

# OptaPlanner @ RJV - Business Case

## Data

### Customer

Belgium Government

- RIJKSDIENST VOOR JAARLIJKSE VAKANTIE
- OFFICE NATIONAL DES VACANCES ANNUELLES
- NATIONAL HOLIDAY ALLOWANCE OFFICE

Site: [rjv.fgov.be](http://rjv.fgov.be)

### Industry

Public Sector / Social Security

### Geography

Belgium

### Business challenge

Automate the daily distribution of +150.000 tasks to a user base of 150 users, taking into account each user's competences, their affinity with particular clients and the daily availability. The distribution rules should be able to adapt with ease to the changing organizational structure.

### Software

- OptaPlanner
  - drools-planner-5.5.0.Final
- Drools
  - drools-5.5.0 Final
- Java
  - java-1.6.0-sun-1.6.0.37.x86\_64
- OS
  - Red Hat Enterprise Linux 6 (64bit)

### Hardware

Virtualized Environment

- CPU: Intel(R) Xeon(R) CPU X5650 @ 2.67GHz - 2 Cores
- RAM: 4 GB

### Benefits

- Eliminate the paper-based and manual distribution of work, saving tons of paper each year.
- Allow the organisation to easily migrate from a task based hierarchy to a more polyvalent, client-oriented structure.

## Background

RJV is a government institution responsible for calculating the *holiday duration* and *holiday allowance* for almost 1,2 million employees in Belgium, effectively paying out a total of 2,1 billion euro every year. RJV's core vision is to perform its task at as low a cost as possible while maintaining an excellent client oriented service.

## Business challenge

As many government institutions, RJV has a very data driven business process: information on a worker's services with an employer is validated

and used to calculate the worker's holiday rights. Within the legacy system, validation errors were printed out and distributed as tasks to different departments in the organisation, applying a task oriented hierarchy: department X treats task Tx. This way of working had a number of drawbacks, the most obvious one being the huge amount of paper required to organize the work. Another factor was the lack of flexibility with the distribution system: the distribution rules were very simple, but did not support changes in organisational structure easily, eg. changing from a task oriented to a client oriented or a competence based structure.

RJV's goal was to eliminate the paper based distribution and to introduce a more flexible way to distribute work, respecting the following constraints:

- Assign tasks based on a user's competences.
- Prefer user's which have already worked for the client related to a task (client affinity). Depending on the type of client and the urgency of the task, this could be a hard or a soft requirement.
- Take into account a task's deadline and priority in the distribution: urgent tasks must be assigned first.
- Include the user's work schedule in the distribution: distribute for maximum 8 hours a day to each user and do not distribute to absent users.

The number of tasks to be taken into account varies within the span of a year with estimated peaks up to 200 000 tasks to be distributed between up to 150 users.

The distribution was to be run overnight, lending the process a window of about 6 hours to determine a solution. Infrastructure requirements must be "reasonable": eg max 4 cores and 8GB of RAM.

A java-based solution which reuses our application's model was preferred.

The project was executed with a mixed development team of Smals consultants, [www.smals.be](http://www.smals.be), and RJV developers.

## Solution

In an early prototype of the new work distribution process, the development team wrote a procedural algorithm to distribute tasks according to the new concepts (competence, affinity,...) and related constraints. The prototype allowed us to conclude that conceptually the new process was feasible, but that at the implementation level two problems had yet to be resolved:

1. Scalability: the algorithm had to make decision based on the state of the whole set of tasks.
2. Flexibility: a change in the importance of the individual constraints could at the worst imply a rewrite of the procedural algorithm.

Research on planning and optimization tools that offered a declarative way of describing constraints, quickly led us to the open source OptaPlanner. Since we were already using Drools Expert for business validation rules, it was a logical step to look more closely to JBoss' planner solution. The problem statements solved with OptaPlanner seemed like a perfect match for our work distribution context. Commercially available alternatives were quickly discarded because of budgetary reasons.

Working with OptaPlanner, we quickly discovered that the declarative way of writing constraints allowed us to effortlessly map the business requirements on the implementation.

The provided benchmark suite was invaluable when evaluating changes in constraint definition and aided us in choosing the correct optimization algorithm.

When the complexity of the requirements increased, more detailed knowledge of OptaPlanner became necessary, including many tips and tricks which were not self-evident. Fortunately, with each new release of OptaPlanner, the user guide became more comprehensive on how to approach certain problems, thus allowing us to advance swiftly in our development.

The RJV-development team quickly decided to give OptaPlanner it's own isolated execution environment. Integration with our application was realized by serializing in- and output to xml and transferring the resulting files between our main application server and the planner server. To make the memory footprint as small as possible, a light-weight model of the business domain was introduced.

Since February 2013 a first version of the automatic work distribution is running in the production environment. The planner receives a 2 hour window to assign 50 000 tasks to about 100 users and produces a more than acceptable result.

## Benefits

- Flexibility when customizing constraints.
- Low system requirements
- Continuously improving feature set with each release allowing for easier configuration.

## Future

A second version of the RJV's work distribution planner needs to address a number of issues:

- a performance issue concerning a custom pillar switch-like move, which limits the maximum number tasks that can be distributed to about 50K, instead of the 200K we were aiming for. The custom move proved to be necessary to escape from local minima.
- a number of remarks from the business users which, as anticipated, imply fine-tuning the weights of the different constraints.

RJV is planning two software upgrades which may result in a significant performance gain for OptaPlanner:

- [OptaPlanner 6](#)
- Java 7 (1.7.0\_21 or higher which uses TimSort by default)