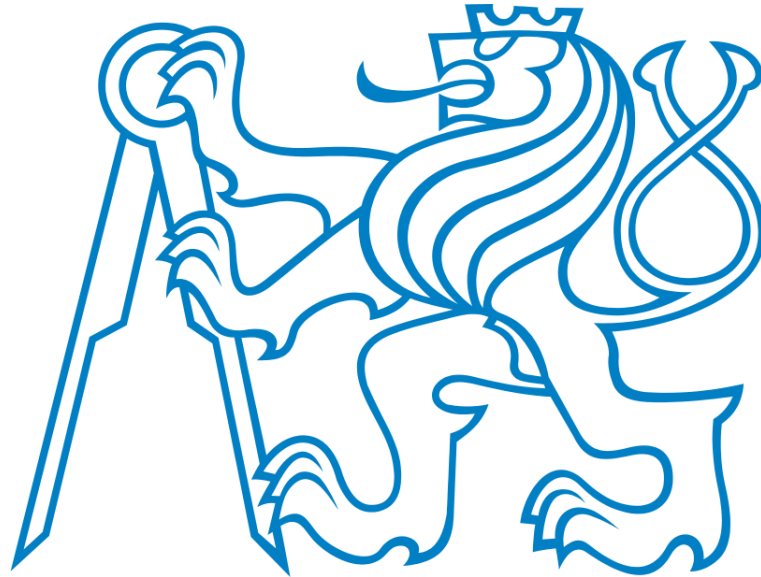


Czech Technical University in Prague  
Faculty of Electrical Engineering



---

**TOOL FOR DECISION SUPPORT IN  
PRODUCTION PLANNING AND  
SCHEDULING**

---

Bachelor Thesis

**Author: Peter Mathia**

Department of Cybernetics  
Supervisor: Ing. Pavel Vrba Ph.D.



## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

**Student:** Peter Mathia  
**Studijní program:** Kybernetika a robotika (bakalářský)  
**Obor:** Robotika  
**Název tématu:** Nástroj pro podporu rozhodování v plánování a rozvrhování výroby

### Pokyny pro vypracování:

Cílem práce je vytvořit nástroj, který bude primárně sloužit k vizualizaci výsledků rozvrhovacích algoritmů implementovaných v rámci projektu ARUM. Nástroj bude naimplementován v jazyce Java s využitím grafické knihovny Java FX. Nástroj bude nabízet tuto funkcionalitu:

1. Zobrazení rozvrhu úloh výrobní stanice ve formě Ganttova diagramu se zobrazením závislosti úloh a kritické cesty.
2. Jednoduchá editace parametrů výrobní stanice (přidávání a odebrání nástrojů a lidských zdrojů) a samotného rozvrhu (posouvání úloh v čase a fixování úlohy).
3. Zobrazení stupně využití a nedostupnosti jednotlivých zdrojů v čase.
4. Kompatibilita s ontologiemi ARUM Core ontology a ARUM Events ontology ve smyslu formátu vstupních a výstupních dat nástroje.

### Seznam odborné literatury:

- [1] Luz, S. ; Masoodian, M. ; McKenzie, D. ; Broeck, W.V., Chronos: A Tool for Interactive Scheduling and Visualisation of Task Hierarchies. 13th International Conference on Information Visualisation, 2009, pp. 241-246.
- [2] Vrba, P., Radakovič, M., Obitko, M., Mařík, V.: Semantic Technologies: Latest Advances in Agent-based Manufacturing Control Systems. International Journal of Production Research, Vol. 49, Issue 5, 2011, pp. 1483-1496.

**Vedoucí bakalářské práce:** Ing. Pavel Vrba, Ph.D.

**Platnost zadání:** do konce letního semestru 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic  
**vedoucí katedry**

prof. Ing. Pavel Ripka, CSc.  
**děkan**

V Praze dne 10. 1. 2014

## BACHELOR PROJECT ASSIGNMENT

**Student:** Peter Mathia

**Study programme:** Cybernetics and Robotics

**Specialisation:** Robotics

**Title of Bachelor Project:** Tool for Decision Support in Production Planning and Scheduling

### Guidelines:

Goal of the thesis is to implement a tool used primarily for the visualization of results of scheduling algorithms, which are implemented within the ARUM project. The tool will be implanted in Java using the graphical library Java FX. The tool will provide following functionality:

1. Visualization of a workstation schedule in form of a Gantt diagram, including the task dependencies and the critical path.
2. Capability to edit the workstation parameters (adding and removing tools and human resources) and the schedule (shifting and fixing tasks in time).
3. Visualization of the level of utilization and inaccessibility of resources in time.
4. Compatibility with ARUM Core and ARUM Events ontologies in terms of input and output data format.

### Bibliography/Sources:

- [1] Luz, S. ; Masoodian, M. ; McKenzie, D. ; Broeck, W.V., Chronos: A Tool for Interactive Scheduling and Visualisation of Task Hierarchies. 13th International Conference on Information Visualisation, 2009, pp. 241-246.
- [2] Vrba, P., Radakovič, M., Obitko, M., Mařík, V.: Semantic Technologies: Latest Advances in Agent-based Manufacturing Control Systems. International Journal of Production Research, Vol. 49, Issue 5, 2011, pp. 1483-1496.

**Bachelor Project Supervisor:** Ing. Pavel Vrba, Ph.D.

**Valid until:** the end of the summer semester of academic year 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic  
**Head of Department**

prof. Ing. Pavel Ripka, CSc.  
**Dean**

Prague, January 10, 2014

## **Acknowledgements**

I'd like to thank my supervisor, Ing Pavel Vrba, PhD for his help during work on this thesis and also my family for their support during my study.

## **Prohlášení autora práce**

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne .....

.....

Podpis autora práce

## **Abstract**

The goal of the thesis is to create a tool used primarily for the visualization of results of scheduling algorithms, which are implemented within the ARUM project. The tool is implemented in Java using the graphical library Java FX and provides the following functionality: visualization of a workstation schedule in form of a Gantt diagram, including the task dependencies and the critical path, capability to edit the workstation parameters (adding and removing tools, human resources) and the schedule (shifting and fixing tasks in time), visualization of the level of utilization and inaccessibility of resources in time, compatibility with ARUM Core and ARUM Events ontologies in terms of input and output data format.

## **Abstrakt**

Cieľom tejto práce je vytvoriť nástroj, ktorý sa bude primárne využívať na vizualizáciu výsledkov rozvrhovacích algoritmov, ktoré sú implementované v rámci projektu ARUM. Nástroj je implementovaný v jazyku Java s využitím grafickej knižnice Java FX a poskytuje nasledujúce funkcie: vizualizácia rozvrhu pracovnej stanice vo forme Ganttovho diagramu vrátane závislostí naplánovaných úloh a kritickej cesty, možnosť editácie parametrov pracovnej stanice (pridávanie a odoberanie dostupných nástrojov, pracovníkov) a rozvrhu (posúvanie a fixácia úloh v čase), vizualizácia úrovne využitia a nedostupnosti zdrojov v čase, kompatibilita s ARUM Core a ARUM Events ontológiami vo forme vstupných a výstupných dát.

## Table of Contents

1	Introduction.....	8
1.1.	Problem of Production Ramp-Up .....	8
1.2.	Needs in Manufacturing Management.....	8
2	ARUM project.....	9
2.1.	Overview.....	9
2.2.	Airbus Assembly line.....	10
2.3.	Gantt chart.....	12
2.3.1.	Information .....	12
2.3.2.	History .....	13
3	Used technology and algorithms.....	14
3.1.	Java FX – the rich client platform.....	14
3.2.	Key features .....	14
3.3.	Algorithms.....	16
3.3.1.	Critical Path Method (CPM).....	16
3.4.	Implementation .....	21
3.4.1.	Data structure .....	21
3.4.1.	Dynamic data loading (lazy loading) .....	22
3.4.2.	Events.....	22
4	User manual.....	23
4.1.	Features overview.....	24
4.1.1.	Table.....	24
4.1.2.	Time axis .....	25
4.1.3.	Chart detail.....	26
4.1.4.	Resources utilization .....	27
4.1.5.	Edit resources at selected station.....	27
4.1.6.	Add new Job.....	33
4.1.7.	Edit selected Job .....	39
4.1.8.	Tool settings .....	41
4.1.9.	Station problems .....	44
4.1.10.	Commit changes dialog .....	44
	Figure 40: commit changes dialog .....	44
	Conclusion .....	45
	Figure list .....	46
	References .....	48

# 1 Introduction

## 1.1. Problem of Production Ramp-Up

It is common nowadays that aviation news report on delayed or even failed aircraft ramp-up. Two major market leaders are affected: Airbus and Boeing. Aircrafts are very complex and highly customized products. There are many risks that affect a successful ramp-up of a commercial aircraft. *These risks cause a significant reduction of the targeted production rate, and as a result, costs are rising.* [1]

*The growing competition and a shorter innovation cycle lead to a rising number of ramp-ups.* [1] The complexity of the product aircraft itself is not the only challenge the future ramp-ups face. *The transition from single to series production and new customization strategies are indispensable in order to reach high production rates.* [1]

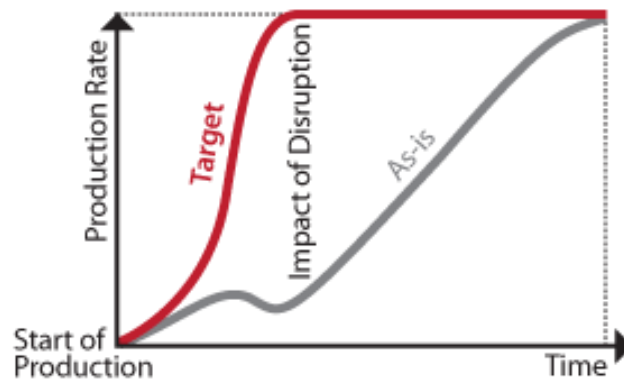


Figure 1: Ramp-up [1]

## 1.2. Needs in Manufacturing Management

*Given the problems in ramp-up and small lot production, an additional trend has to be taken into consideration in terms of:*

- *More variants and products in shorter sequence but also significant increase of automation, ICT controlled manufacturing systems and planning and control solutions*
- *Weak integration of engineering into production (horizontal) and enterprise ICT to shop floor automation and sensor level (vertical)[1]*

Consequently, future manufacturing needs to become a process of dynamic, event-sensitive operations, resource allocation, scheduling, optimization and controlling. Manufacturing management appears more and more to integrate and balance several factors, like quality, risks, costs and critical application time issues.



## 2 ARUM project

### 2.1. Overview

The ARUM project aims to improve planning and control systems for complex, small-lot products manufacturing, such as aircraft, aircraft interiors, and ships.

Producing aircraft is incomparable to producing large-quantity products such as cars or computers. A car manufacturer makes a relatively small investment in design and production ramp-up (the phase between designing a prototype and smooth production of the final product line), compared to the number of cars that are eventually sold. A company like Airbus faces a similar investment, even though it will only sell a small number of aircraft of the same design. Ships and aircraft are often produced in batches as small as 3 to 5, or are even completely customized. Thus, small-lot manufacturers need ICT solutions and tools to speed up their learning curve, allowing a smoother transit from design to production. They simply cannot learn from mistakes at the beginning of a series, but must get it right on the very first model.

Some of the issues that these tools need to deal with include:

- last-minute engineering changes
- new or changing requests by customers
- delayed delivery of parts
- little control over pre-assembly by partners

ARUM proposes to develop an intelligent Enterprise Service-Based platform (i-ESB). The platform will integrate a service-based architecture with a knowledge-based Multi-Agent System.

The i-ESB platform will gather information from sources such as sensors and resource management systems, giving decision makers and planners better insight into and control over the design-to-production process. Also, time-, cost- and risk-analysis will take place within the platform.

The project has a double approach, making use of both prediction (in the pre-planning phase) and real-time control (in the production phase). [1]

## 2.2. Airbus Assembly line

Assembly line of Airbus consists of multiple workstations. Workstation is a place, where certain tasks need to be completed before the whole aircraft section, for example fuselage, can be moved to the next one. The time, when all stations have completed their tasks so that all aircraft sections can be moved at the same time, is called cycle time.



Figure 2: Airbus assembly line

### Multi-agent platform

Each station in every cycle time is scheduled by an agent. As a result, the complex schedule of all stations is created. These schedule needs to be visualized. For this purpose, a new tool needed to be implemented.

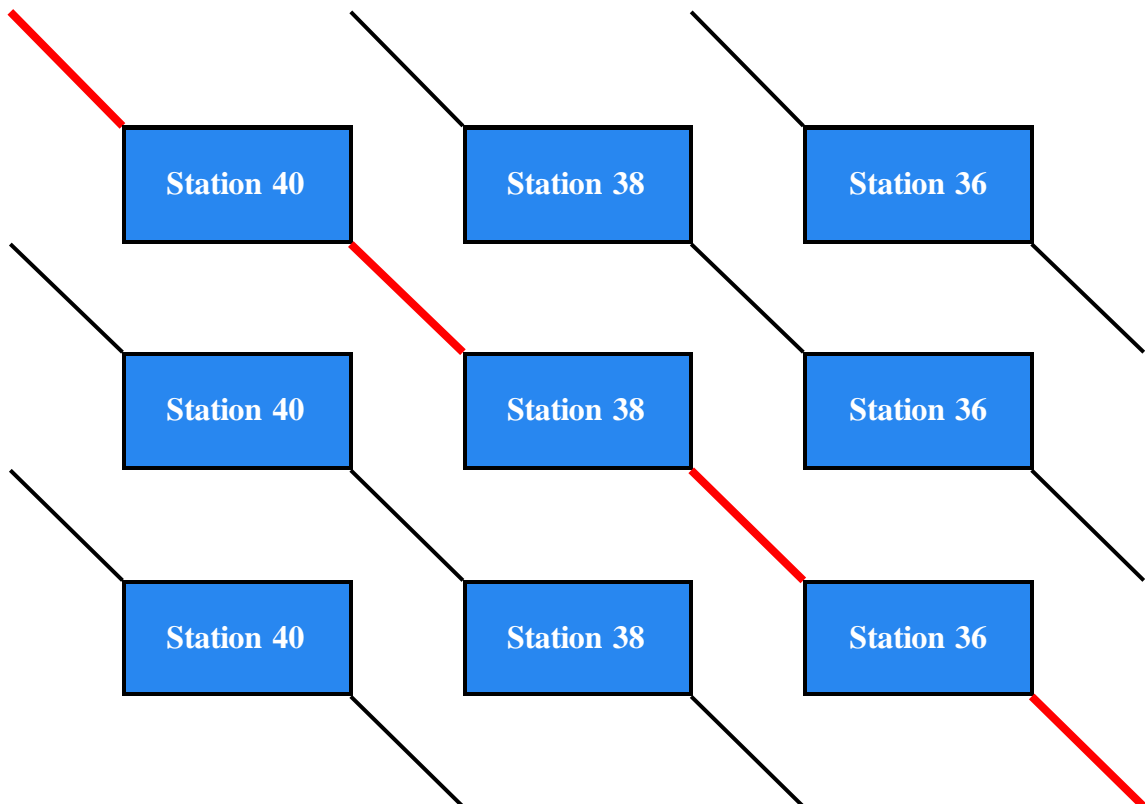


Figure 3: Multi-agent platform

## 2.2. Requirements of ARUM project

Develop a tool to visualize Gantt chart of one or multiple stations at a production line that meets the following requirements:

- Compatible with ARUM ontology - loads schedule from RDF/OWL
- Interactive chart - drag and drop tasks
- Highlight start and end of current station
- Show real or industrial time axis
- Zoom from minutes to days
- Different colors for different types and states of tasks
- Show task dependencies
- Show actual progress of each workorder and workstation in %-complete
- Compute and highlight critical path of schedule (over more stations)
- Add and remove tasks
- Edit task parameters
- Edit station parameters, for example available human resources and resources, their unavailability in time, skills, show unavailability chart
- Draw resources utilization chart
- Record changes made by user and send them to the agent platform if necessary
- Compare two different solutions (Gantt charts)

## 2.3. Gantt chart

### 2.3.1. Information

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity.

This allows you to see at a glance:

- What the various activities are
- When each activity begins and ends
- How long each activity is scheduled to last
- Where activities overlap with other activities, and by how much
- The start and end date of the whole project

To summarize, a Gantt chart shows you what has to be done (the activities) and when (the schedule). [5]

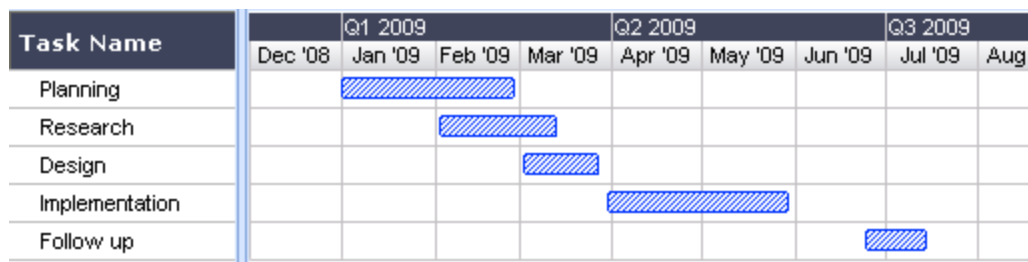


Figure 4: Gantt chart example [5]

### 2.3.2. History

The first Gantt chart was devised in the mid-1890s by Karol Adamiecki, a Polish engineer who ran a steelworks in southern Poland and had become interested in management ideas and techniques. Some 15 years after Adamiecki, Henry Gantt, an American engineer and management consultant, devised his own version of the chart and it was this that became widely known and popular in western countries.

Consequently it was Henry Gantt whose name was to become associated with charts of this type. [5]



*Figure 5: Henry Gantt and Karol Adamiecki [5]*

Originally Gantt charts were prepared laboriously by hand; each time a project changed, it was necessary to amend or redraw the chart and this limited their usefulness, continual change being a feature of most projects. Nowadays, however, with the advent of computers and project management software, Gantt charts can be created, updated and printed easily.

Today, Gantt charts are most commonly used for tracking project schedules. For this it is useful to be able to show additional information about the various tasks or phases of the project, for example how the tasks relate to each other, how far each task has progressed, what resources are being used for each task and so on. [5]

### 3 Used technology and algorithms

The first task was to find some free tool available on the internet to visualize Gantt charts. Although there are some open-source tools (for example SwiftGantt, GanttProject), which APIs might be extended by new features, it would take much more time to study these APIs than to develop own tool. Therefore, the decision to create a completely new tool from scratch to visualize Gantt chart was made using Java FX technology.

#### 3.1. Java FX – the rich client platform

Java FX is the next step in the evolution of Java as a rich client platform. It is designed to provide a lightweight, hardware-accelerated Java UI platform for enterprise business applications. Java FX applications are completely developed in Java. Java FX provides a rich set of UI controls, graphics and media API with high-performance hardware-accelerated graphics and media engines to simplify development of visual applications. It is designed to replace Swing UI library in near future. [4]

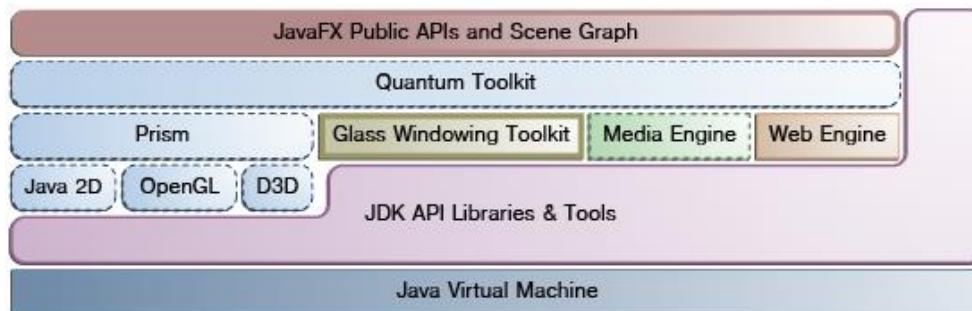


Figure 6: Java FX platform [4]

#### 3.2. Key features

- **Java APIs**  
Java FX is a Java library that consists of classes and interfaces that are written in native Java code.
- **FXML and Scene Builder**  
FXML is an XML-based declarative markup language for constructing a Java FX application user interface. Generated FXML markup can be ported to an IDE where a developer can add the business logic (MVC).
- **WebView**  
A web component that uses WebKitHTML technology to make it possible to embed web pages within a Java FX application.
- **Swing interoperability**  
Existing Swing applications can be updated with new Java FX features.

- **Built-in UI controls and CSS**

Components can be skinned with standard Web technologies such as CSS.

- **Canvas API**

The Canvas API enables drawing directly within an area of the Java FX scene that consists of one graphical element (Node).

- **Multi touch Support**

Java FX provides support for multi touch operations, based on the capabilities of the underlying platform.

- **Hardware-accelerated graphics pipeline**

Java FX graphics are based on the graphics rendering pipeline (Prism). It can use DirectX, OpenGL or Java2D technology based on system properties.

- **High-performance media engine [4]**

## 3.3. Algorithms

### 3.3.1. Critical Path Method (CPM)

#### Description

The Critical path method or Critical path analysis is a mathematically based algorithm for scheduling a set of project activities. It is an important tool for effective project management that is commonly used with all forms of projects, including construction, software development, research projects, product development, engineering, and plant maintenance, among others.

This method of scheduling can be applied to any project with interdependent activities. [9]

#### Background and history

Critical path method was developed in 1950s by the US Navy. It originally considered only logical dependencies between terminal elements. Since then, it has been expanded to allow the inclusion of resources related to each activity, through processes called activity-based resource assignments and resource leveling. [9]

#### Definitions

**Float (slack)** – amount of time that a task can be delayed without causing a delay to:

- subsequent tasks (free float)
- project completion date (total float)

**Critical path** is the sequence of activities which add up to the longest overall duration. It is the shortest time possible to complete the project. Any delay of an activity on the critical path directly impacts the planned project completion date (there is no float on the critical path). A project can have several, parallel, near critical paths. An additional parallel path through the network with the total durations shorter than the critical path is called a sub-critical or non-critical path.

**Critical activity** – activity with zero float.

**Resource leveling** – iterative process of assigning crews to activities in order to calculate their duration. [9]

#### Implementation

The essential technique for using CPM is to construct a model of a project that includes the following:

A list of all activities (tasks) required to complete the project (also known as Work Breakdown Structure)

- the time (duration) that each activity will take to completion
- the dependencies between the activities [10]



**CPM calculates:**

- the longest path of planned activities to the end of the project
- the earliest and latest that each activity can start and finish without making project longer
- determines “critical” activities (on the longest path).

To calculate the project’s overall duration, the CPM method uses two calculations:

- Forward-pass calculation
- Backward-pass calculation

In forward-pass, the early start and early finish dates are calculated. In backward-pass, the late start and late finish dates are calculated. Early start (ES) is the earliest date a task can start. Early finish (EF) is the earliest date a task can be completed. Late start (LS) is the latest date a task can start without delaying the project end date.

Late finish (LF) is the latest date a task can finish without delaying the project end date. [10]

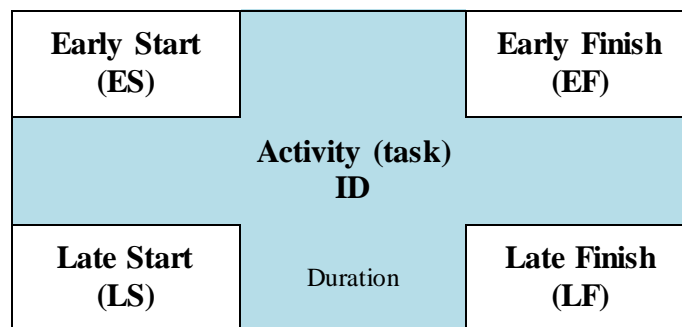


Figure 7: Activity (task) description [9]

**Calculation Example**

To understand how the forward and backward-pass analysis calculate early start, early finish, late start and late finish, let’s look at simple example: [10]

**Forward-pass**

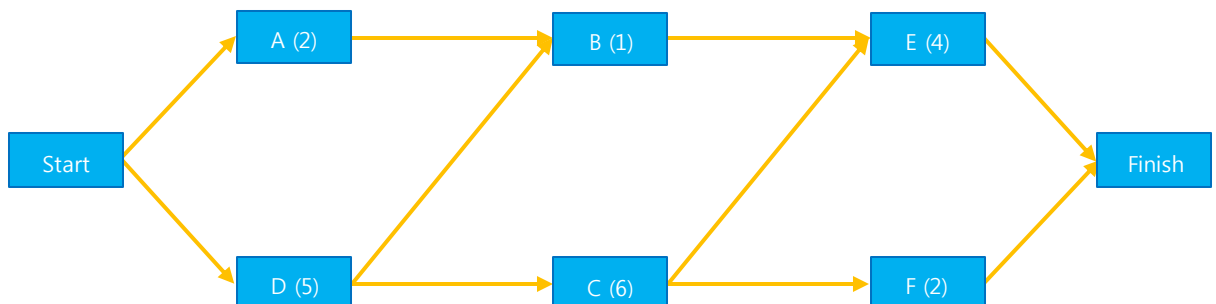


Figure 8: CMP example data [10]

By assuming zero dates for the Start milestone and zero dates for the start of activities A and D, what is the earliest time the activity A can be completed? It's 2 days, because A takes 2 days to complete. [10]

The same for activity D, the earliest time activity D can be completed is 5 days. What about activity B? Activities A and D are predecessors of activity B. It means that activity B cannot start until A and D are finished. The earliest time activities A and D can finish is 5 days. So the earliest time that activity B can start is 5 days. [10]

Applying the same method to all activities, we get the following values:

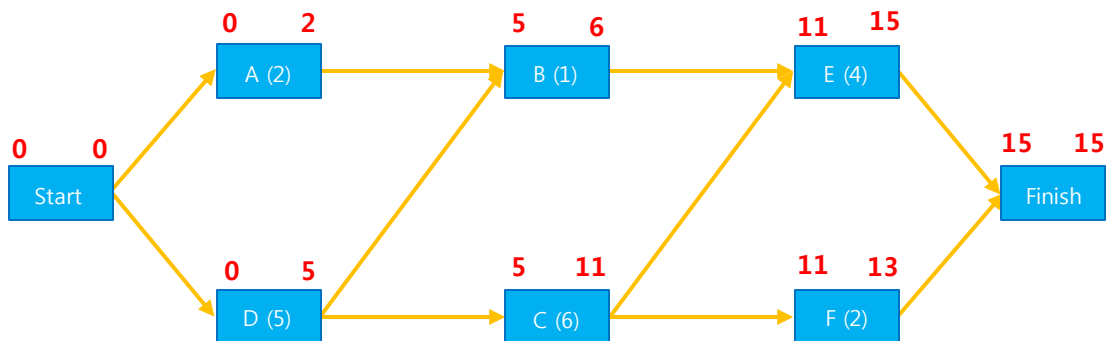


Figure 9: CMP forward-pass example [10]

As we have examined the forward-pass in the previous example, you may notice, that the early start (ES) of an activity is the latest related early date of all immediate predecessors. Another words, the early start of an activity is the maximum early start of its predecessors.

The early finish (EF) of an activity is its early start plus the duration. [10]

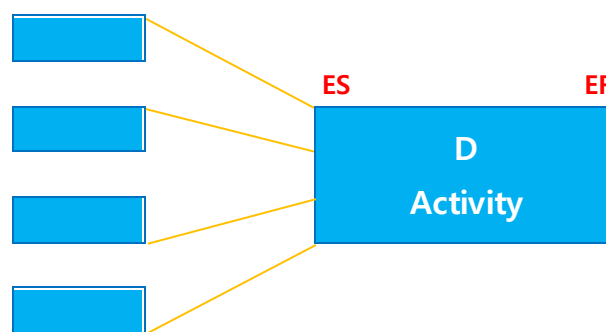


Figure 10: Compute early start and early finish of an activity [10]

ES = Immediate predecessors ESmax

EF = ES + D

## Backward-pass

For the backward-pass, we start from the back, or Finish milestone, change the direction of arrows and move backwards. In this example, the latest finish (LF) date for Finish milestone is 15 days. When we move backwards from the finish milestone, the latest finish date of the activity "F" will be 15 days. The LF date of activity E is also 15 days. The late start (LS) date of the activity E is its LF date minus the duration, so  $15-4=11$  days. What about activity C? In the backward-pass, the activity C is dependent on activities E and F. So, the late finish of activity C is 11 days. If the activity C finishes anytime later than 11 days, activity E will exceed 15 days which will delay the whole project. [10]

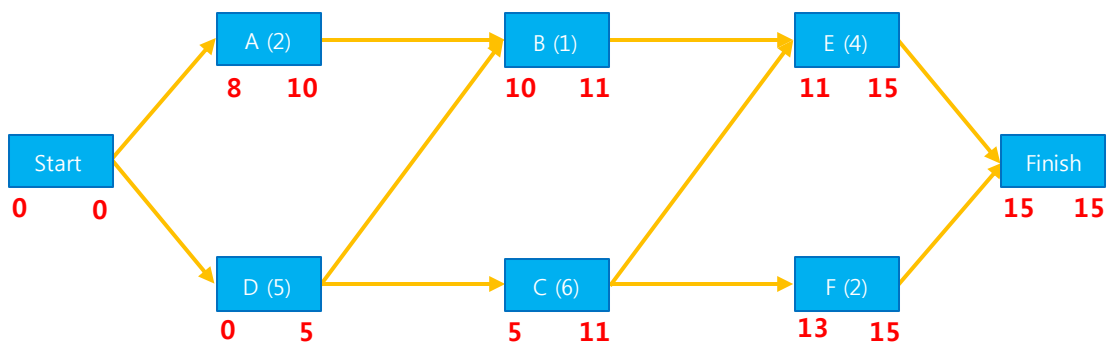


Figure 11: Backward-pass example [10]

On the backward-pass, the late finish of an activity is the minimum of all its related successors.

As we have examined the backward pass in the previous example, we've seen that the late finish of an activity is the earliest related early start of all its immediate successors. Another words, the late finish of an activity is the minimum late start of all its immediate successors.

The late start of an activity is its late finish minus the duration of activity.

$LF = \text{Immediate successors } LS_{\min}$

$LS = LF - D$

[10]

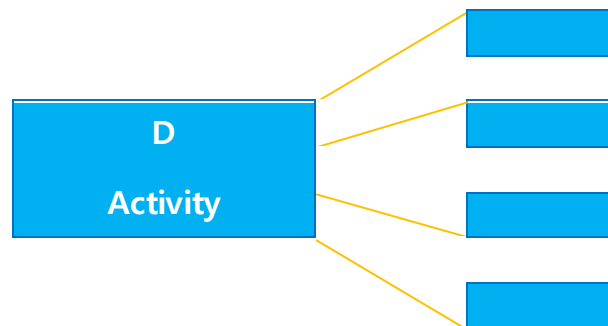


Figure 12: Compute late start and late finish of an activity [10]

## Computing Total Float

Total float of activity  $TF = LF - ES - D$

The activity with zero float is on critical path. [10]

In figure below, we can see the activities (tasks) on critical path (green color) are computed and visualized using the Tool.

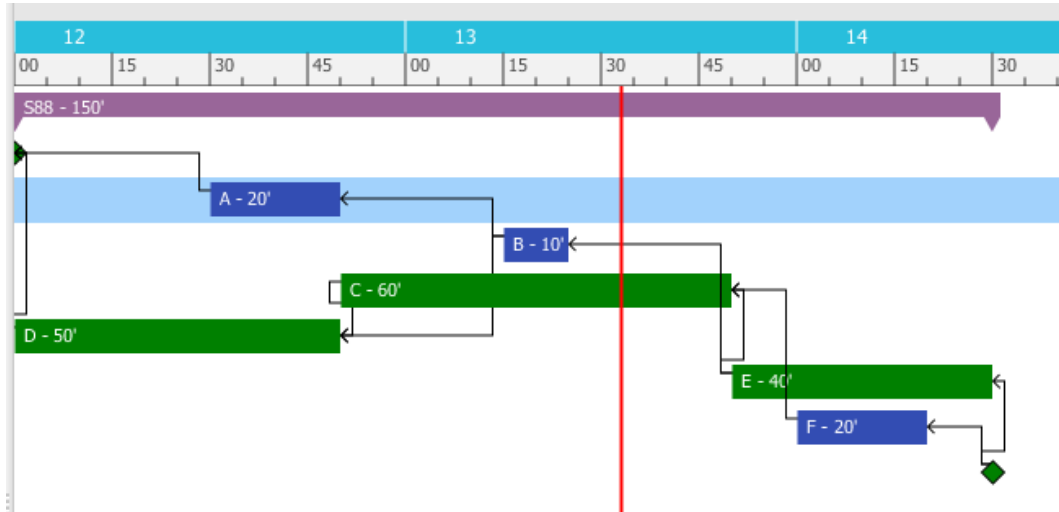


Figure 13: critical path computed and visualized in Gantt diagram

### 3.4. Implementation

#### 3.4.1. Data structure

The schedule loaded from the scheduling platform (RDF/XML) is based on the list of work stations.

Every station has some available resources, human resources and of course scheduled tasks. The tasks are wrapped into workorders, which means that tasks in one workorder have many common parameters, for example wing manufacturing may be one workorder with some task to do.

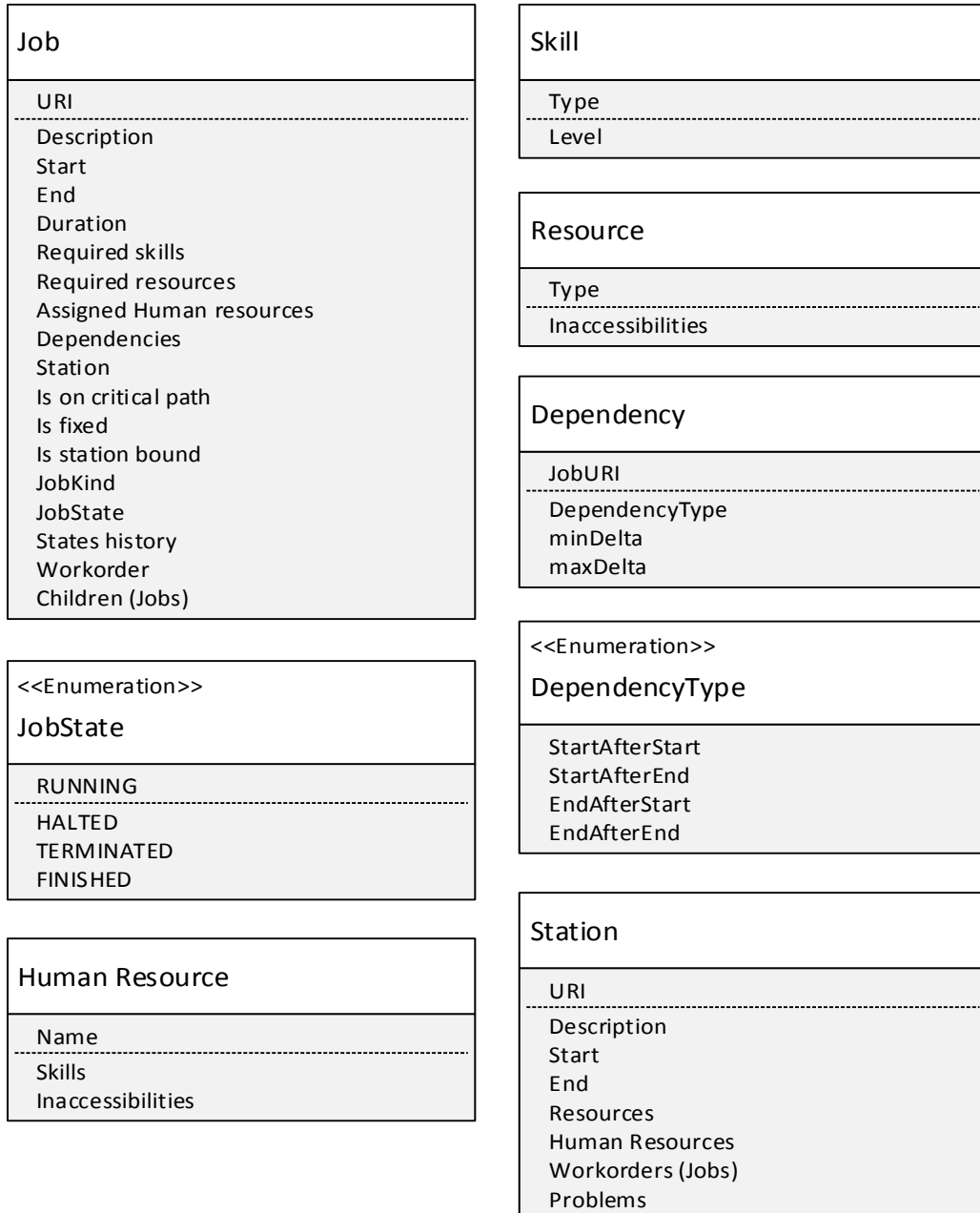


Figure 14: data structure

### 3.4.1. Dynamic data loading (lazy loading)

Because of many tasks in schedule (thousands), the table in main window cannot show all tasks at the same time. Therefore, dynamic data loading technique needed to be implemented.

Dynamic data loading loads data to the table from schedule list according to actual position of vertical scrolling bar. The amount of data that is loaded is based on actual size (number of rows) in table.

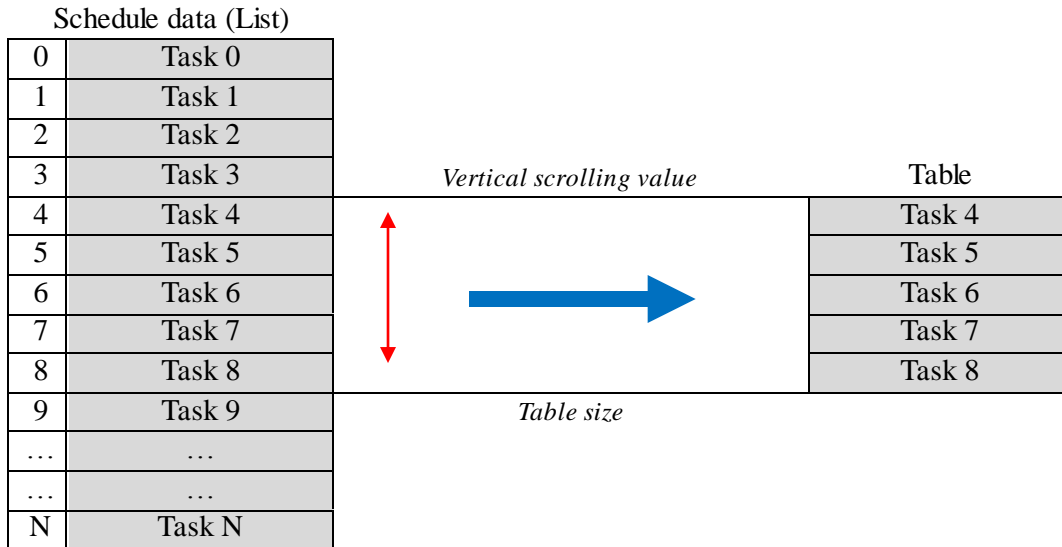


Figure 15: dynamic data loading

### 3.4.2. Events

Every change in data made by user (see User manual) is recorded and stored as event with datetime when the event occurred. This changes can be sent to the agent platform for further use.

## 4 User manual

The main window consists of two main parts: table with tasks on the left and actual Gantt chart with corresponding tasks visualized in time.

The main menu provides several features the user can use to edit the schedule and turn on/off some features, for example show background grid of chart.

Below the chart is zoom slider that allows the user to zoom chart from days to minutes.

The tool provides features to show, edit and send the changes made in chart. These changes can be made in different dialogs as explained later.

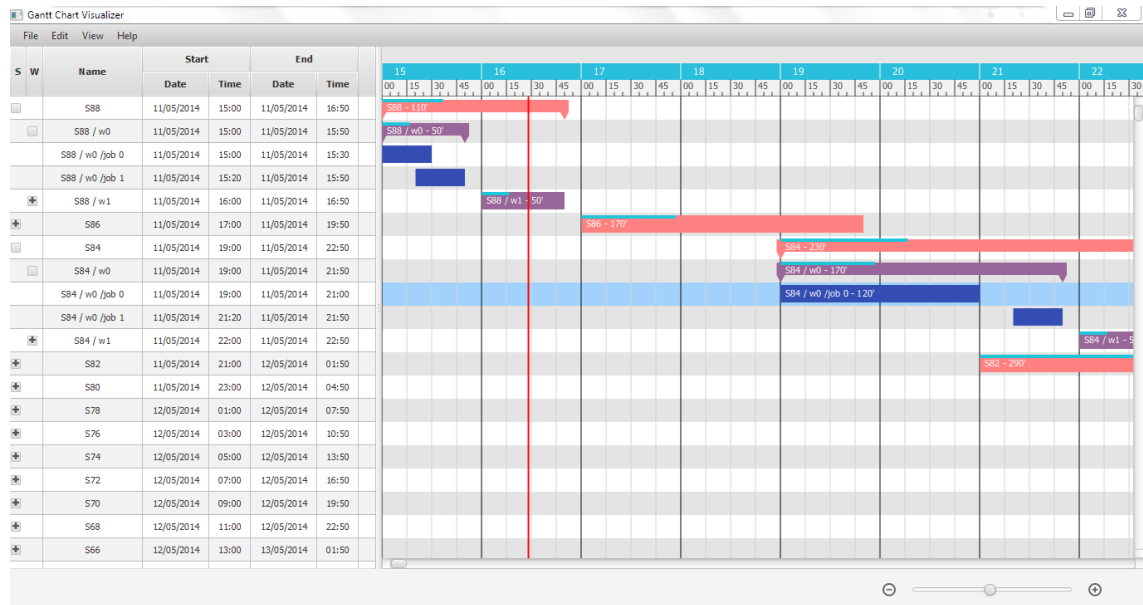


Figure 16: Main window

## 4.1. Features overview

### 4.1.1. Table

The table with tasks consists of several columns. The first two columns are implementing tree structure of schedule data, that is, a station (node) has some children (workorders), which also have children (tasks). This tree can be expanded to see the whole structure of the schedule by pressing check-box in a corresponding table row.

The next column “Name” shows actual name of task that is based on its unique ID and description.

The “Start” column consists of two nested columns showing date and time of the start of task. The “End” column also consists of two nested columns showing date and time of the end of task. The last column shows the current “State” of task, which can be: RUNNING, HALTED, TERMINATED, and FINISHED.

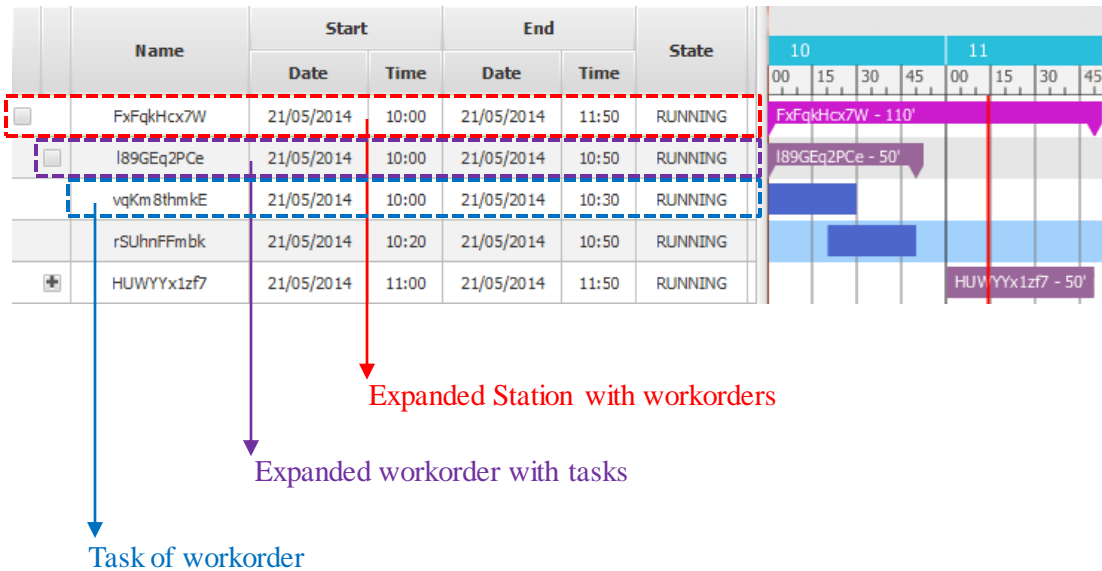


Figure 17: table description



### 4.1.2. Time axis

Time axis of the chart shows the time, in which each task is scheduled. The time axis is divided into 3 rows: the very top row shows the date and critical milestones, for example start and end of station, start and end of shift. The middle row shows hour of day and the last row show minute of hour. Based on the value of zoom, the time axis can show only days (middle and bottom rows are empty) or the smallest time interval, which is 5 minutes.

There are two types of time axis: real time and industrial time. Real time hour has 60 minutes. However, industrial minute is shorter than real, so industrial hour, which is as long as real hour, has 100 minutes. The middle row also changes when switched to industrial time. It show not the hour of day, but the hour of current shift, which is 8 hours long. The current type of time axis can be switched in main menu.

- Real time axis

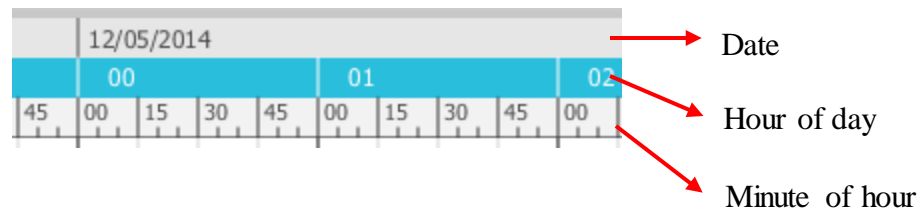


Figure 18: real time axis

- Industrial time axis

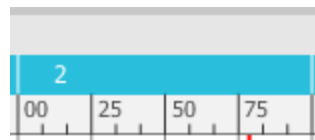


Figure 19: industrial time axis

### 4.1.3. Chart detail

As mentioned above, every station consists of workorders with scheduled tasks. This data model creates a tree that is also visualized in Gantt chart. When not expanded, each station is represented by purple rectangle. However, when the station is expanded, each of its workorders is also visualized in the chart. The same method can be used to expand workorder into its tasks.

Based on the current state of tasks, each workorder has a progress bar, which shows the actual completion. The station completion is also visualized by a progress bar that is based on the number of completed workorders.

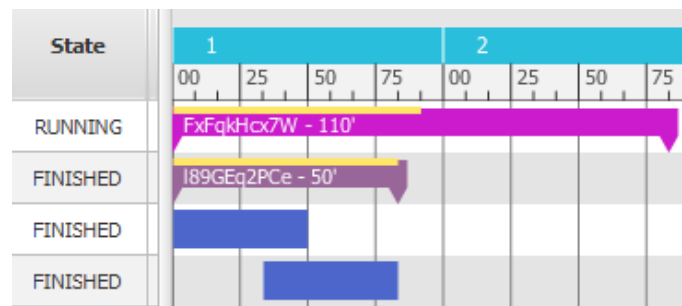


Figure 20: data model used in Gantt chart tool that creates a tree

#### 4.1.4. Resources utilization

Bellow the Gantt chart is placed resources utilization component, which consists of table with resources available at stations on the left and actual chart on the right side. This chart computes the total utilization of every resource, based on its amount, tasks that use it and their start and end time. The chart is recalculated and repainted every time a task changes its date.

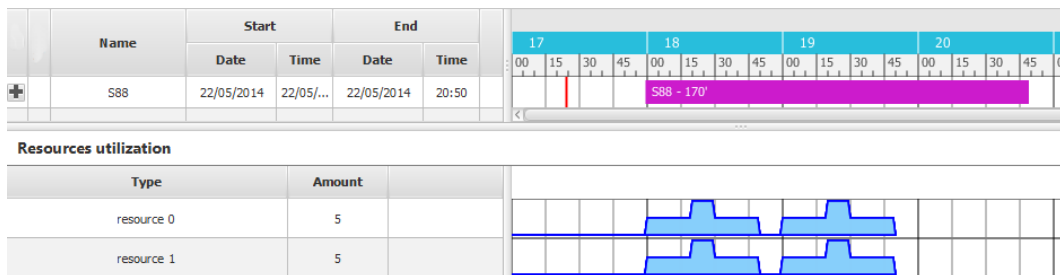


Figure 21: resources utilization chart

#### 4.1.5. Edit resources at selected station

To add, remove or edit resources and human resources that are available at the station, we need to open “Resources” dialog from the View-Resources menu. On the top of the dialog, click the choice menu (1) and select a station in which you want to edit resources.

#### Human resources at selected station with available skills

- Add new human resource

In Human resources tab (0), on the left side, enter the name of new worker into the text field (2) above the list of workers (4) and click the “Add” button (3). The new worker is added to the list below.

- Remove human resource

In Human resources tab, on the left side, select a worker in list of available workers (4) and press the “Remove” button (5).

- Add skill to human resource

On the right side of Human resources dialog, make sure that “Skills” tab (6) is visible.

On the left side of the dialog, in the list of workers (4), select a worker by left click to show its available skills in the skills table (7) to the right. Above the skills table, enter the type (8) and level (9) of new skill and press the “Add” button (10). The new skill is added to the table with available skills of selected worker.

- Remove skill from human resource

On the right side of Human resources dialog (0), make sure that “Skills” tab (6) is visible.

On the left side of the dialog, in the list of workers (4), select a worker to show its available skills in the skills table (7) to the right. Select a skill that needs to be removed and press the “Remove” button (11).

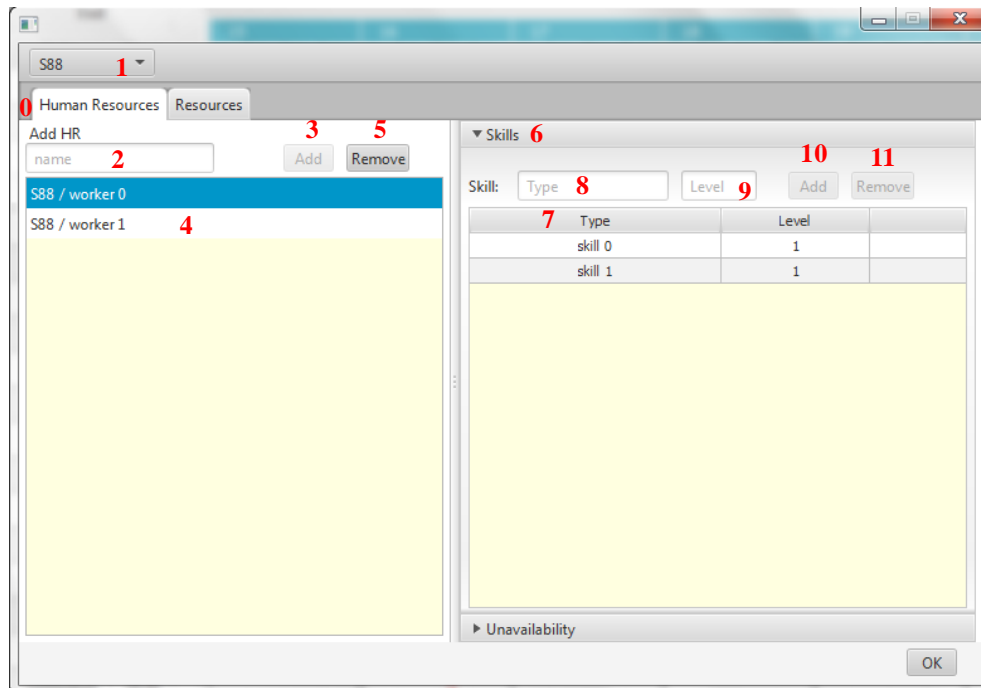


Figure 22: human resources and skills at selected station

## Human resources at selected station with unavailabilities in time

To add or remove unavailability (inaccessibility) in time of some worker at the station, make sure that Unavailability tab (2) is visible on the right side of the dialog.

- Add unavailability in time

To add unavailability in time to some worker, select a worker in the human resources list on the left side (1). Next, fill in the text fields with start and end of time interval, in which the worker won't be available at the station (3). Make sure that these times don't exceed start and end time of selected station. Click the "Add" button above the table (4).

- Remove unavailability in time

To remove unavailability in time of some worker, select a worker in the human resources list on the left side (1). Next, in the table with his unavailabilities (6), select the one that needs to be removed and press the "Remove" button above the table (5).

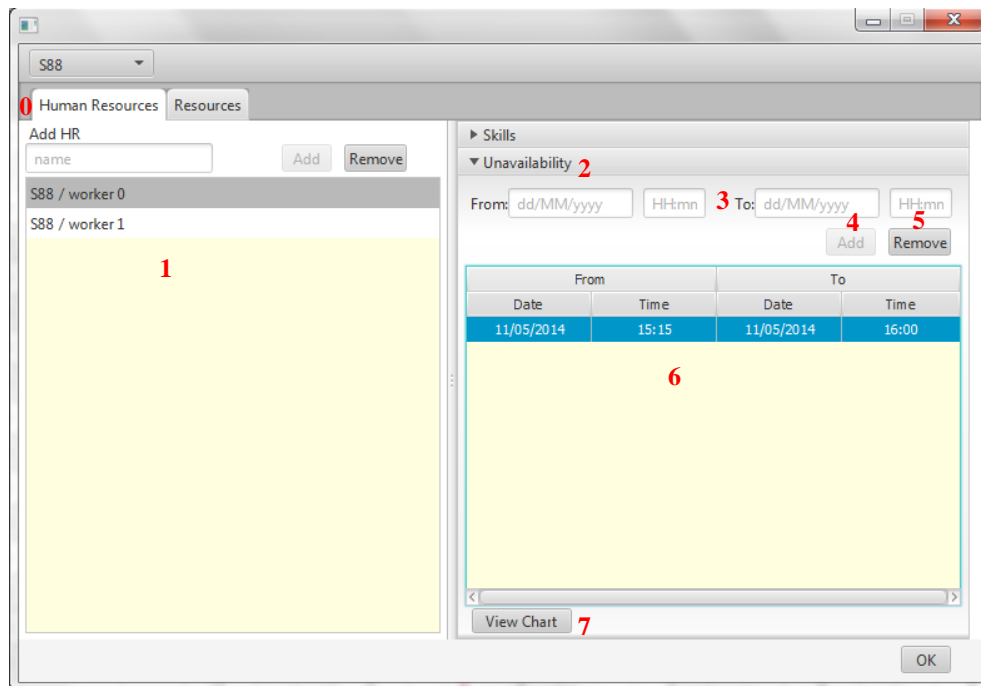


Figure 23: Add or remove unavailability of human resource

## Unavailability chart of selected Human resource

To visualize time intervals, in which the selected worker is available at the station, select a worker in the human resources list (4) and press the “View chart” button (7) below the Unavailabilities table.

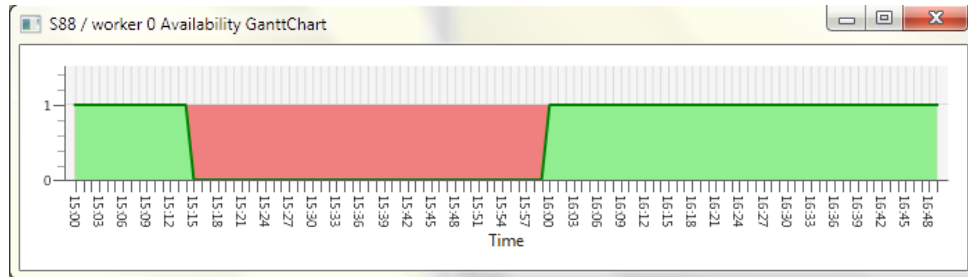


Figure 24: Unavailability chart of human resource

## Resources at selected station with unavailability in time

To add or remove available resources at selected station, the same principle applies as by Human resources.

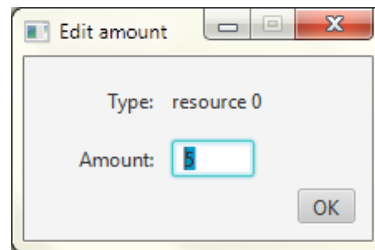
- Add new resource

In resources tab (1), on the left side, enter the type of new resource into the text field (2) above the list of workers (4) and the amount of resource into the text field next to it (3) and click the “Add” button (5). The new resource is added to the list below.

- Edit resource

To edit the amount of a resource available at the station, select a resource by left-click in resources table and press “Edit” button (6) above.

New dialog is open to input new amount value.



*Figure 25: Edit amount of resource*

- Remove resource

In Resources tab, on the left side, select a resource in table of available resources (4) and press the “Remove” button (7).

- Add unavailability in time

To add unavailability in time to some resource, select a resource in the resources table on the left side (4). Next, fill in the text fields with start and end of time interval, in which the resource won't be available at the station and also the amount of unavailable resources (8). Make sure that these times don't exceed start and end time of selected station. Click the “Add” button above the table (9).

- Remove unavailability in time

To remove unavailability in time of some resource, select a resource in the resources table on the left side (4). Next, in the table with unavailabilities (11), select the one that needs to be removed and press the “Remove” button above the table (10).

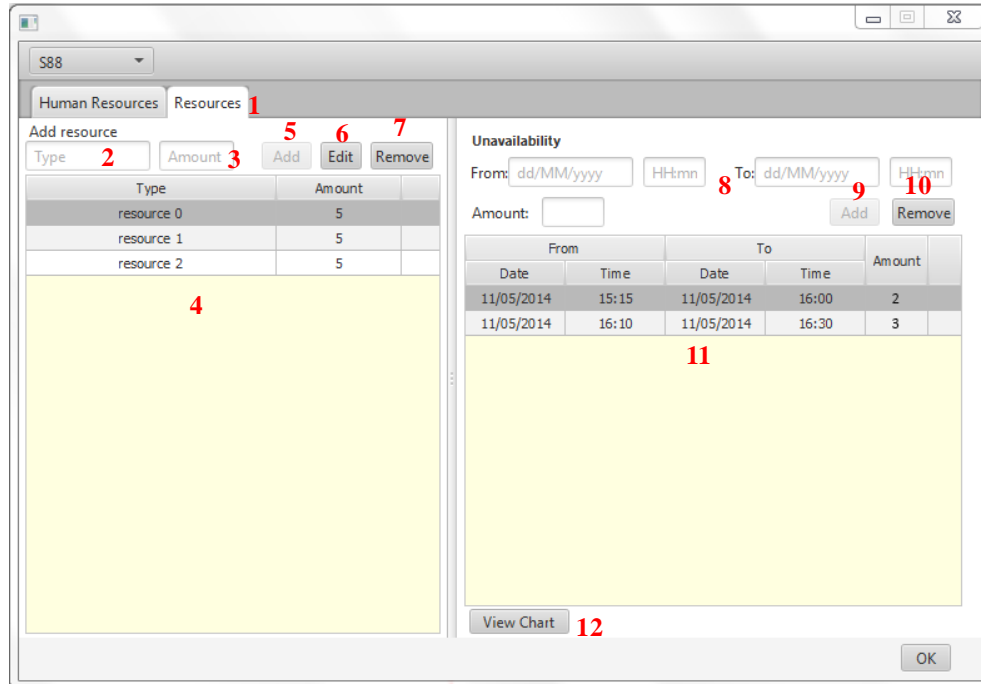


Figure 26: add, edit and remove resources at station

### Unavailability chart of selected resource

To visualize time intervals, in which the selected resource is available at the station in certain amount, select a resource in the resources table (4) and press the “View chart” button (12) below the Unavailabilities table.

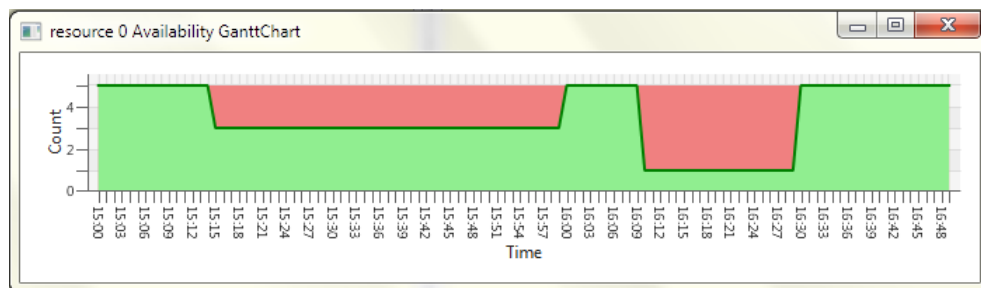


Figure 27: Unavailability chart of selected resource



#### 4.1.6. Add new Job

##### Main parameters

To add new Job to the schedule, select Edit-Add new Job menu.

A new dialog to input parameters of new Job is created:

- Info tab

In info tab, input all the Job parameters in the table, starting from the station and workorder in which the Job should be done.

New Job	
Info   Required Skills   Resources   Dependencies	
Station	<input type="text"/>
Workorder	<input type="text"/>
ID	<input type="text"/>
Description	<input type="text"/>
Planned Start	<input type="text" value="dd/MM/yyyy"/> Time <input type="text" value="HH:mm"/>
Planned End	<input type="text" value="dd/MM/yyyy"/> Time <input type="text" value="HH:mm"/>
Kind	<input type="text"/>
State	Label <input type="button" value="Change"/>
Fixed	<input type="checkbox"/>
Station bound	<input type="checkbox"/>
Is on critical path	<input type="checkbox"/>

Figure 28: New Job parameters

## Required Skills

To add or remove required skills to or from a new Job (if not done by the scheduler), open the “Required skills” tab.

- Add required skill

Enter the type, level and amount into text fields above the table, and press “Add” button. The new skill will be inserted into the table.

- Remove required skill

Select the skill that should be removed by left-click in the table with skills and press “Remove” button.

The screenshot shows a window titled "New..." with a tabbed interface. The "Required Skills" tab is selected. The interface includes three input fields for "Type:", "Level:", and "Amount:" (with "1" entered). There are "Add" and "Remove" buttons next to the "Amount" field. Below these is a table with columns "Type", "Level", and "Amount". The table is empty, displaying "No content in table". "OK" and "Cancel" buttons are at the bottom right.

*Figure 29: new Job required skills*

### Required resources

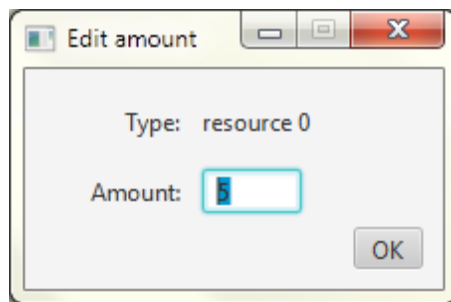
To add or remove required resources to or from a new Job (if not done by the scheduler), open the “Resources” tab.

- Add required resource

Enter the type and amount into text fields above the table, and press “Add” button. The new resource will be inserted into the table.

- Edit required resource amount

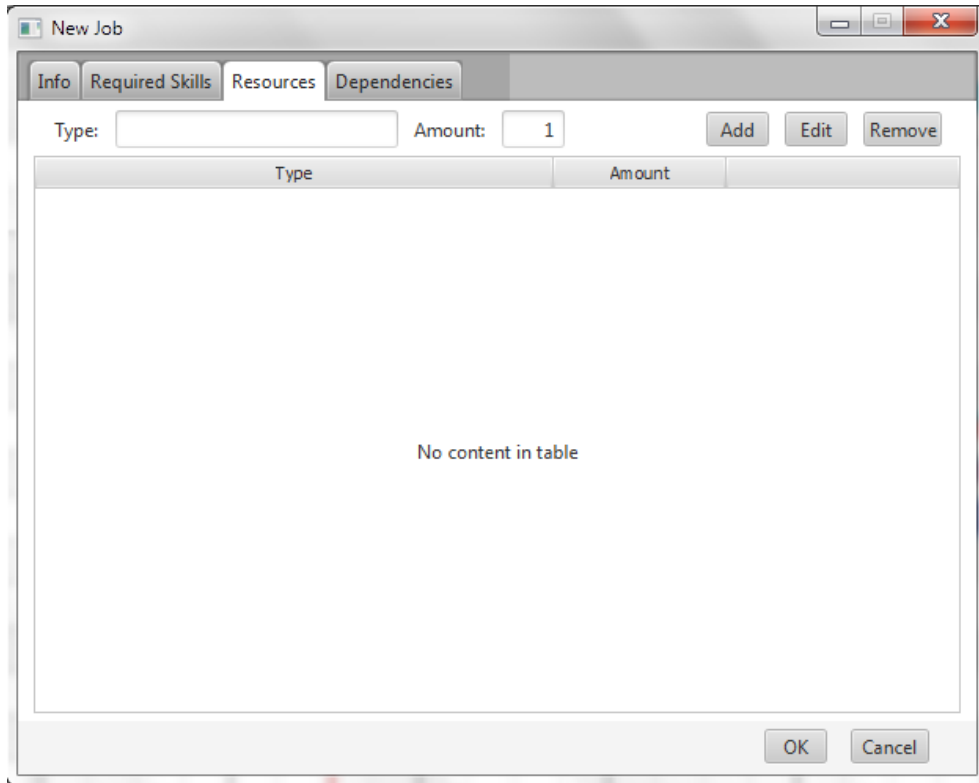
To edit amount of required resource, select the resource in table by left-click and press “Edit” button. New dialog is created to enter new amount value.



*Figure 30: Edit required resource amount*

- Remove required resource

Select the resource that should be removed by left-click in the table with resources and press “Remove” button.



*Figure 31: new Job resources*

## Dependencies

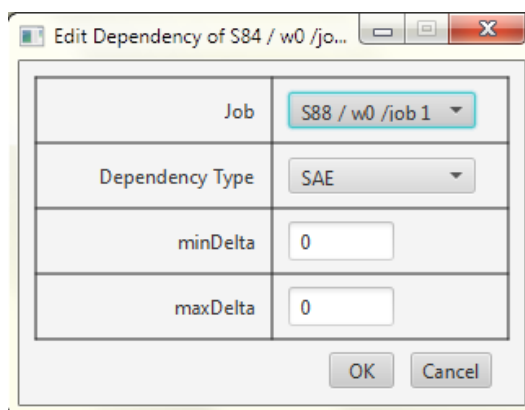
To add or remove dependencies to or from a new Job (if not done by the scheduler), open the “Dependencies” tab.

- Add dependency

Select the related Job from the choice box above the table and also the type of Dependency, and press “Add” button. The new dependency will be inserted into the table.

- Edit dependency

To edit min and maxDelta of dependency, select the dependency in table by left-click and press “Edit” button. New dialog is created to enter new delta values.



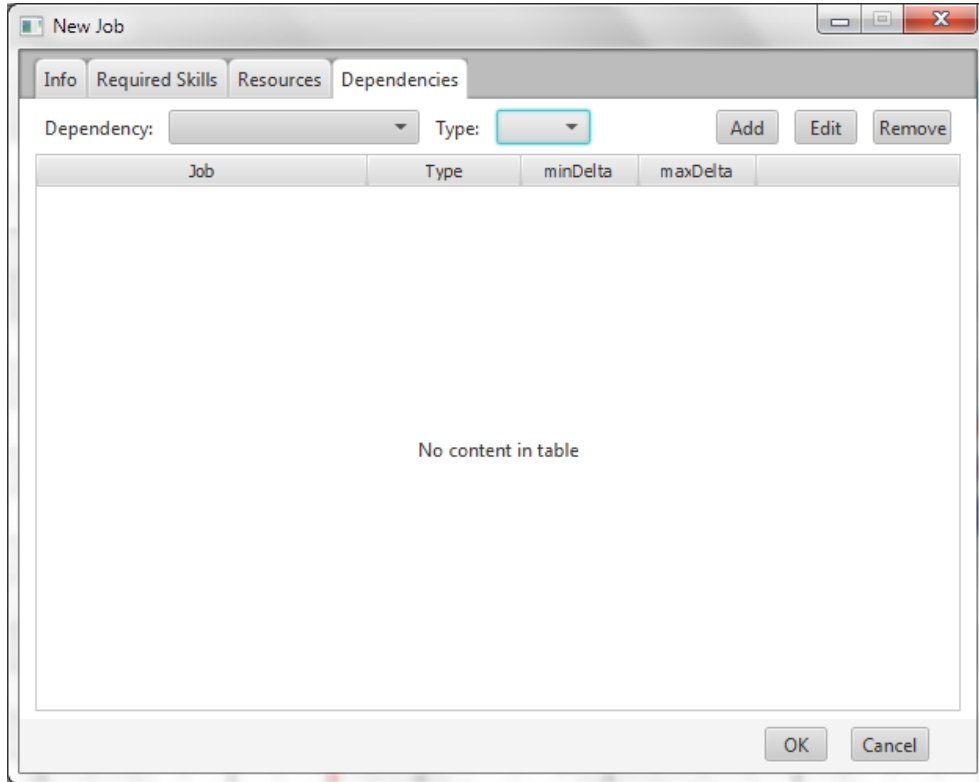
Job	S88 / w0 /job 1
Dependency Type	SAE
minDelta	0
maxDelta	0

OK Cancel

Figure 32: Edit delta parameters of a dependency

- Remove dependency

Select the dependency that should be removed by left-click in the table with dependencies and press “Remove” button.



*Figure 33: new Job dependencies*

#### 4.1.7. Edit selected Job

##### Main parameters

To edit selected Job in schedule, select the Job that you want to edit in main table and press Edit- Edit selected Job menu.

A new dialog to edit parameters of Job is created:

- Info tab  
See 4.1.5 Info tab

##### Human resources

- Assign human resources to required skills

To manually assign workers to the Job, open the “Human resources” tab.

On the left side, you can see a table with available workers and their skills. On the right side is table of required skills. Select a worker and also select required skill you want to assign HR to by left-click, and press “→” button. You can see assigned workers in “Assigned HR” table.

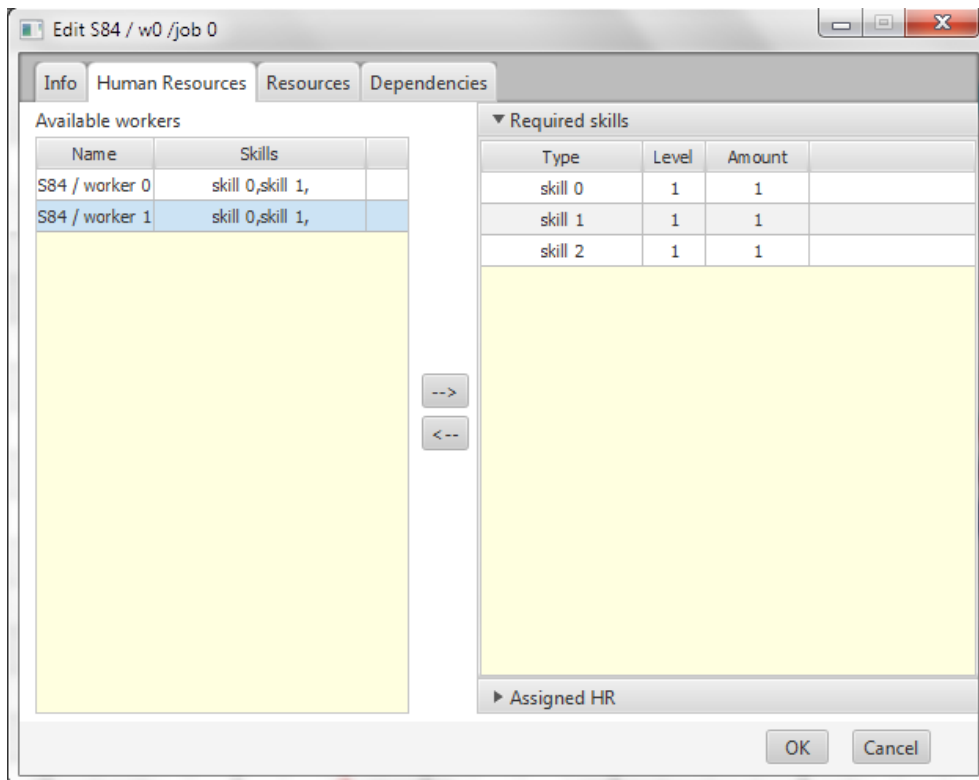
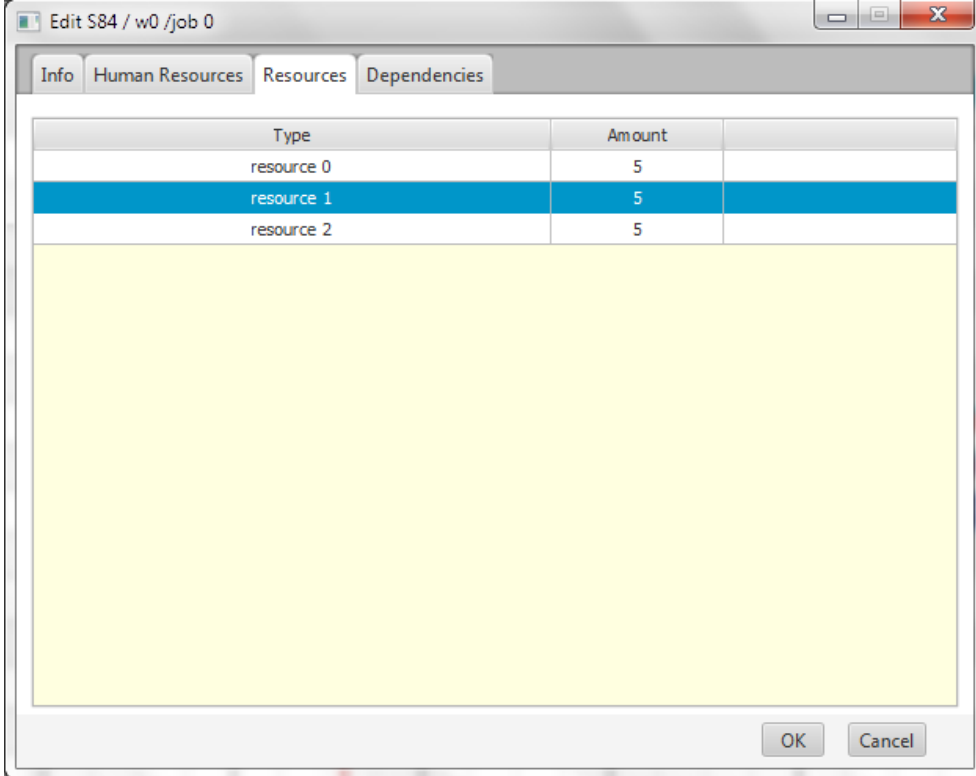


Figure 34: Assign human resources to a Job

## Required resources

To view required resources of selected Job, open “Resources” tab.



Type	Amount
resource 0	5
resource 1	5
resource 2	5

*Figure 35: View required resources of a Job*



#### 4.1.8. Tool settings

To edit color setting of the tool, open View- Settings dialog. You need to restart the tool to apply changes.

- Task color

Edit colors of different types of tasks (Jobs), or load default values by pressing “Load defaults” button.

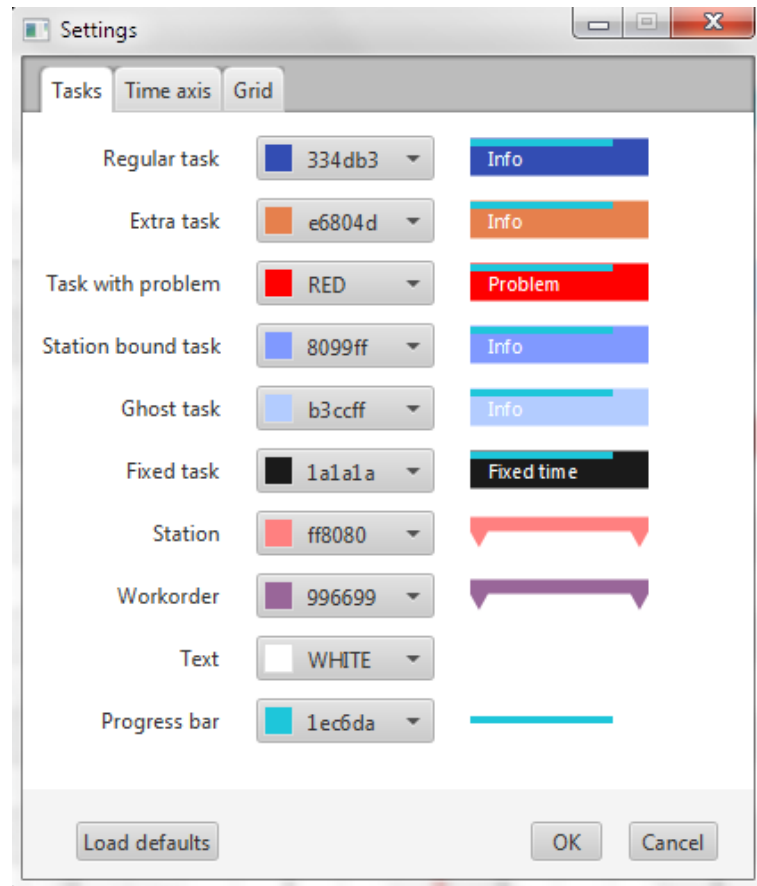


Figure 36: Color settings

- Time axis colors

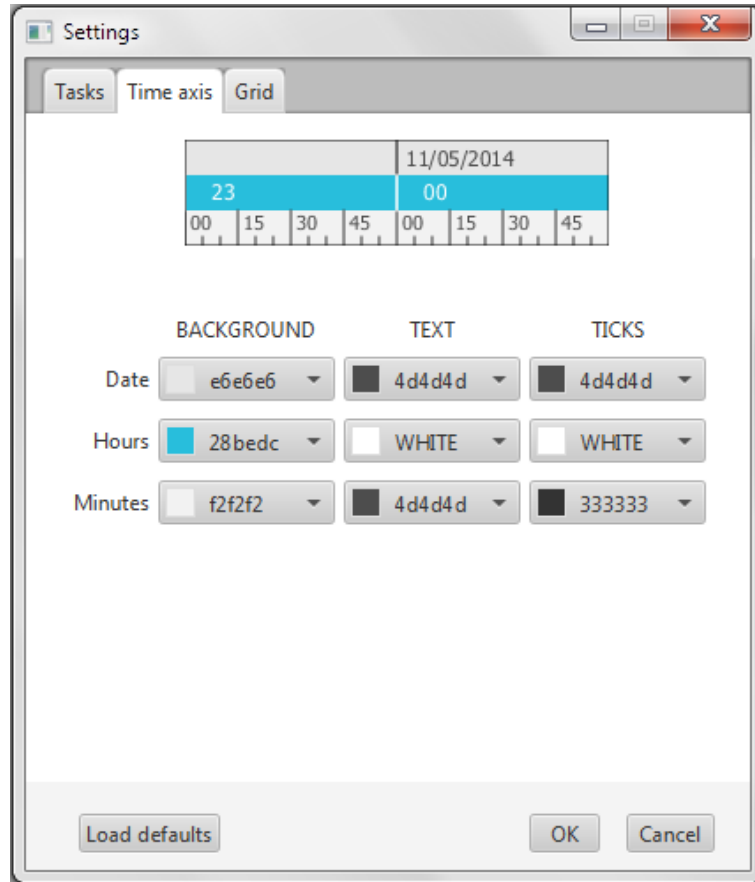
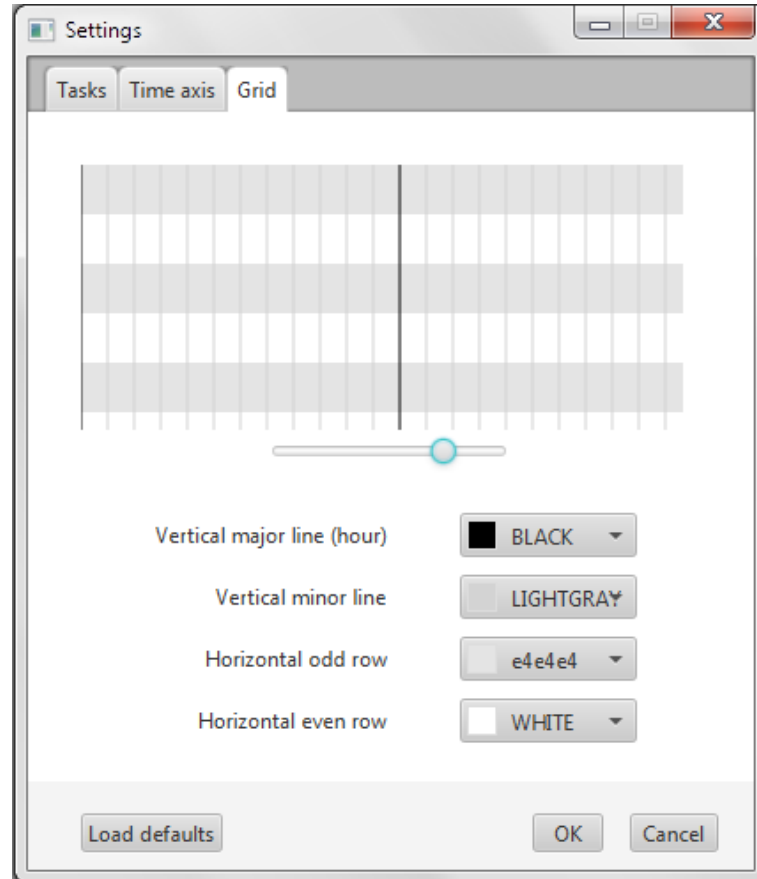


Figure 37: time axis colors

- Chart grid colors



*Figure 38: chart grid colors*

#### 4.1.9. Station problems

To view the problems that occurred by station scheduling, go to View-problems menu.

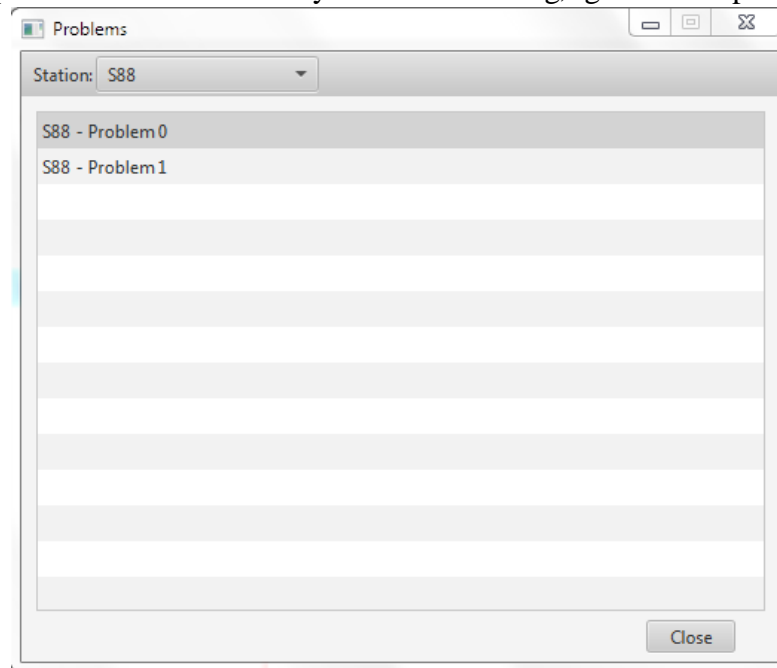


Figure 39: station problems

#### 4.1.10. Commit changes dialog

To send changes (events) made in the schedule by user, go to File-commit changes menu. New dialog is created that shows list of events with their description, as in Figure 40:

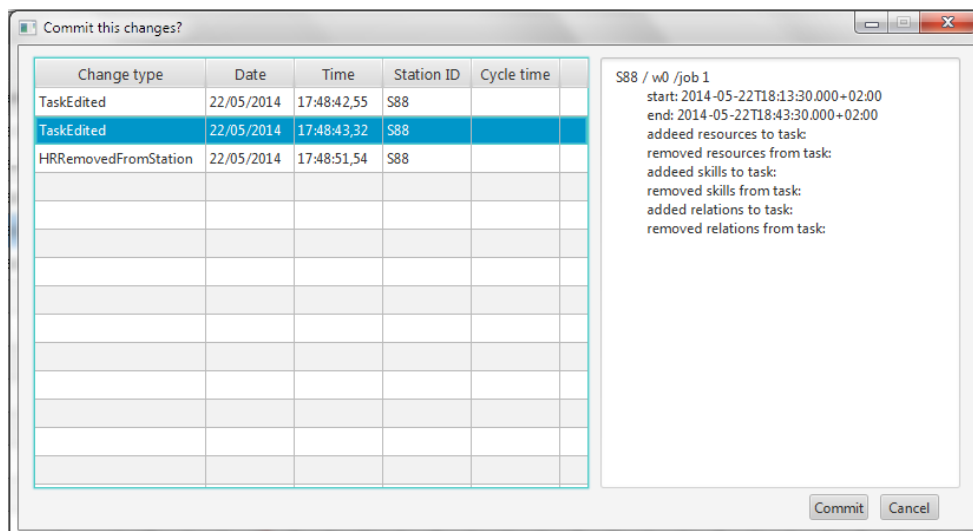


Figure 40: commit changes dialog

## Conclusion

The tool provides the required functionality as described in ARUM requirements (2.2). It can be used to visualize results of scheduling algorithms of ARUM project.

### **Future work**

The tool must be more robust and also the performance must be significantly increased by using worker threads. The resources utilization chart need to be implemented to show resources of multiple station in schedule, not only one.

Finally, the whole project will be implemented as Netbeans Platform module, which will be used in more complex UI application.

## Figure list

- Figure 1: Ramp-up [1] (page 9)
- Figure 2: Airbus assembly line (page 11)
- Figure 3: Multi-agent platform (page 11)
- Figure 4: Gantt chart example [5] (page 13)
- Figure 5: Henry Gantt and Karol Adamiecki [5] (page 14)
- Figure 6: Java FX platform [4] (page 15)
- Figure 7: Activity (task) description [9] (page 18)
- Figure 8: CMP example data [10] (page 18)
- Figure 9: CMP forward-pass example [10] (page 19)
- Figure 10: Compute early start and early finish of an activity [10] (page 19)
- Figure 11: Backward-pass example [10] (page 20)
- Figure 12: Compute late start and late finish of an activity [10] (page 20)
- Figure 13: critical path computed and visualized in Gantt diagram (page 21)
- Figure 14: data structure (page 22)
- Figure 15: dynamic data loading (page 23)
- Figure 16: Main window (page 24)
- Figure 17: table description (page 25)
- Figure 18: real time axis (page 26)
- Figure 19: industrial time axis (page 26)
- Figure 20: data model used in Gantt chart tool that creates a tree (page 27)
- Figure 21: resources utilization chart (page 28)
- Figure 22: human resources and skills at selected station (page 29)
- Figure 23: Add or remove unavailability of human resource (page 30)
- Figure 24: Unavailability chart of human resource (page 31)
- Figure 25: Edit amount of resource (page 32)
- Figure 26: add, edit and remove resources at station (page 33)
- Figure 27: Unavailability chart of selected resource (page 33)
- Figure 28: New Job parameters (page 34)
- Figure 29: new Job required skills (page 35)
- Figure 30: Edit required resource amount (page 36)
- Figure 31: new Job resources (page 37)
- Figure 32: Edit delta parameters of a dependency (page 38)
- Figure 33: new Job dependencies (page 39)
- Figure 34: Assign human resources to a Job (page 40)
- Figure 35: View required resources of a Job (page 41)
- Figure 36: Color settings (page 42)
- Figure 37: time axis colors (page 43)
- Figure 38: chart grid colors (page 44)
- Figure 39: station problems (page 45)
- Figure 40: commit changes dialog (page 45)



## References

- [1] <http://arum-project.eu/index.php/en/overview/context>
- [2] <http://arum-project.eu/index.php/en/overview>
- [3] <http://www.oracle.com/technetwork/java/javase/overview/javafx-overview-2158620.html>
- [4] <http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#JFXST784>
- [5] <http://www.gantt.com/>
- [6] Pro JavaFX 2: A Definitive Guide to Rich Clients with Java Technology, By James Weaver, Weiqi Gao, Stephen Chin, Dean Iverson, Johan Vos, ISBN13: 978-1-4302-6872-7, 640 Pages, User Level: Intermediate, Publication Date: March 20, 2012, Apress
- [7] JavaFX 2.0: Introduction by Example, By Carl Dea, ISBN13: 978-1-4302-4257-4, 200 Pages, User Level: Intermediate to Advanced Publication Date: December 20, 2011, Apress
- [8] <http://code.makery.ch/java/javafx-2-tutorial-intro/>
- [9] CPM Method, CEE 320 – VDC SEMINAR, Jesse Santiago & Desirae Magallon, 4 February 2009, <http://www.stanford.edu/class/cee320/CEE320B/CPM.pdf>
- [10] CPM tutorial video by Shohreh Ghorbani, <http://www.projectcontrolacademy.com/>  
<https://www.youtube.com/watch?v=SF53ZZsP4ik>