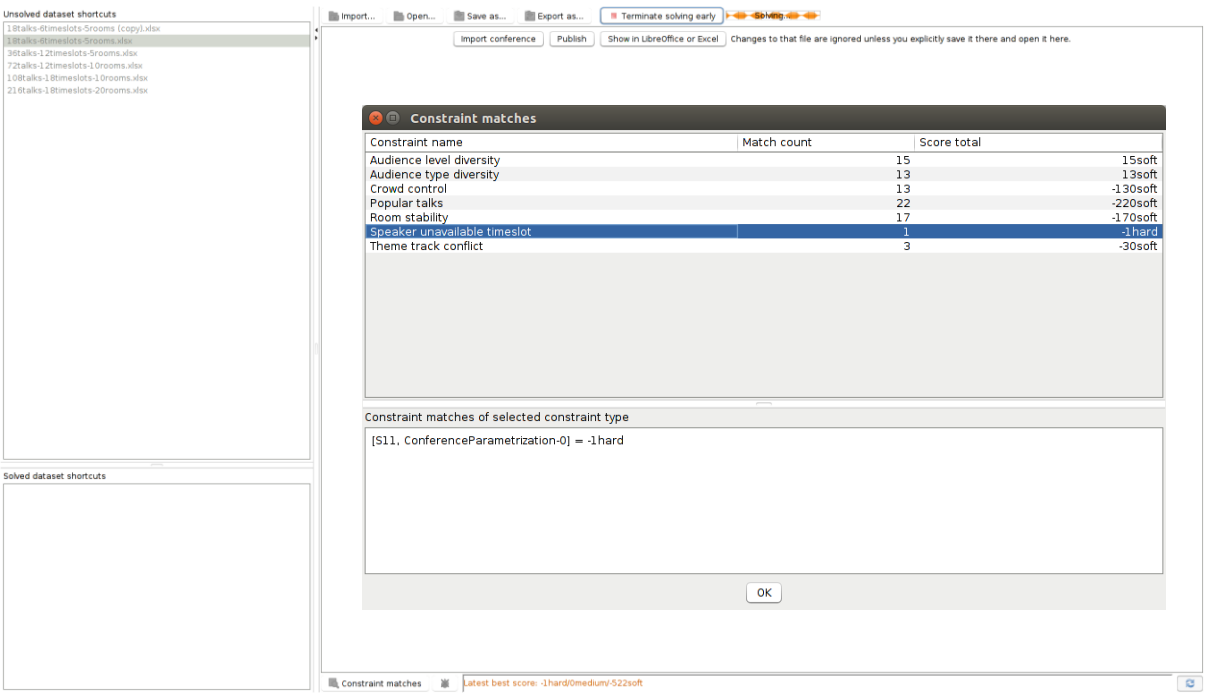
Configuration | Timeslots | Rooms | Speakers | Talks | Rooms view | Speakers view | Theme tracks view

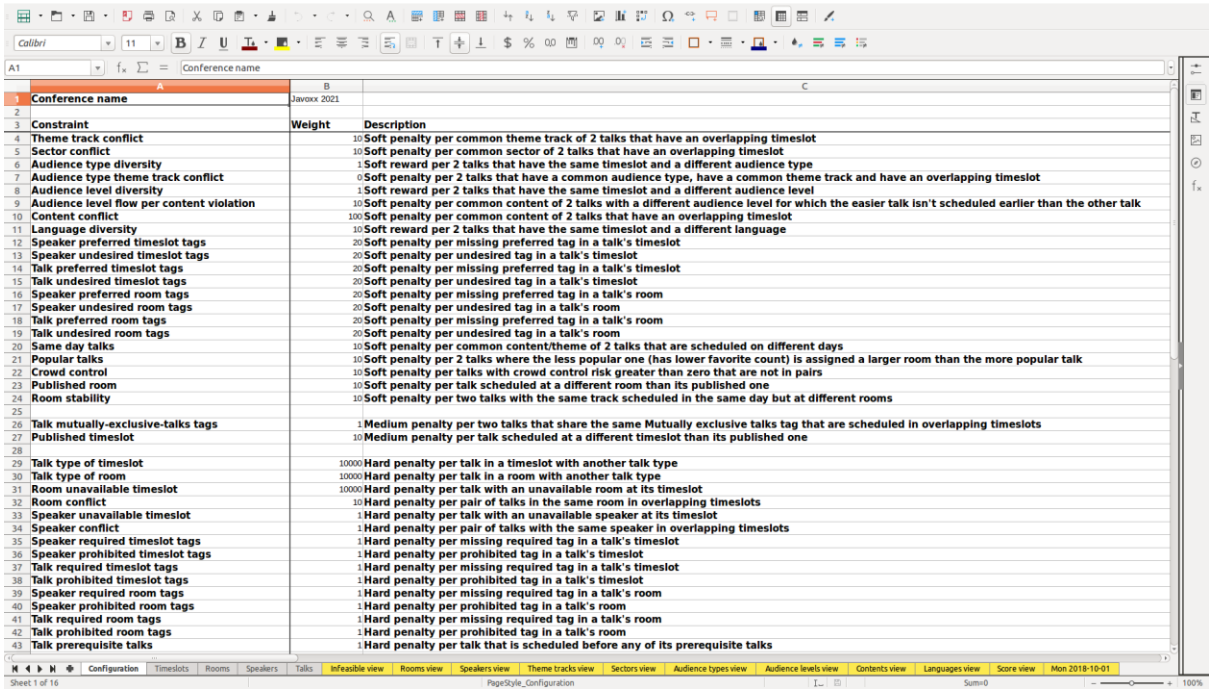
Conference scheduling class diagram



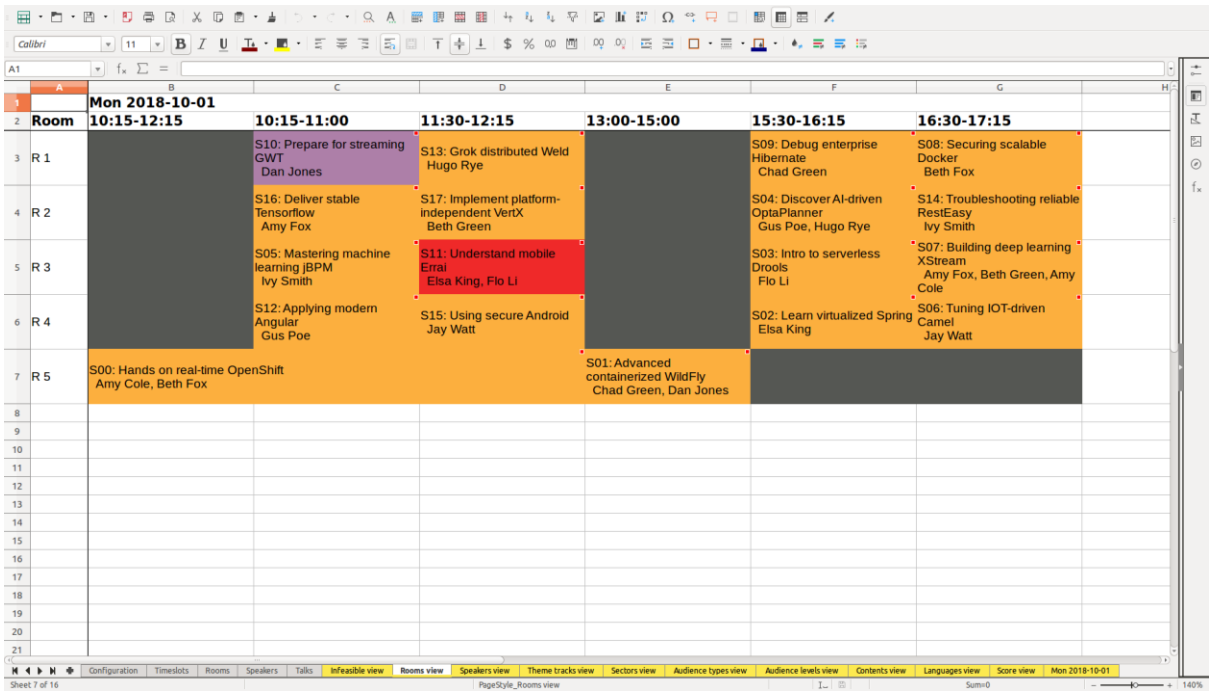


Devoxx Belgium assigns 214 talks to 40 timeslots and 10 rooms for 3500 attendees.





Conference name	Javoxx 2021	
Constraint	**Weight**	**Description**
Theme track conflict	10	Soft penalty per common theme track of 2 talks that have an overlapping timeslot
Sector conflict	10	Soft penalty per common sector of 2 talks that have an overlapping timeslot
Audience type diversity	1	Soft reward per 2 talks that have the same timeslot and a different audience type
Audience type theme track conflict	0	Soft penalty per 2 talks that have a common audience type, have a common theme track and have an overlapping timeslot
Audience level diversity	1	Soft reward per 2 talks that have the same timeslot and a different audience level
Audience level flow per content violation	10	Soft penalty per common content of 2 talks with a different audience level for which the easier talk isn't scheduled earlier than the other talk
Content conflict	100	Soft penalty per common content of 2 talks that have an overlapping timeslot
Language diversity	10	Soft reward per 2 talks that have the same timeslot and a different language
Speaker preferred timeslot tags	20	Soft penalty per missing preferred tag in a talk's timeslot
Speaker undesired timeslot tags	20	Soft penalty per undesired tag in a talk's timeslot
Talk preferred timeslot tags	20	Soft penalty per missing preferred tag in a talk's timeslot
Talk undesired timeslot tags	20	Soft penalty per undesired tag in a talk's timeslot
Speaker preferred room tags	20	Soft penalty per missing preferred tag in a talk's room
Speaker undesired room tags	20	Soft penalty per undesired tag in a talk's room
Talk preferred room tags	20	Soft penalty per missing preferred tag in a talk's room
Talk undesired room tags	20	Soft penalty per undesired tag in a talk's room
Same day talks	10	Soft penalty per common content/theme of 2 talks that are scheduled on different days
Popular talks	10	Soft penalty per 2 talks where the less popular one (has lower favorite count) is assigned a larger room than the more popular talk
Crowd control	10	Soft penalty per talks with crowd control risk greater than zero that are not in pairs
Published room	10	Soft penalty per talk scheduled at a different room than its published one
Room stability	10	Soft penalty per two talks with the same track scheduled in the same day but at different rooms
Talk mutually-exclusive-talks tags	1	Medium penalty per two talks that share the same Mutually exclusive talks tag that are scheduled in overlapping timeslots
Published timeslot	10	Medium penalty per talk scheduled at a different timeslot than its published one
Talk type of timeslot	10000	Hard penalty per talk in a timeslot with another talk type
Talk type of room	10000	Hard penalty per talk in a room with another talk type
Room unavailable timeslot	10000	Hard penalty per talk with an unavailable room at its timeslot
Room conflict	10	Hard penalty per pair of talks in the same room in overlapping timeslots
Speaker unavailable timeslot	1	Hard penalty per talk with an unavailable speaker at its timeslot
Speaker conflict	1	Hard penalty per pair of talks with the same speaker in overlapping timeslots
Speaker required timeslot tags	1	Hard penalty per missing required tag in a talk's timeslot
Speaker prohibited timeslot tags	1	Hard penalty per prohibited tag in a talk's timeslot
Talk required timeslot tags	1	Hard penalty per missing required tag in a talk's timeslot
Talk prohibited timeslot tags	1	Hard penalty per prohibited tag in a talk's timeslot
Speaker required room tags	1	Hard penalty per missing required tag in a talk's room
Speaker prohibited room tags	1	Hard penalty per prohibited tag in a talk's room
Talk required room tags	1	Hard penalty per missing required tag in a talk's room
Talk prohibited room tags	1	Hard penalty per prohibited tag in a talk's room
Talk prerequisite talks	1	Hard penalty per talk that is scheduled before any of its prerequisite talks



	Mon 2018-10-01					
Room	**10:15-12:15**	**10:15-11:00**	**11:30-12:15**	**13:00-15:00**	**15:30-16:15**	**16:30-17:15**
R 1		S10: Prepare for streaming GWT Dan Jones	S13: Grok distributed Weld Hugo Rye		S09: Debug enterprise Hibernate Chad Green	S08: Securing scalable Docker Beth Fox
R 2		S16: Deliver stable Tensorflow Amy Fox	S17: Implement platform-independent VertX Beth Green		S04: Discover AI-driven OptaPlanner Gus Poe, Hugo Rye	S14: Troubleshooting reliable RestEasy Ivy Smith
R 3		S05: Mastering machine learning jBPM Ivy Smith	S11: Understand mobile Errai Elsa King, Flo Li		S03: Intro to serverless Drools Flo Li	S07: Building deep learning XStream Amy Fox, Beth Green, Amy Cole
R 4		S12: Applying modern Angular Gus Poe	S15: Using secure Android Jay Watt		S02: Learn virtualized Spring Elsa King	S06: Tuning IOT-driven Camel Jay Watt
R 5	S00: Hands on real-time OpenShift Amy Cole, Beth Fox			S01: Advanced containerized WildFly Chad Green, Dan Jones		

ANNEX 4 : KIE OPTAPLANNER EXAMPLES

KIE OptaPlanner Deep Dive – Rock Tour

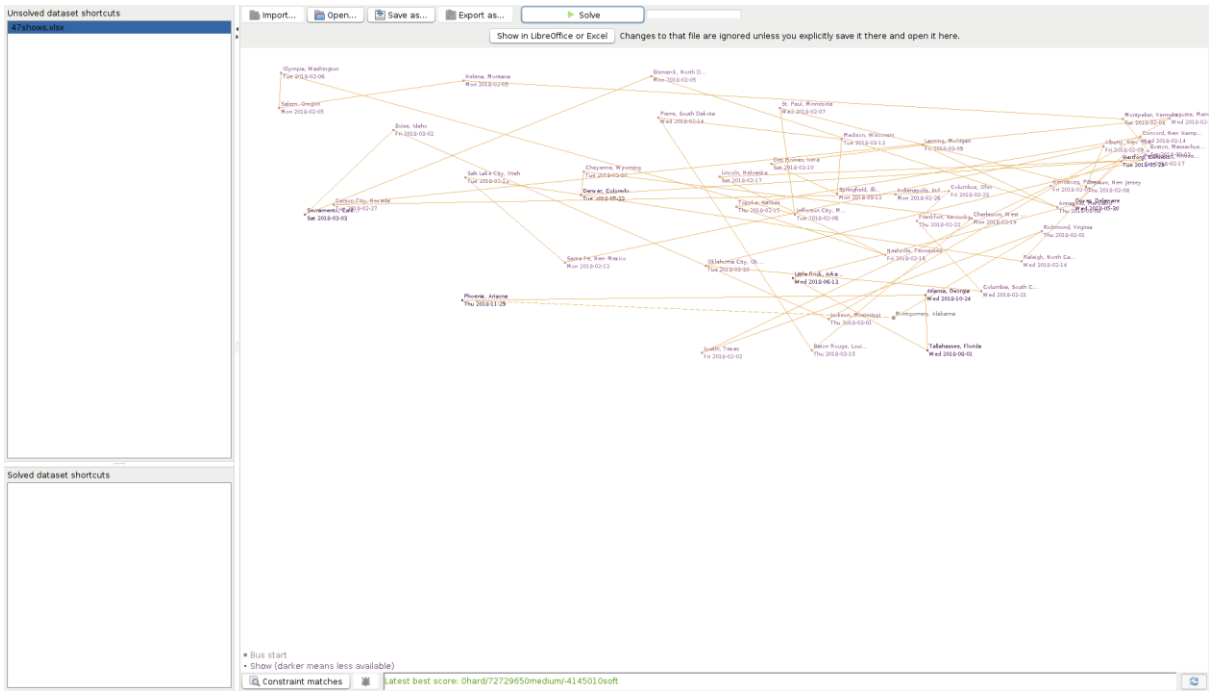
- **Business Scenario / Problem Description**
- **Drive the rock bus from rock-show to rock-show, but schedule rock-shows only on available days.**
- **Hard constraints:**
  - Schedule every required show.
  - Schedule as many shows as possible.
- **Medium constraints:**
  - Maximize revenue opportunity.
  - Minimize driving time.
  - Visit sooner than later.
- **Soft constraints:**
  - Avoid long driving times.

409

ANNEX 4 : KIE OPTAPLANNER EXAMPLES

Exercise: Draw Class Diagram

410



	Date	Venue name	City name	Driving time	Driving time per week	Latitude	Longitude	Duration (in days)	Revenue opportunity	Required	Available dates size
2	Thu 2018-02-01	Richmond, Virginia	Richmond, Virginia	0 hours 6 minutes		37.538857	-77.43364	0.5	2400000	0	224
3	Fri 2018-02-02	Austin, Texas	Austin, Texas	0 hours 2 minutes		30.27467	-97.740349	0.5	200000	0	256
4		Harrisburg, Pennsylvania	Harrisburg, Pennsylvania	0 hours 0 minutes		40.264378	-76.883598	0.5	100000	0	252
5	Sat 2018-02-03	Boston, Massachusetts	Boston, Massachusetts	0 hours 1 minutes		42.358162	-71.063698	0.5	600000	0	225
6		Montpelier, Vermont	Montpelier, Vermont	0 hours 0 minutes		44.262436	-72.580536	0.5	100000	0	209
7	Sun 2018-02-04				0 hours 12 minutes						
8	Mon 2018-02-05	Helena, Montana	Helena, Montana	0 hours 1 minutes		46.585709	-112.018417	0.5	1300000	0	215
9		Salem, Oregon	Salem, Oregon	0 hours 0 minutes		44.938461	-123.030403	0.5	1200000	0	231
10	Tue 2018-02-06	Olympia, Washington	Olympia, Washington	0 hours 0 minutes		47.035805	-122.905014	0.5	1800000	0	222
11		Jefferson City, Missouri	Jefferson City, Missouri	0 hours 3 minutes		38.579201	-92.172935	0.5	1800000	0	235
12	Wed 2018-02-07	St. Paul, Minnesota	St. Paul, Minnesota	0 hours 1 minutes		44.955097	-93.102211	1	600000	0	248
13	Thu 2018-02-08	Trenton, New Jersey	Trenton, New Jersey	0 hours 3 minutes		40.220596	-74.769913	0.5	1100000	0	227
	Fri 2018-	Albany, New	Albany, New	0 hours 0							

ANNEX 4 : KIE OPTAPLANNER EXAMPLES

KIE OptaPlanner Deep Dive – Flight Crew Scheduling

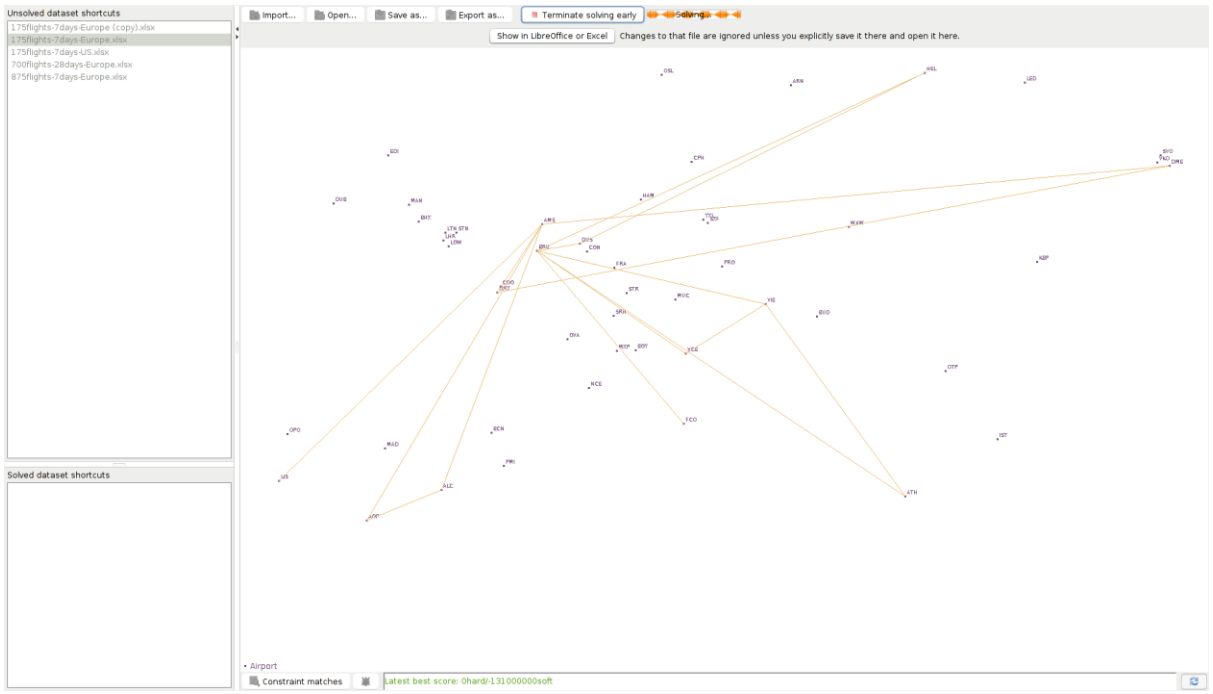
- **Business Scenario / Problem Description**
- **Assign flights to pilots and flight attendants.**
- **Hard constraints:**
  - Required skill: each flight assignment has a required skill. For example, flight AB0001 requires 2 pilots and 3 flight attendants.
  - Flight conflict: each employee can only attend one flight at the same time
  - Transfer between two flights: between two flights, an employee must be able to transfer from the arrival airport to the departure airport. For example, Ann arrives in Brussels at 10:00 and departs in Amsterdam at 15:00.
  - Employee unavailability: the employee must be available on the day of the flight. For example, Ann is on PTO on 1-Feb.
- **Soft constraints:**
  - First assignment departing from home
  - Last assignment arriving at home
  - Load balance flight duration total per employee

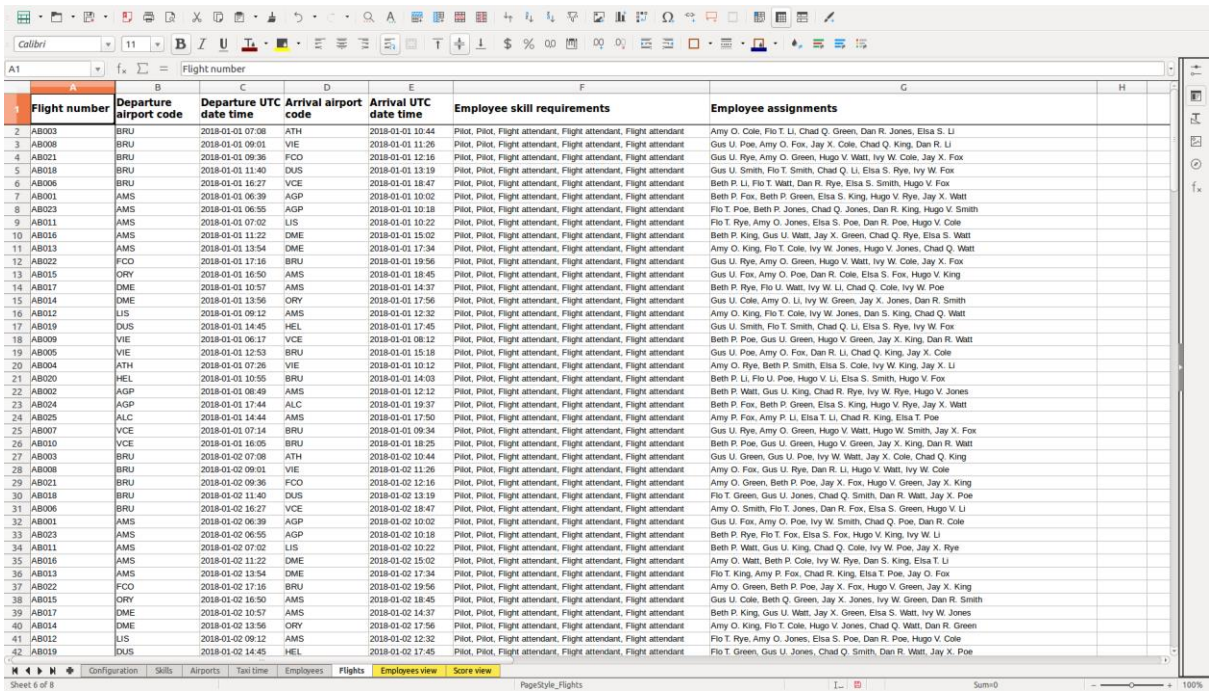
420

ANNEX 4 : KIE OPTAPLANNER EXAMPLES

Exercise: Draw Class Diagram

421





	A	B	C	D	E	F	G	H
1	Flight number	Departure airport code	Departure UTC date time	Arrival airport code	Arrival UTC date time	Employee skill requirements	Employee assignments	
2	AB003	BRU	2018-01-01 07:08	ATH	2018-01-01 10:44	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Amy O. Cole, Flo T. Li, Chad Q. Green, Dan R. Jones, Elsa S. Li	
3	AB008	BRU	2018-01-01 09:01	VIE	2018-01-01 11:26	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Gus U. Poe, Amy O. Fox, Jay X. Cole, Chad Q. King, Dan R. Li	
4	AB021	BRU	2018-01-01 09:36	FCO	2018-01-01 12:16	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Gus U. Rye, Amy O. Green, Hugo V. Watt, Ivy W. Cole, Jay X. Fox	
5	AB018	BRU	2018-01-01 11:40	DUS	2018-01-01 13:19	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Gus U. Smith, Flo T. Smith, Chad Q. Li, Elsa S. Rye, Ivy W. Fox	
6	AB006	BRU	2018-01-01 16:27	VCE	2018-01-01 18:47	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Beth P. Li, Flo T. Watt, Dan R. Rye, Elsa S. Smith, Hugo V. Fox	
7	AB001	AMS	2018-01-01 06:39	AGP	2018-01-01 10:02	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Beth P. Fox, Beth P. Green, Elsa S. King, Hugo V. Rye, Jay X. Watt	
8	AB023	AMS	2018-01-01 06:55	AGP	2018-01-01 10:18	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Flo T. Poe, Beth P. Jones, Chad Q. Jones, Dan R. King, Hugo V. Smith	
9	AB011	AMS	2018-01-01 07:02	LIS	2018-01-01 10:22	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Flo T. Rye, Amy O. Jones, Elsa S. Poe, Dan R. Poe, Hugo V. Cole	
10	AB016	AMS	2018-01-01 11:22	DME	2018-01-01 15:02	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Beth P. King, Gus U. Watt, Jay X. Green, Chad Q. Rye, Elsa S. Watt	
11	AB013	AMS	2018-01-01 13:54	DME	2018-01-01 17:34	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Amy O. King, Flo T. Cole, Ivy W. Jones, Hugo V. Jones, Chad Q. Watt	
12	AB022	FCO	2018-01-01 17:16	BRU	2018-01-01 19:56	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Gus U. Rye, Amy O. Green, Hugo V. Watt, Ivy W. Cole, Jay X. Fox	
13	AB015	ORY	2018-01-01 16:50	AMS	2018-01-01 18:45	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Gus U. Fox, Amy O. Poe, Dan R. Cole, Elsa S. Fox, Hugo V. King	
14	AB017	DME	2018-01-01 10:57	AMS	2018-01-01 14:37	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Beth P. Rye, Flo U. Watt, Ivy W. Li, Chad Q. Cole, Ivy W. Poe	
15	AB014	DME	2018-01-01 13:56	ORY	2018-01-01 17:56	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Gus U. Cole, Amy O. Li, Ivy W. Green, Jay X. Jones, Dan R. Smith	
16	AB012	LIS	2018-01-01 09:12	AMS	2018-01-01 12:32	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Amy O. King, Flo T. Cole, Ivy W. Jones, Dan S. King, Chad Q. Watt	
17	AB019	DUS	2018-01-01 14:45	HEL	2018-01-01 17:45	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Gus U. Smith, Flo T. Smith, Chad Q. Li, Elsa S. Rye, Ivy W. Fox	
18	AB009	VIE	2018-01-01 06:17	VCE	2018-01-01 08:12	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Beth P. Poe, Gus U. Green, Hugo V. Green, Jay X. King, Dan R. Watt	
19	AB005	VIE	2018-01-01 12:53	BRU	2018-01-01 15:18	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Gus U. Poe, Amy O. Fox, Dan R. Li, Chad Q. King, Jay X. Cole	
20	AB004	ATH	2018-01-01 07:26	VIE	2018-01-01 10:12	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Amy O. Rye, Beth P. Smith, Elsa S. Cole, Ivy W. King, Jay X. Li	
21	AB020	HEL	2018-01-01 10:55	BRU	2018-01-01 14:03	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Beth P. Li, Flo U. Poe, Hugo V. Li, Elsa S. Smith, Hugo V. Fox	
22	AB002	AGP	2018-01-01 08:49	AMS	2018-01-01 12:12	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Beth P. Watt, Gus U. King, Chad R. Rye, Ivy W. Rye, Hugo V. Jones	
23	AB024	AGP	2018-01-01 17:44	ALC	2018-01-01 19:37	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Beth P. Fox, Beth P. Green, Elsa S. King, Hugo V. Rye, Jay X. Watt	
24	AB025	ALC	2018-01-01 14:44	AMS	2018-01-01 17:50	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Amy P. Fox, Amy P. Li, Elsa T. Li, Chad R. King, Elsa T. Poe	
25	AB007	VCE	2018-01-01 07:14	BRU	2018-01-01 09:34	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Gus U. Rye, Amy O. Green, Hugo V. Watt, Hugo W. Smith, Jay X. Fox	
26	AB010	VCE	2018-01-01 16:05	BRU	2018-01-01 18:25	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Beth P. Poe, Gus U. Green, Hugo V. Green, Jay X. King, Dan R. Watt	
27	AB003	BRU	2018-01-02 07:08	ATH	2018-01-02 10:44	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Gus U. Green, Gus U. Poe, Ivy W. Watt, Jay X. Cole, Chad Q. King	
28	AB008	BRU	2018-01-02 09:01	VIE	2018-01-02 11:26	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Amy O. Fox, Gus U. Rye, Dan R. Li, Hugo V. Watt, Ivy W. Cole	
29	AB021	BRU	2018-01-02 09:36	FCO	2018-01-02 12:16	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Amy O. Green, Beth P. Poe, Jay X. Fox, Hugo V. Green, Jay X. King	
30	AB018	BRU	2018-01-02 11:40	DUS	2018-01-02 13:19	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Flo T. Green, Gus U. Jones, Chad Q. Smith, Dan R. Watt, Jay X. Poe	
31	AB006	BRU	2018-01-02 16:27	VCE	2018-01-02 18:47	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Amy O. Smith, Flo T. Jones, Dan R. Fox, Elsa S. Green, Hugo V. Li	
32	AB001	AMS	2018-01-02 06:39	AGP	2018-01-02 10:02	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Gus U. Fox, Amy O. Poe, Ivy W. Smith, Chad Q. Poe, Dan R. Cole	
33	AB023	AMS	2018-01-02 06:55	AGP	2018-01-02 10:18	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Beth P. Rye, Flo T. Fox, Elsa S. Fox, Hugo V. King, Ivy W. Li	
34	AB011	AMS	2018-01-02 07:02	LIS	2018-01-02 10:22	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Beth P. Watt, Gus U. King, Chad Q. Cole, Ivy W. Poe, Jay X. Rye	
35	AB016	AMS	2018-01-02 11:22	DME	2018-01-02 15:02	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Amy O. Watt, Beth P. Cole, Ivy W. Rye, Dan S. King, Elsa T. Li	
36	AB013	AMS	2018-01-02 13:54	DME	2018-01-02 17:34	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Flo T. King, Amy P. Fox, Chad R. King, Elsa T. Poe, Jay O. Fox	
37	AB022	FCO	2018-01-02 17:16	BRU	2018-01-02 19:56	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Amy O. Green, Beth P. Poe, Jay X. Fox, Hugo V. Green, Jay X. King	
38	AB015	ORY	2018-01-02 16:50	AMS	2018-01-02 18:45	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Gus U. Cole, Beth Q. Green, Jay X. Jones, Ivy W. Green, Dan R. Smith	
39	AB017	DME	2018-01-02 10:57	AMS	2018-01-02 14:37	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Beth P. King, Gus U. Watt, Jay X. Green, Elsa S. Watt, Ivy W. Jones	
40	AB014	DME	2018-01-02 13:56	ORY	2018-01-02 17:56	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Amy O. King, Flo T. Cole, Hugo V. Jones, Chad Q. Watt, Dan R. Green	
41	AB012	LIS	2018-01-02 09:12	AMS	2018-01-02 12:32	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Flo T. Rye, Amy O. Jones, Elsa S. Poe, Dan R. Poe, Hugo V. Cole	
42	AB019	DUS	2018-01-02 14:45	HEL	2018-01-02 17:45	Pilot, Pilot, Flight attendant, Flight attendant, Flight attendant	Flo T. Green, Gus U. Jones, Chad Q. Smith, Dan R. Watt, Jay X. Poe	

The End of Workshop Project Guide