Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Computer Graphics and Interaction

DIPLOMA THESIS ASSIGNMENT

Student: **Bc. Ondřej Harcuba**

Study programme: Open Informatics
Specialisation: Software Engineering

Title of Diploma Thesis: **Tool for operational monitoring of production processes**

Guidelines:

The goal of the thesis is to design and implement a managerial tool for configuration, monitoring, and visualization of production stations and processes. The tool consists of the following modules: (i) design of the factory layout consisting of work stations; (ii) configuration of work station details (assignment of human resources, machines, tools, and materials); (iii) specification and visualizations of production plans (sequences of operations, required resources, durations of operations, and technological dependencies); and (iv) visualization of production progress (finished operations, status of actually running operations, error states).
The tool has to be integrated into the platform developed within the European project ARUM. The documentation has to contain the functional requirements analysis, description of software architecture, and the user manual.

Bibliography/Sources:

T. Allweyer, BPMN 2.0. Books on Demand GmbH, 2010
A. Shalloway, J. R. Trott, Design Patterns Explained: A New Perspective on Object-Oriented Design (2nd Edition). Addison Wesley, 2005

Diploma Thesis Supervisor: Ing. Pavel Vrba, Ph.D.

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

L.S.

prof. Ing. Jiří Žára, CSc.
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, February 21, 2014

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Science and Engineering

Master's Thesis

Tool for operational monitoring of production processes

Bc. Ondřej Harcuba

Supervisor: Ing. Pavel Vrba, Ph.D.

Study Programme: Open Informatics

Field of Study: Software Engineering

May 11, 2014

Acknowledgements

I would like to express my gratitude to my supervisor Ing. Pavel Vrba, Ph.D. for providing me the theme of my master thesis as well as for his valuable advices, comments and time spent on this project. I would also like to thank to all my colleagues from the Certicon a.s. for all their support, namely to my colleague Luis Moreno for his constructive criticism and ideas.

My greatest thanks go to my family and friends for their infinite support and care during my study.

Declaration

I hereby declare that I have completed this thesis independently and that I have listed all the literature and publications used.
I have no objection to usage of this work in compliance with the act §60 Zákon č. 121/2000Sb. (copyright law), and with the rights connected with the copyright act including the changes in the act.

In Prague on May 10, 2014 ....

Abstract

The aim of this thesis is design and implementation of a management tool for the operational monitoring of manufacturing processes. The tool will allow visualization and creation of manufacturing processes composed of sequences of manufacturing operations, the necessary materials, workers and other resources. Subsequently it will allow to the user to monitor the progress of production and offer the means to perform impact analysis relating to external fault events (e.g., late delivery of materials, lack of resources, poor quality of components, etc.). The tool will be integrated into the platform defined in the European project ARUM.

Abstrakt

Cílem této diplomové práce je návrh a implementace manažerského nástroje pro operativní monitoring výrobních procesů. Nástroj bude umožňovat vizualizaci a vytváření výrobních procesů skládající se z posloupnosti výrobních operací, potřebných materiálů, pracovníků a jiných potřebných zdrojů. Následně bude umožňovat uživateli sledovaní průběhu výroby a nabízet prostředky pro provádění dopadové analýzy týkající se externích poruchových událostí (např. zpožděná dodávka materiálu, chybějící zdroje, nevyhovující kvalita součástky, apod.). Nástroj bude integrován do platformy definované v rámci evropského projektu ARUM.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Aim

The aim of this thesis is to design and implement managerial tool for operative monitoring of manufacturing processes. This tool will be divided to the several modules, each of them will provide specialized set of functionalities for the selected user roles defined in the project ARUM. This thesis will mainly focus on describing, designing and implementing of the two modules of this tool, the Designer module and the Production module. First module will provide tools for specification of the production plan, consisting of sequences of manufacturing operations, the necessary materials, workers and other necessary resources. Second module will allow to the user monitoring of the production progress and it will provide tools for operative solving of external faults events.

A secondary goal of this thesis is to give the reader introduction to the ARUM project and briefly describe developed platform architecture.

1.2 Motivation

Small lot production such as aircrafts, ships or even galley inserts significantly differ from standard production scenarios. It is characterised by a high degree of customisation of the individual products, highly distributed supply chains, with a large number of suppliers spread across countries and even continents. Large aircraft can be built from up to 6 million parts, which in combination with mentioned distributed supply chains and possible designs faults creates potential uncertainty of deliveries during the stages of ramp-up and production, which can result in significant financial losses. As examples of such production disturbances during the ramp-up stage can be the Airbus A380 production delay in 2009 caused by design failures or grounding of 50 operating Boeing 787 in January 2013 because of fire hazards from new Lithium-Ion batteries.[11]

Thus, small lot productions bring challenges to innovative technologies and ICT systems, such as the need for adaptive and robust information system, based on the distribution and decentralisation of control and decision support.[13]

Developing of such ICT solution is the main objective for the ARUM project. Managerial tool, presented in this thesis, will be integrated into it, serving as front-end application for the users operating the system.

1.3 Thesis outline

The thesis is structured as follows.

The first part of the chapter Analysis[2] provides a general overview about the ARUM project, what kind of challenges it tries to deal with and what ICT solutions it uses and develops. The second part of this chapter briefly describes ARUM project, production process of the involved industries, and what are the most troublesome issues the small lot productions has to deal with. The last part of the chapter contains ARUM solution architecture overview and requirements for the managerial tool, which design and implementation are the aims of this thesis.

Chapter Design[3] presents user interface designs of the developed tool that should fulfill presented functional requirements. First part presents design of the user interface of the Designer module and the second part presents the Production module design.

Chapter Implementation[4] describes the rich client platform (RCP) used for the tool implementation, the tool architecture and its integration within the ARUM platform. The last part of the chapter describes what part of the proposed solution has been implemented for the purpose of this thesis and its implementation details.

Chapter Testing[5] includes results of the first and second tests of the low-fidelity prototype with users and partners.

The last chapter Conclusion[6] then provides thesis summarization, evaluates the fulfillment of the objectives and presents ideas for further extensions and improvements of the developed tool.

Chapter 2

Analysis

2.1 ARUM project

This section describes participants, aim and solution approach of the ARUM (Adaptive Production Management) project.

2.1.1 Project participants

ARUM is a collaborative research project within the European Unions 7th Framework Programme[4] and part of the European Research Initiative "Factory of the future."[3] ARUM started in September 2012 and will last for 37 moths involving 14 partners from seven EU countries and one from Russia. The participants are:

- **Technology providers**
  - TIE
  - Certicon
  - Smart Solutions
  - Almende
  - P3 Ingénieurs
- **Research organizations**
  - Cologne University of Applied Sciences
  - University of Manchester
  - Instituto Politécnico de Braganca
  - ICCS of the National Technical University of Athens
  - Czech Technical University of Prague
  - University of Hagen

- **User organizations**
  - EADS Innovation Works
  - Airbus
  - Iacobucci

2.1.2 Problems of small lot production

Production of an aircrafts or ships differs from more common productions such as cars or electronics devices. Aircrafts and ships nowadays are usually produced in batches of two to five pieces or can be even completely customized for the customer. This production type is called the small lot production, which focuses on small series production of highly customized products. That means that, unlike the car production, the aircrafts or ships producers will sell only a small number of products of the same design, but with similar investments for design and production ramp-up. Ramp-up is the phase of transition between product development and the maximum capacity utilization. This phase is often affected by unpredictable disturbances such as design faults or delays in the supply chain, which can result in the reduction of targeted production rate and high additional production costs. Ramp-up phase also suffers from other issues such as:

- last-minute engineering changes
- new or changing requests by customers
- sufficiency of production capacities
- little control over pre-assembly by supply partners

The growing competition and shorter innovation cycle leads to a rising number of ramp-ups. Experience from German shipyards shows that the production achieves the expected level of planning, product maturity and manufacturing quality after production of the third product of a series. Frequently occurring error showed that the product designs, production lines, suppliers, IT and logistics are not ready during production of the very first products of the series. Consequently the future production process needs to become more dynamic, event-sensitive, focusing on scheduling, optimization and controlling. The production management needs to be deeper integrated and balance several factors, as quality, risks, costs and critical application time issues.

2.1.3 Aim of the ARUM project

The ARUM research project aims to improve planning and control systems for the manufacturing of complex, small lot production such as aircraft, aircraft interiors or ships.

The main objective of the ARUM project is to significantly increase the operational and economic performance of production ramp-ups by delivering novel strategies, ICT systems and tools for automation control and optimization, to the industrial end-users.

2.1.4 ARUM solution approach

For achieving the above described objectives the ARUM focuses in research in the following areas:

- Risk management
  - Multiple risk mitigation strategies development for the planning phase and real-time production of the production ramp-up and small lot production.
- ICT system platform
  - Development of a new generation of service-oriented enterprise information platform, integrating service-based architecture and knowledge-based multi-agent systems(MAS).
- Planning and production tools
  - Development of a tool, supporting control and dynamic optimization of factory assets.
- Legacy system integration
  - Integration of the ARUM solution into legacy systems and information aggregation across existing legacy systems and production levels.

ARUM is strongly end-user driven and the results will be tested on industrial use cases with a focus on aircraft and aircraft interiors. The solution will be validated in a real industrial environment by industrial partners and benchmarked against today's ICT solutions.

At the end of the ARUM project, the results will be transferred to academic partners, for the further use of the ARUM infrastructure in other use cases by European researchers or industrial end-users.

2.2 ARUM solution architecture

Below is presented a general overview of the arum platform architecture. Only relevant entities with respect to the Factory Network Scenario Designer (FNSD) communication with other services) and communication canals are illustrated. Factory Network Scenario Designer is a working name of the previously mentioned managerial tool.

2.2.1 ARUM system components description

- **Factory Network Scenario Designer** – Represents a user client for operating the ARUM platform tools. It is separated to three main modules, each used by different ARUM user role. The FNSD is a thin client using the Client Service for handling user requests via REST provided interface. Design and development of this application is the main objective of this thesis.

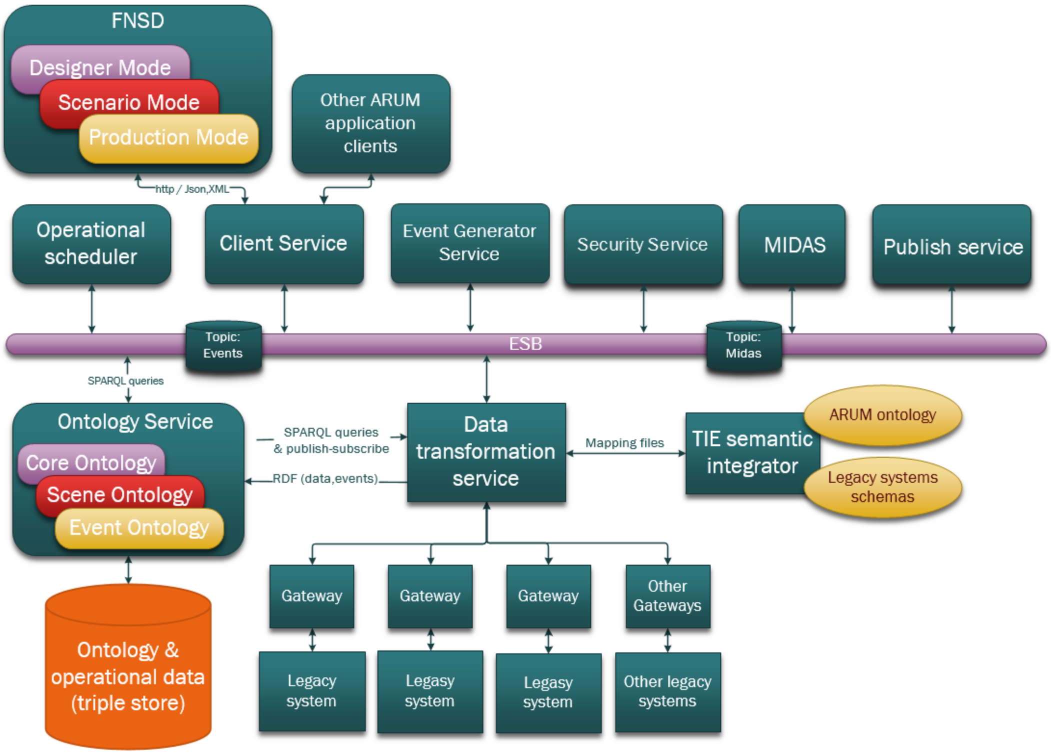


Figure 2.1: Arum platform achitecture overview

- **Other ARUM application clients** – Represents any other ARUM tool (worker console, warehouse console, etc.)
- **Client Service** – The purpose of the Client Service is handling request's incoming from the UI clients. It provides a REST interface for the client application tools. The data interchange between this component and internal ARUM entities is provided via the Enterprise Service Bus (ESB).
- **Operational scheduler** – ARUM Tool for task scheduling/planning.
- **Event generator Service** – The "Event Generator Service" is used by the Scenario mode with the purpose of obtaining generated hypothetical production events on request. This component will be used in scenario simulations.
- **Security Service** – Service, which handles the client authentication and maintains client's permissions.
- **MIDAS** – A component for the purpose of storing production events along with their respective solutions. The component will provide data to facilitate the troubleshooting

process. Specifically, in the case of the FNSD, the component will be able to provide the estimated troubleshoot time for given production events.

- **Ontology Service** – Service, which provides data in the "ARUM ontology format." The data will be loaded from the triple store or retrieved from the „Data transformation service."
- **Data transformation service** – Service, which provides the transformation of data from legacy systems into the RDF format.
- **TIE semantic integrator** Component, which provides data mapping files. Its purpose is the data transformation from legacy systems into the RDF format.
- **Publish service** – Service, which handles upcoming messages from ARUM components. After a message is received, it is forwarded to the corresponding "topic". Every topic has a unique identifier.
- **Production events queue** Component of the "publish-subscribe" messaging model. It represents a queue – like data structure receiving production event messages. In this case, the source of a production event can be an ARUM client, or legacy systems.
- **Midas queue** – Component of the "publish-subscribe" messaging model. It represents a queue – like data structure receiving messages from Midas.

2.3 ARUM industrial partners description

Following sections will briefly describe the manufacturing processes of ARUM industrial partners Airbus and Iacobucci.

2.3.1 Airbus general process description

Airplane industry is a Make-to-Order business where every order entails a high level of customization. Customization covers a large number of activities starting with the collection of customer needs until the design implementation in the aircraft by engineering facilities and the final delivery of the aircraft. Numerous actors and competencies are involved. For the ARUM use-case, an A350 long-distance aircraft has been picked from the Airbus portfolio.

The assembly of aircraft is organized as a line of workstations. Each station is organized as self-contained cell equipped with workers and production manager. International suppliers provide a high percentage of parts. Due to the high value, parts are ordered and assembled just-in-time. Therefore, suppliers deliver components and modules to the distribution center. From there, they are pulled to the assembly station by the production control manager according to just-in-time. The goal of the capacity alignment process is to ensure that, at all times, all resources required to run production at the maximum possible rate is available.

Production control manager is in charge for the work order management, capacity alignment and the material pull. The update is carried out at the beginning of every shift. The goal of work order management is to keep production running as smoothly as possible despite disturbances like material shortages or unexpected issues during assembly operations. Customization increases the number of new parts and components in the assembly process, which leads to an additional variation of processing times at every station. When not handled properly the result might be delays. [7].

2.3.2 Iacobucci general process description

The portfolio of products of Iacobucci(IHF) that is considered to be integrated into the ARUM use-case includes four standard products, of which one is in ramp-up and the production of harnessing:

- **Espresso makers** (about 1,000 / year, holding more than 90 % of the market)
- **Coffee makers** (about 1,000 / year)
- **Trash compactors** (about 140 / year)
- **Induction heating units** (new product in ramp-up phase, 20 / year planned)
- **Harnessing for galley appliances** (as internal supplier)

The assembly lines for the products run in parallel. Workstations in assembly lines are dedicated to particular products. Two people share the workload on each workstation in the assembly line. In an average, they assemble five or six units per week. Components are taken from the kitting. The work and related signatures by workers, quality checks by the Operative Area Responsible (RAO) and related reporting are organised into a number of cycles defined for a particular product. A lower performance typically depends on Missing Resource events(MR) and, sometimes, on failures detected in checks or final tests.[7]

2.4 Requirements

This section describes application user roles and functional and nonfunctional requirements on the designer mode and the production mode of the FNSD. These requirements were gathered from the both use-cases[7] of our industrial partners and user feedbacks from end-user meetings.

2.4.1 User roles specification

Several user roles were identified in both use cases, which were then transformed into ARUM user roles. Finally, mapping between both use cases roles and ARUM roles has been created. The four of these ARUM user roles are appropriate for using the same tool with unified user interface as their use cases are closely related and sometimes even overlaps.

ARUM Role	Functionality
Process Manager	1. Provide technological process for order execution. 2. Provide technological process for non-conformities fixing.
Production/Planning Manager	1. Create a long-term plan for a workshop. 2. Create schedule for the workshop. 3. Receive information on changes in orders, supplies, etc. 4. Receive up-to-date information on the current situation in production. 5. Play what-if games.
Team Leader	1. Assign tasks to workers. 2. Monitor task execution.
Station Manager	1. Create the schedule for the workstation. 2. Monitor schedule execution. 3. Adjust the schedule according to incoming events. 4. Provide information on changing situation. 5. Manage workers, their skills etc.

Table 2.1: Factory network designer primary user roles

Because four different user roles will use this tool, it will be divided into the three modules. That way, each user will have accessible only a set of functionalities and tools, which are required for his work. The three modules are the Design mode, the Scenario mode and the Production mode. Design mode and production mode are further described in Design3 and Implementation4 chapters. The Scenario mode is not in scope of this thesis, but it will use some of the components from the design mode. User role assignments to each mode are as follows:

ARUM Role	Assigned application mode
Process Manager	Design mode
Production/Planning Manager	Scenario mode Production mode
Team Leader	Production mode
Station Manager	Production mode

Table 2.2: User roles application assignment

2.4.2 Functional requirements

This section contains functional requirements for the Design and the Production modes of the application.

Definition: *Functional requirements specify the software functionality that the developers* *must build into the product to enable users to accomplish their tasks, thereby satisfying the business requirements.*[18]

2.4.2.1 Functional requirement 1.0

ID: FR_01
TITLE: Log-in to the application
MODE: All
DESC: All users allowed to use the application should be able to log-in via login form after its start. The system should automatically load the user information and set permissions according to his role.
RAT: In order for a user to use the application
DEP: None

2.4.2.2 Functional requirement 2.0

ID: FR_FR02
TITLE: Create new or modify assembly plan
MODE: Design mode
DESC: The Process Manager should be able to create an assembly plan of a new product. This includes:

- Creating work orders and their dependencies
- Adding jobs to the work orders and creating jobs dependencies
- Editing jobs information
  - Setting required resources and materials
  - Setting workers required and their skills
  - Adding related assembly instruction documents.

RAT: In order for the Process Manager to create a new product assembly plan.
DEP: FR_01 [2.4.2.1]

2.4.2.3 Functional requirement 3.0

ID: FR_03
TITLE: Load or copy an existing assembly plan
MODE: Design mode
DESC: The Process Manager should be able to load or copy an existing assembly plan.
RAT: In order for the Process Manager to adjust the existing assembly plan according to the customer order.
DEP: FR_01 [2.4.2.1]

2.4.2.4 Functional requirement 4.0

ID: FR_04
TITLE: Save or load an assembly plan from local or remote storage
MODE: Design mode, Scenario mode
DESC: The Process Manager should be able to save assembly plan to the local storage so no one else can use it before all changes has been made. After he finishes the work, he should be able to upload it to a remote storage (Ontology Service) for future use by others.
RAT: In order to store unfinished work in local storage.
DEP: FR_01 [2.4.2.1],FR_02 [2.4.2.2]

2.4.2.5 Functional requirement 5.0

ID: FR_05
TITLE: Visualization of an assembly plan
MODE: All
DESC: The assembly plan may have consist of a large number of work orders and jobs. The user should be capable of browsing through the assembly plan layers easily even if it is very extensive. In order to fulfill this function requirement the tool has to support following nonfunctional sub-requirements:

ID: FR_05.1
TITLE: Notation
DESC: The assembly plan should be displayed in a standardized notation such as BPMN or Petry nets. application.

ID: FR_05.2
TITLE: Zoom
DESC: The user should be able to zoom in and out the assembly plan to see different details of it. application.

ID: FR_05.3
TITLE: Semantic zoom
DESC: The visualization component should support semantic zoom to display only the most important parts of the assembly plan at a time.

ID: FR_05.4
TITLE: Layers browsing
DESC: The visualization component should support browsing throw the different layers of the assembly plan.

ID: FR_05.5
TITLE: Layouts
DESC: The visualization component should organize the assembly plan with various graph layouts for easy reading.

ID: FR_05.6
TITLE: Highlighting
DESC: The visualization component should support highlighting a predecessors and successors of selected task.

ID: FR_05.7
TITLE: Searching
DESC: The user should be able to search in the assembly plan according to specified criteria.

ID: FR_05.8
TITLE: Table form
DESC: The visualization component should support displaying a list of tasks of the assembly plan in form of a table.

RAT: In order to display the assembly plan in an "easy to use" form.
DEP: None

2.4.2.6 Functional requirement 6.0

ID: FR_06
MODE: Design mode, Scenario mode
TITLE: Assign an assembly plan to a scene
DESC: The Process Manager should be able to assign an assembly plan to a scene. I.e. assigning work orders of the assembly plan to the predefined work stations.
RAT: In order to define which workstation will perform what work order.
DEP: FR_01 [2.4.2.1],FR_02 [2.4.2.2],FR_03 [2.4.2.3]

2.4.2.7 Functional requirement 7.0

ID: FR_07
TITLE: Visualization of production process in time
MODE: Production mode
DESC: The Production/Planning Manager should be able to see up-to-date information on the current situation in production and its performance. This includes following sub-requirements:

- Provide a customizable dashboard start screen with important information summary different for each user role:
  - Status of work orders
  - Live schedule (Gantt chart)
  - Jobs to be done today

  - Availability of resources
  - List of events (NCs, missing resources)
  - Shifts information
  - Present management staff
  - KPIs

- Provide different views on production progress
  - Production progress information of each work station
  - Production progress information of each product
  - List of events for each product
  - Schedule of the work stations in production line
  - Human resources availability and their workload
  - Other resources availability and their workload

- Provide production performance information:
  - Evaluation of critical path in Gantt chart
  - Evaluation of critical station (station with the longest lead time)
  - Evaluation of critical resource
  - KPIs for each station, product, and entire production
  - Minimum of tardiness (per shift, station, entire line) with or without a minimal additional costs

RAT: In order to user could monitor the production process.
DEP: FR_01 [2.4.2.1],FR_05 [2.4.2.5]

2.4.2.8 Functional requirement 8.0

ID: FR_08
TITLE: Resolving of event
DESC: The Production or Station Manager should be able to react on an incoming event and be able to resolve it, i.e:

- create new traveling work
- reschedule task
- modify the assembly plan
- delegate event resolution
- invoke a new event
- see the event resolution progress

- see a list of related events
- see the event impact on the production

RAT: In order to handle events.
DEP:FR_01 [2.4.2.1],FR_02 [2.4.2.2],FR_05 [2.4.2.5],FR_07 [2.4.2.7]

2.4.3 Nonfunctional requirements

Definition:*Requirements which are not specifically concerned with the functionality of a system. They place restrictions on the product being developed and the development process, and they specify external constraints that the product must meet.*[17]

2.4.3.1 List of common nonfunctional requirements

ID:	NFR_01
TITLE:	Nonfunctional requirement 1.0
DESC:	Java technologies will be used for creation of the application.

ID:	NFR_02
TITLE:	Nonfunctional requirement 2.0
DESC:	Application client will be integrated to the ARUM platform via REST Client service interface.

ID:	NFR_03
TITLE:	Nonfunctional requirement 3.0
DESC:	Application client will be build upon the Netbeans or the Eclipse Rich client platform.

ID:	NFR_04
TITLE:	Nonfunctional requirement 4.0
DESC:	Application client functionality will be divided into the three modules, uses shared core api.

2.5 Solution approach

Based on previous described use-cases and list of requirements a low-fidelity prototype has been created.

„*Low-fidelity (lo-fi) prototyping is characterised by quick and easy translation of high-level design concepts into tangible and testable artefacts. Lo-fi also know as low-tech, as the means required for such an implementation consist, most of the time, of a mixture of paper, cardboard, post-it notes, acetone sheets, etc. A clear advantage of lo-fi prototyping is its extremely low cost and the fact that non-programmers can actively be part of the idea-crystallisation process.*“[10]

For the purpose of easy sharing of created designs with our European partners and for its fast modifications, the prototype has been created in tool Balsamiq Mockups[2]. Balsamiq is a small graphical tool to sketch out user interfaces for websites, desktop or mobile applications. It supports exporting created mock-ups to the image format or exporting all mock-ups to the pdf document. It is possible to join mock-ups together via hyper-links so the result pdf prototype can be highly interactive.

The thesis guidelines declare that the tool will consists of the four modules. However, during the design phase of the tool, has been decided, that first and second module from the guidelines will be merged into one module called the Scenario designer, which is out of scope of this thesis. However, for designing of the factory layout in the Scenario mode will be used extended Designer component[3.2.3]. As mentioned before the Scenario designer mode will be used for factory layout designing, playing what if games and for comparing of scheduling results. Detailed description of this mode can be found in the bachelor thesis of student Luis Moreno[15].

Chapter 3

Design

This chapter presents the application user interface design. The first section describes the general application layout and its basic components. The second and third section of this chapter describes the user interface and specific components for the Design and Production mode.

3.1 Application layout and common components

The default application layout consists of four main parts with four default application panels, see Figure[3.1]. The main layout sections are:

- **Section 1** - Ribbon
  Ribbon is a modern graphical component containing tabbed toolbars, filled with graphical buttons and other controls, which are grouped by functionality. It uses tabs to expose different sets of controls, eliminating the need for many parallel toolbars. It can contain contextual tabs that appear only when the user needs them. Each of the application modes will have defined its set of tabs and functionalities. The Ribbon bar can be minimized but unlike the other panels, it can not be docked to the other part of the application main window.

- **Section 2** - Browser Panel
  The browser panel is located by default on the middle left side of the main application window, above the navigation panel. The browser panel component displays hierarchical data in a tree form. The panel is organized into several tabs, each containing a different data (assembly plans, scenes, etc.). Different tabs are displayed in different modes (e.g. SD – scenes, simulations, ND – Assembly plans, scenes, etc.). A detailed description of the displayed data in each mode is provided in the corresponding sections.

- **Section 3** - Navigation and properties panel
  The properties and navigation panel is located by default on the bottom left side of the main application window, below the browser panel. The properties panel will

display detailed information about objects (assembly plans, jobs, scenes, stations, etc.) displayed in the Main content panel. If the user selects any object located in the Main content panel, its properties will be displayed in the properties panel. The navigation panel will be, by default, folded together with the Properties panel creating a single component, and it will display (if supported) a content thumbnail of the tab currently holding the focus in the Main content panel. The thumbnail displayed in the navigation panel displays a complete preview of a given tab (the whole scene, all work orders in the assembly plans). The user is allowed to navigate across the object by moving the visible section box and the position of the view in the Main content window will change accordingly.

- **Section 4** - Main content panel
  The Main content panel is located by default in the middle of the main application window and it serves as the main working area. The panel can display data or other components in multiple tabs. The user is allowed to open, close or move tab to a separate window or change its position in the tab header.

Beyond these sections, there is a name of the logged user as well as log out button in the right top part of the ribbon and the bottom part of the main application window is reserved for additional controls such as zoom or status indicators.

The content of the mentioned panels is further described in following sections.

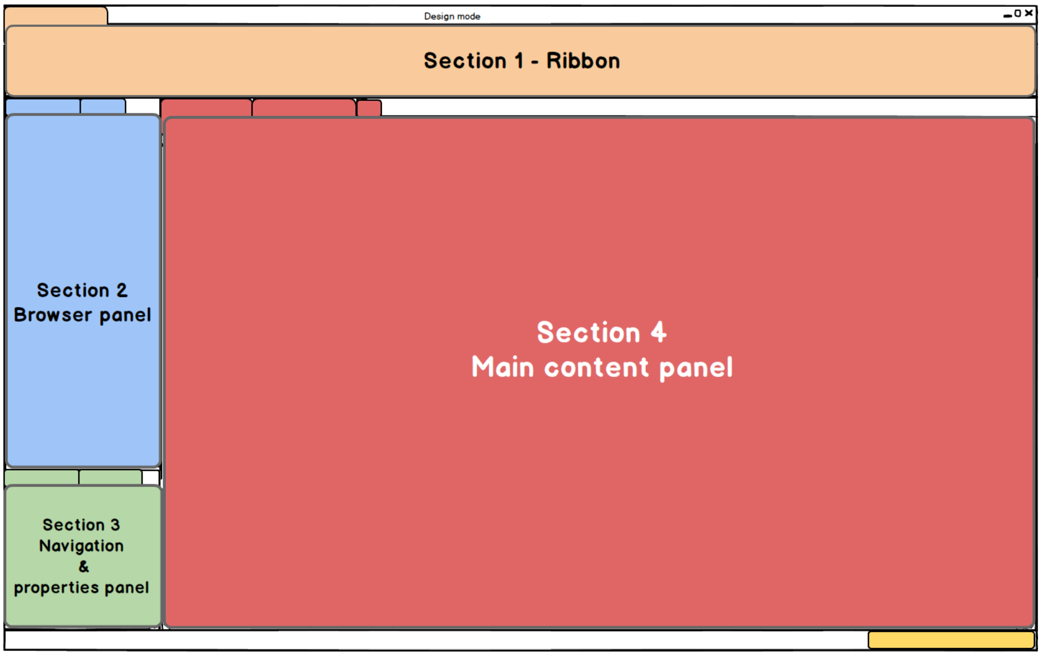


Figure 3.1: Application sections and layout

3.2 Designer mode

The Designer mode is used by the Process Manager to create or edit assembly plans of the product. He can also edit assembly plan of a product that is already in production, for example, for fixing non-conformities.
Designing of an assembly plan consist of creating work orders, jobs and their mutual dependencies, setting required resources and finally assigning work orders to the work stations which will execute them.

The Designer mode will fulfill requirements one to six from the functional requirements list[2.4.2].

3.2.1 Designer mode ribbon

In first release of the application, the ribbon of the designer will be rather simple. The main tab (Home) consist of control button groups for designing the assembly plan (Blocks), general tools group (Tools) and button group for switching between display modes (View as), see Figure[3.2].

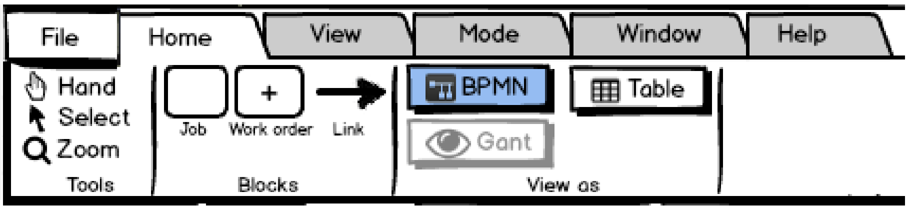


Figure 3.2: Designer mode ribbon

3.2.2 Designer mode browser panel

In Design mode, the browser panel contains two tabs: Assembly plan tab and Scenes tab. The assembly plans tab shows opened assembly plans in the form of a tree structure. User can browse through the assembly plans and display selected part in Assembly plan designer, see Figure[3.3]. The Scenes tab contains a list of opened scenes. User can assign assembly plan to the scene through it. Scene creation is not in scope of this thesis. The icon in front of the assembly plan or scene name indicates in which storage is it stored. Icon of house indicates a local storage and planet icon indicates a remote storage. User can upload the assembly plan into the remote storage in save as dialog or by right-clicking on it and selecting "save to remote storage" option.

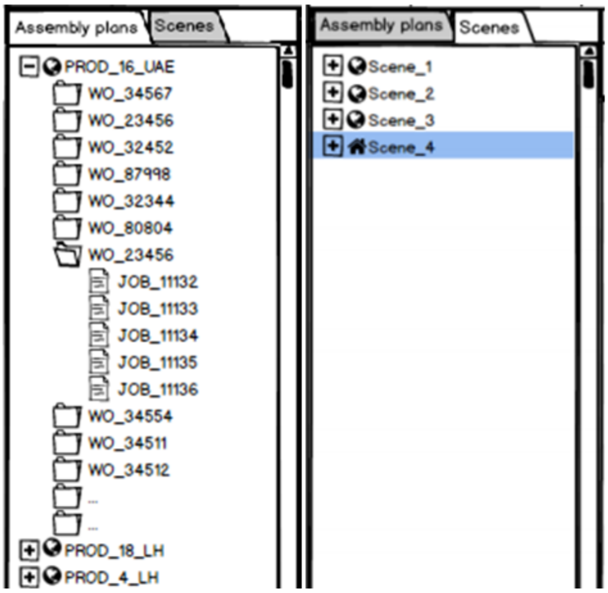


Figure 3.3: Designer mode browser panel

3.2.3 Assembly plans designer

Visualization and editing of assembly plans will be a major part of the Designer mode functionality. The assembly plan may consist of a large number of work orders and jobs. Designer component will fulfill all functional requirements FR_02 [2.4.2.2] and FR_05 [2.4.2.5].

3.2.3.1 Visualization notations

Designer component should support displaying the assembly plan in multiple notations such as BPMN[16], UML[8], Petry nets[12], etc. In the first release of the assembly plan designer component will use only customized BPMN notation, as it serves all requirements and can display any assembly plans from defined use cases. Designer can be extended by other notation in the future, if BPMN will not be sufficient, or expanded by another BPMN elements (for example events elements).

3.2.3.2 Visualization of assembly plan layers

Assembly plan consists of a list of work orders, and their mutual dependencies. Each work order consists of a list of sub-work orders or a list of jobs. Every job then has defined a list of required materials, resources, human resources and assembly instructions. Designer has to be capable of displaying these layers and allow the user to navigate through them. This can be done either by that each work order and job expands to its sub-layer, see Figure[3.4] or by displaying overall assembly plan in the form of WBS. For describing and displaying the assembly plan, we have selected a sub-set of the BPMN notation elements.

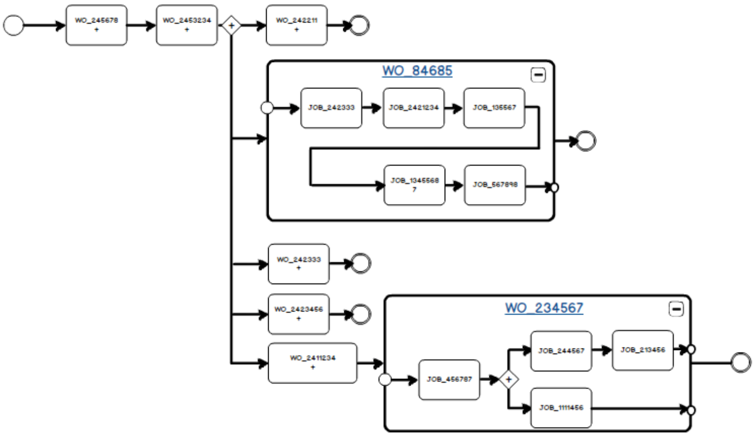
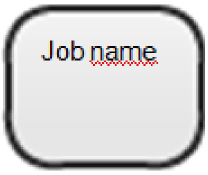
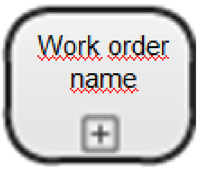
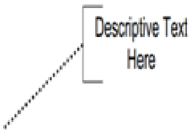
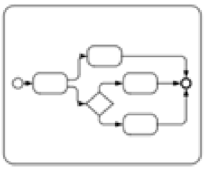
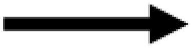
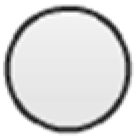
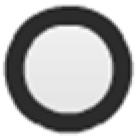
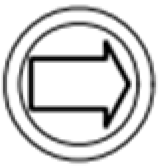
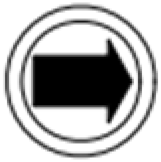
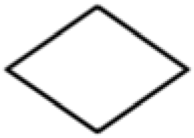
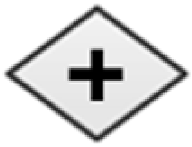


Figure 3.4: Assembly plan layers example

List of used BPMN elements and their description follows:

Element	Name	Description
	**Job**	Job is an atomic part of the assembly plan. It cannot be expanded to sub jobs or sub work orders. Every job contains its list of required materials, resources, human resources and their skills and all assembly instructions. When selected, user can browse or edit those lists in a separate panel, or see a general description in the properties menu. User can also display additional information text by moving mouse cursor over the Job shape.
	**Collapsed work order**	Work order is a group of related jobs or other work orders. By selecting the work order (mouse click) user can see a general description of the work order in the properties menu. User can expand the work order by double click. User can also display additional information text by moving mouse cursor over the Collapsed work order shape.

	**Descriptive text**	Additional descriptive text is displayed when user moves the mouse cursor over the Job or Collapsed work order. Content of the text depends on the configuration.
	**Expanded work order**	Collapsed work order is expanded when user double clicks on it. Size of the work order shape is enlarged to the size of its content and screen position and zoom is adjusted that the whole work order content is displayed. User can collapse it by clicking on the minimize icon in its top right corner.
	**Dependency**	Work orders and jobs may be connected with mutual dependencies. For example, if one job is connected to another it means that the destination job cannot start before the source job ends.
	**Start**	Start shape indicates start of the production process and that the following tasks do not depend on any other tasks. Assembly plan can contain multiple start shapes. Start shape can be sticked to the left edge of the expanded work order. This indicates that all following tasks will start immediately with the start of the work order. If there are other start shapes inside the work order, then the following tasks can start any time depending on the scheduler decision.
	**End**	End shape indicates end of the assembly process branch. Assembly plan can contain multiple end shapes. Production is finished when all assembly process end points are reached.
	**In-station connector**	When work orders of the assembly process are divided to the stations, this connector indicates that the subsequent work orders have dependencies located on previous station. When user clicks on this connector, tool should display the work orders assigned to the previous station and highlight these dependencies. The name of the station should be displayed in descriptive text.

	**Off-station connector**	When work orders of the assembly process are divided to the stations, this connector indicates that preceding work orders have dependencies located on the next station. When user clicks on this connector, tool should display the work orders assigned to the next station and highlight these dependencies. The name of the station should be displayed in descriptive text.
	**Gateway**	A Gateway is used to control the divergence and convergence of Sequence Flows in a Process. Thus, it will determine branching, forking, merging, and joining of paths. Internal markers will indicate the type of behavior control (see below).
	**Parallel gateway**	Parallel Gateway is used for forking and joining. Parallel gateway can be used for dividing of a path into two or more parallel paths (also known as an AND-Split). It is a place in the Process where activities can be performed concurrently, rather than sequentially. Joining path refers to the combining of two or more parallel paths into one path (also known as an AND-Join or synchronization). A Parallel Gateway is used to show the joining of multiple Sequence Flows.

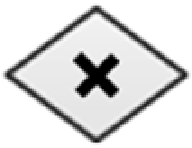
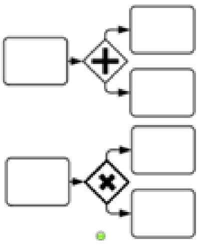
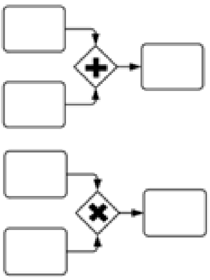
	**Exclusive gateway**	. Exclusive gateway Exclusive Gateway is used for branching and merging. This Decision represents a branching point where alternatives are based on conditional expressions contained within. Only one of the alternative path will be chosen. Merge refers to the exclusive combining of two or more paths into one path (also known as an OR-Join). A Merging Exclusive Gateway is used to show the merging of multiple Sequence Flows. In the first release of the application, this gateway will be used only for display the state of assigning the same task to the multiple stations, see section Work orders assignment[3.2.4].
	**Fork and branch**	. Fork means that all following tasks will be performed concurrently. Branch means that only one branch of the following tasks will be performed based on the conditional. In current application case, the scheduler will decides on which station will be the task performed, see Work orders assignment[3.2.4].
	**Join and merge**	. When two or more paths are join, the following tasks cannot start before all joining task are finished (Synchronization point). When two or more paths are merged together, the following tasks can start when one of the joining tasks is finished. Currently used only in work orders assignment display, see Work orders assignment[3.2.4].

Table 3.1: Designer component elements

Previous core elements description can be further extended and modified in the future development.

3.2.3.3 Layouts strategies

Extensive assembly plans can be hard to read due the large number of tasks and their dependencies. A graph layout strategy has to be used to sort the assembly plan elements for easier reading. Layout system has to ensure that tasks will not overlap each other and dependencies won't overlaps tasks. Dependent tasks should be placed near to each other and in the predefined order, for example, from left to right or in the top down order. Layout system should also reduce length and number of relation's edges (edges can overlaps each other). Suitable layout strategies are: Hierarchical Layout, Orthogonal Layout, Tree Layout, Organic Layout, see Figure[3.5] for layout strategies example.

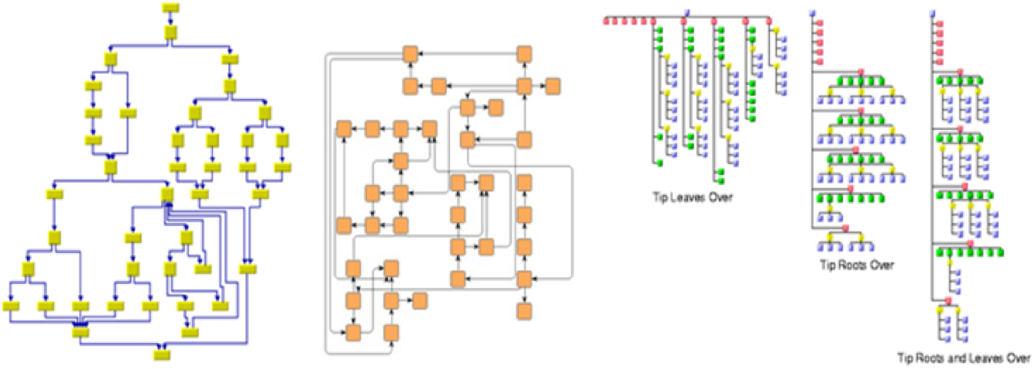


Figure 3.5: Hierarchical, Orthogonal and Tree Layout examples

3.2.3.4 Assembly plan modification

Designer component has to support assembly plan modifications such as adding new tasks, create their dependencies or edit their properties. User will do this by selecting the desired element from ribbon menu and clicking on the free space in the working area or connecting two tasks together. For editing the properties, he only selects the task and change their properties in the properties panel or, in case of job resources modifications, in the panel that will appear at the right side of the main content panel.

3.2.3.5 Filters

Designer component has to support filters, for example, to show the critical path in assembly plan, or displaying tasks status (used in production mode, see Figure[3.26], to show finished tasks, planned tasks, etc. - Required in the Production mode). It should also provide a function to display only selected task and its dependencies at a time and hide everything else. By selecting any dependent task, designer will focus on it and display only it and its dependencies and again hides everything else, see Figure3.6: Selection filter example.

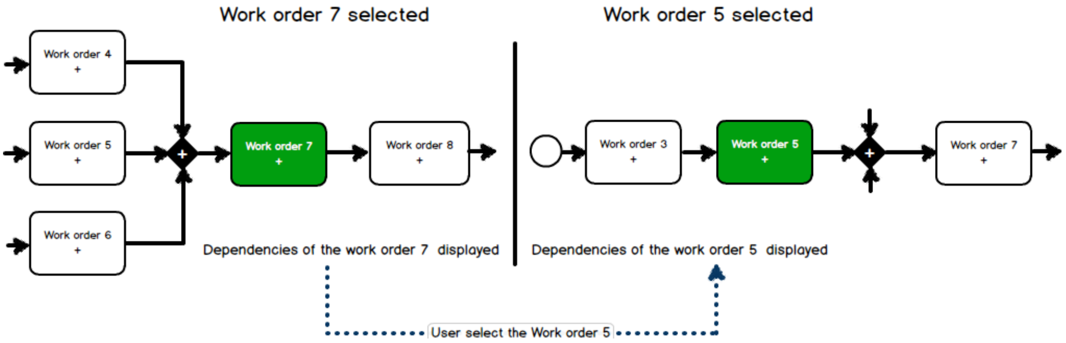


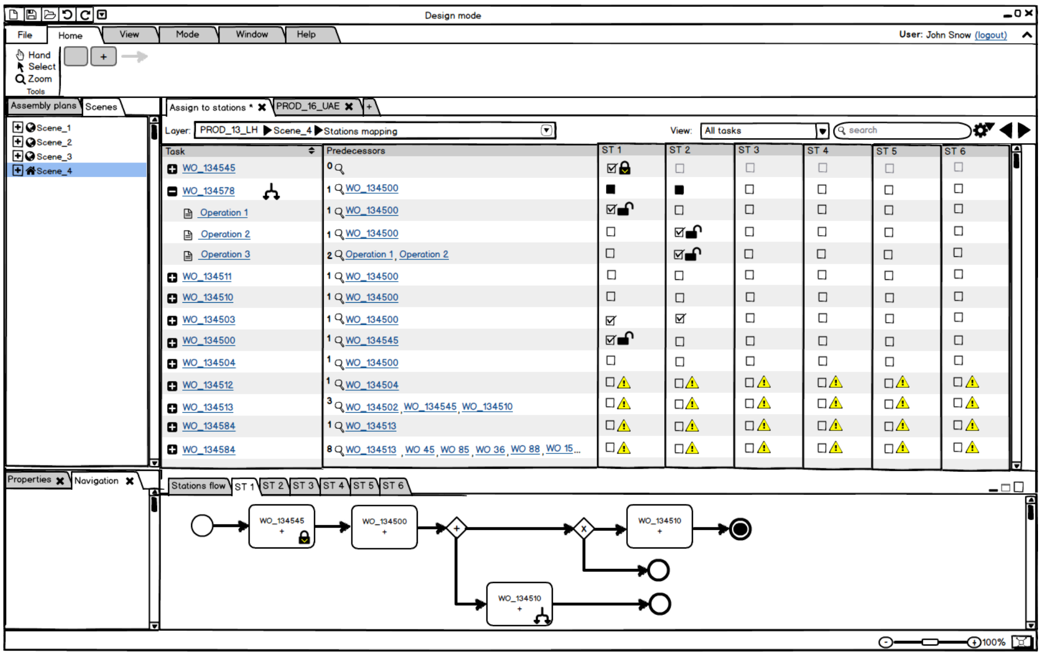
Figure 3.6: Selection filter example

3.2.3.6 Searching

Designer has to support searching in assembly plan by the given criteria. It will use search component, that is in the top right corner of each tab. Results, matching the given criteria, should be highlighted with color. By clicking on the "find the next result" button, designer will focus on the next found work order/job.

3.2.4 Work orders assignment

This component is used for assigning work orders to workstations, see Figure[3.7].



Layer: PROD_13_LH ▶Scene_4 ▶Stations mapping

View: All tasks

search

Task	Predecessors	ST 1	ST 2	ST 3	ST 4	ST 5	ST 6
⊞ WO_134545	0	☑🔒	□	□	□	□	□
⊟ WO_134578	1 WO_134500	■	■	□	□	□	□
Operation 1	1 WO_134500	☑🔓	□	□	□	□	□
Operation 2	1 WO_134500	□	☑🔓	□	□	□	□
Operation 3	2 Operation 1 , Operation 2	□	☑🔓	□	□	□	□
⊞ WO_134511	1 WO_134500	□	□	□	□	□	□
⊞ WO_134510	1 WO_134500	□	□	□	□	□	□
⊞ WO_134503	1 WO_134500	☑	☑	□	□	□	□
⊞ WO_134500	1 WO_134545	☑🔓	□	□	□	□	□
⊞ WO_134504	1 WO_134500	□	□	□	□	□	□
⊞ WO_134512	1 WO_134504	□⚠	□⚠	□⚠	□⚠	□⚠	□⚠
⊞ WO_134513	3 WO_134502 , WO_134545 , WO_134510	□⚠	□⚠	□⚠	□⚠	□⚠	□⚠
⊞ WO_134584	1 WO_134513	□⚠	□⚠	□⚠	□⚠	□⚠	□⚠
⊞ WO_134584	8 WO_134513 , WO 45 , WO 85 , WO 36 , WO 88 , WO 15...	□⚠	□⚠	□⚠	□⚠	□⚠	□⚠

Figure 3.7: Work orders assignment component

Work orders assignment component is composed of two main parts:

- Table with list of work orders
- Display window

Table with list of work orders consist of following columns:

- Name of the work order or job.
  - Column can be sorted in ascending or descending order.
  - Before each task name can be one of the following icons:
    - ⊞ expandable task (work order)
    - ⊟ collapsible task (work order)
    - 🗎 leaf task (job)

  - This column contains a tree structure of the assembly plan meaning that a list of sub-work orders or sub-jobs of the each task can be expanded or collapsed by clicking on the corresponding icon.

- List of predecessors
  - Consists of three parts:
    - Total number of predecessors
    - Display all predecessors button
    - List of all predecessors separated by comma
  - All rows with predecessors of the selected task can be highlighted in the table, by clicking on Display all predecessors button. Selected task will be highlighted by predefined color and all predecessors by another, see Figure[3.8].
  - Single predecessor task can be highlighted, by clicking on its name from the predecessors list.
  - Position of highlighted tasks that are not currently in visible area will be shown on the right side of the table vertical scrollbar, see Figure[3.8].
  - When predecessors are highlighted, a button with display icon will appear above the table, which, when pressed, will cancel all highlights.
  - Column can be sorted in ascending or descending order by the number of predecessors.

- Columns with workstations
  - Number of columns corresponds with number of workstations in flow.
  - Each column has the same name as the station it represents.
  - Each row consists of a checkbox, which indicates if task can be processed on this station.
  - Workstations columns can't be sorted.

Table with list of work orders can be filtered by selecting one of the following options from dropdown located on the left side of the table search input:

- Show all tasks
- Show unassigned tasks
- Show assigned tasks

Station assignment rules are rather complex, because the component has to avoid faults like wrong tasks dependency order. For the list of indicators and icons of this component see Table[3.2.4]. The assignment rules are as follows:

When a single station is selected for the task, button with an open lock icon appears next to its checkbox. By pressing the open lock button the task is marked as "locked" which means that it has to be done on this selected station and user prohibits creating traveling work. Open lock icon will change to closed lock icon, and all other unselected checkboxes will be disabled. User can cancel the task lock by clicking on the closed lock button again. By locking the task to the station, component has to disable all assign checkboxes of upper flow stations for all successors of that task.

Task can be assigned to the multiple workstations (depends on the component configuration), by selecting multiple checkboxes. If the task is assigned to the more workstations it can no longer be locked and scheduler will decides, which station will execute it.

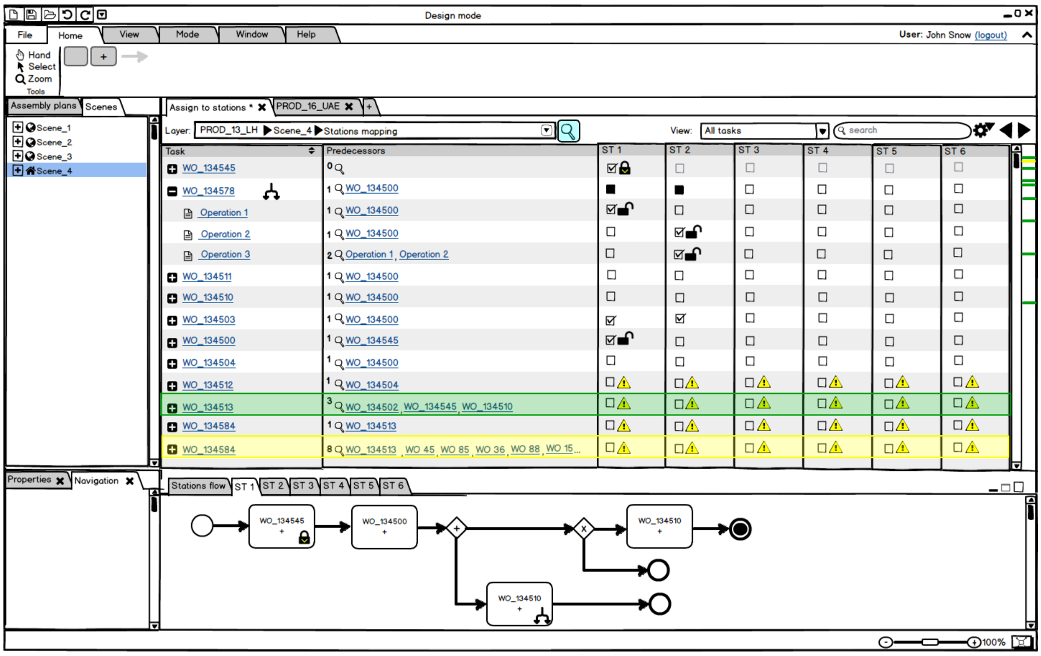
User can also assign subtasks to multiple stations (depends on the component configuration). If a subtask is assigned to multiple station than its main task check box is in an indeterminate state. If the user will not assign all subtasks than error icon will appear next to its main task name and this assignment will be invalid (can't be sent to the scheduler) until the user fixes it.

If any task predecessor has not been assigned to any station yet or is assigned to a station in lower station flow a warning icon will appears next to the corresponding station checkbox. When user tries to assign a task with unassigned predecessor to a station a warning window will pops up with an option to assign unassigned predecessors to the same station. If user decline to assign predecessors and force the assigning only the selected task, an error icon will appear next to that task name and this assignment will be invalid (can't be sent to the scheduler) until the user fixes it. If the user assigns subtasks to multiple stations correctly, a split icon will appear next to their main task name.

Display window is located beneath the Table with the list of work orders. It consists of one tab displaying the selected station flow and several others, each for one station in the flow. Each tab has the same name as its corresponding station and displays flow of its assigned tasks. If the task is assigned to multiple stations, it will be represented by a BPMN "or" gateway. If the task is locked than lock icon is also displayed in the left bottom part of the task box. If the task has subtasks assigned to the multiple stations than split icon will be displayed instead.

🔍	**Display predecessors**	Highlights the predecessors of the task
■	**Indeterminate state**	Indeterminate state of tree component
🔓	**Lock task option**	Lock task to the station
🔒	**Task locked**	Unlock task
⚠	**Warning**	Predecessors of this task has not been assigned
▲	**Error**	Invalid assignment
⑂	**Split task**	Sub tasks are assigned to the multiple stations

Table 3.2: Work orders assignment component icons and indicators



Task	Predecessors	ST 1	ST 2	ST 3	ST 4	ST 5	ST 6
WO_134545	0	☑🔒	□	□	□	□	□
WO_134578	1 WO_134500	■	■	□	□	□	□
Operation 1	1 WO_134500	☑🔓	□	□	□	□	□
Operation 2	1 WO_134500	□	☑🔓	□	□	□	□
Operation 3	2 Operation 1, Operation 2	□	☑🔓	□	□	□	□
WO_134511	1 WO_134500	□	□	□	□	□	□
WO_134510	1 WO_134500	□	□	□	□	□	□
WO_134503	1 WO_134500	☑	☑	□	□	□	□
WO_134500	1 WO_134545	☑🔓	□	□	□	□	□
WO_134504	1 WO_134500	□	□	□	□	□	□
WO_134512	1 WO_134504	□⚠	□⚠	□⚠	□⚠	□⚠	□⚠
WO_134513	3 WO_134502, WO_134545, WO_134510	□⚠	□⚠	□⚠	□⚠	□⚠	□⚠
WO_134584	1 WO_134513	□⚠	□⚠	□⚠	□⚠	□⚠	□⚠
WO_134584	8 WO_134513, WO 45, WO 85, WO 36, WO 88, WO 15...	□⚠	□⚠	□⚠	□⚠	□⚠	□⚠

Figure 3.8: Work orders assignment component

3.3 Production mode

Purpose of the production mode is to give the user up-to date information about the current production status. It consists of several views which should provide him with all the relevant information. The tool has to provide information about the global production state as well as the state of each of the production unit. All of the following views will be inside of the main content panel.

3.3.1 Production mode ribbon

The ribbon for the production mode consist of several groups of controls for fast switching between views, specific options for each view and scheduler controls.

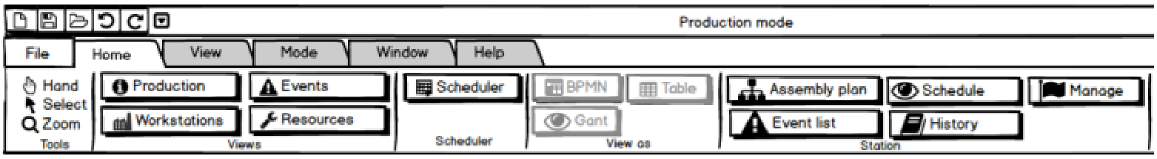


Figure 3.9: Production mode ribbon

The production ribbon contains the following groups of controls:

Name	Description
Tools	Group with zoom, select text or move buttons
Views	Fast access buttons to the production views
Scheduler	Scheduler options and status
View as	Controls for various display formats
Station	Options for the station view
Product	Options for the product view
Gantt	Gantt chart options

Table 3.3: Production ribbon groups of controls

3.3.2 Browser panel

The browser panel, see Figure[3.10], in production mode has two tabs: Production tab and Workstations tab. Production tab contains all products currently in the production. User can open a Product view by selecting any of the products in the browser. Icons before each product name indicate status of the product. Workstations tab contains a list of all workstations in the production line. User can open a workstation view by selecting any workstation in the list.

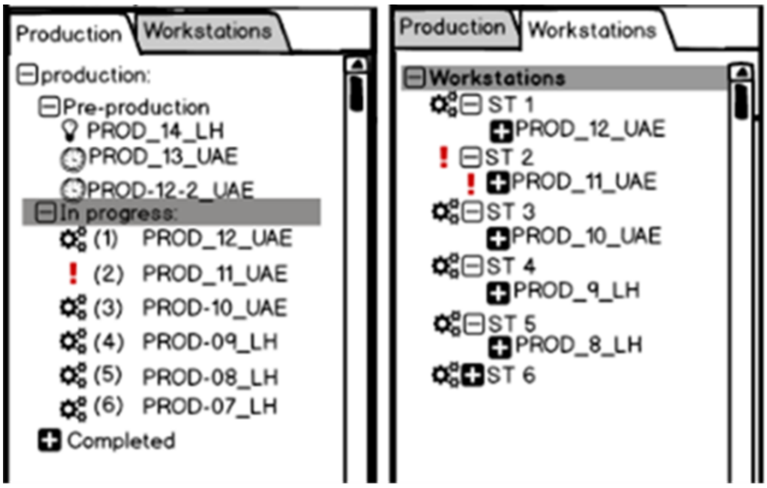
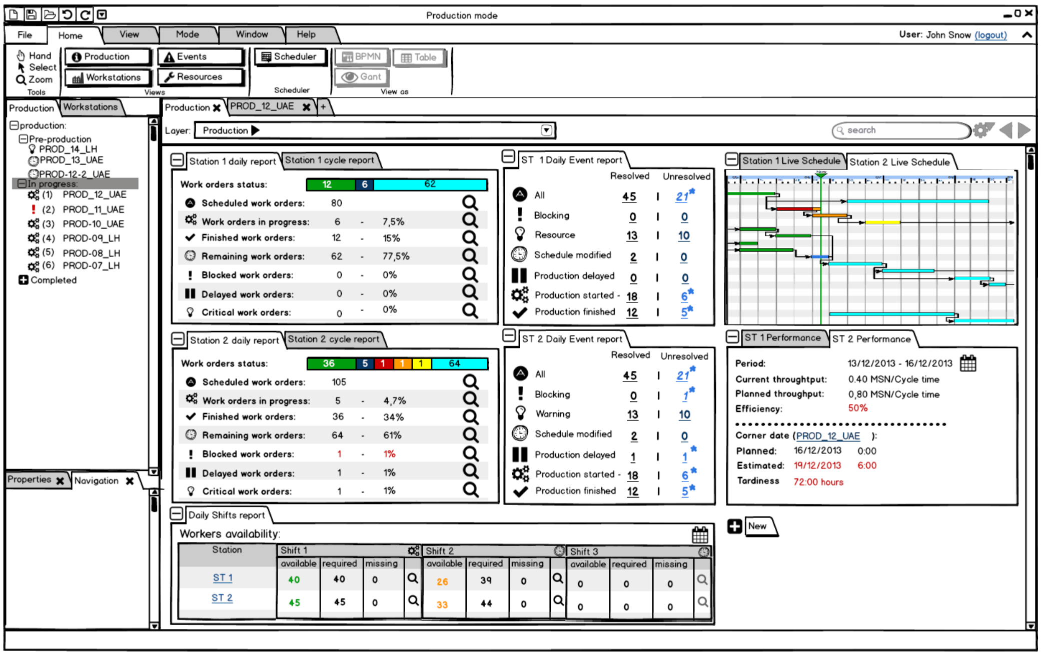


Figure 3.10: Production mode browser panel

3.3.3 Production mode dashboard

Production mode dashboard is customizable overview of the production, see 3.11 for the production dashboard example.

This view should open automatically after switching to the production mode, and it consists of several widgets that can be adjusted to the user needs. The widget option menu can be opened by clicking on the widget with right mouse button. This menu contains options to remove, minimize, move, or open the widget settings menu. The minimize button is also in the left corner of each widget. If the widget is minimized, then it is reduced to a maximize button and the widget header. User can add to the dashboard any number of widgets of the same type, each with different configuration so he can see, for example, daily reports of two different stations. Widgets can be resized, but maximum and minimum size of each widget depends on its configuration and content. Every ARUM role should have predefined set of default widgets. List of the considered widgets follows.



Station 1 daily report | Station 1 cycle report

Work orders status: 12 6 62

Scheduled work orders:	80		
Work orders in progress:	6	-	7,5%
Finished work orders:	12	-	15%
Remaining work orders:	62	-	77,5%
Blocked work orders:	0	-	0%
Delayed work orders:	0	-	0%
Critical work orders:	0	-	0%

Station 2 daily report | Station 2 cycle report

Work orders status: 36 5 1 1 1 64

Scheduled work orders:	105		
Work orders in progress:	5	-	4,7%
Finished work orders:	36	-	34%
Remaining work orders:	64	-	61%
Blocked work orders:	1	-	1%
Delayed work orders:	1	-	1%
Critical work orders:	1	-	1%

ST 1 Daily Event report

	Resolved		Unresolved
All	45	I	21*
Blocking	0	I	0
Resource	13	I	10
Schedule modified	2	I	0
Production delayed	0	I	0
Production started -	18	I	6*
Production finished	12	I	5*

ST 2 Daily Event report

	Resolved		Unresolved
All	45	I	21*
Blocking	0	I	1*
Warning	13	I	10
Schedule modified	2	I	0
Production delayed	1	I	1*
Production started -	18	I	6*
Production finished	12	I	5*

ST 1 Performance | ST 2 Performance

Period: 13/12/2013 - 16/12/2013
Current throughtput: 0.40 MSN/Cycle time
Planned throughput: 0,80 MSN/Cycle time
Efficiency: 50%

Corner date (PROD_12_UAE):
Planned: 16/12/2013 0:00
Estimated: 19/12/2013 6:00
Tardiness 72:00 hours

Daily Shifts report

Workers availability:

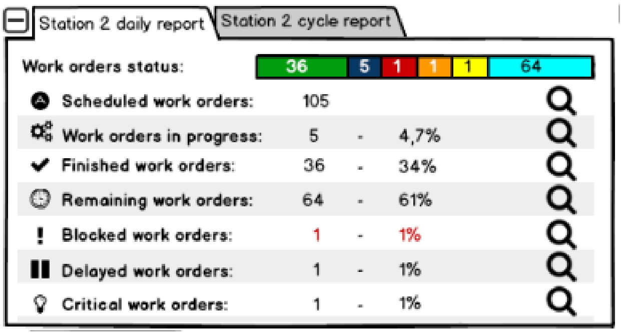
Station	Shift 1				Shift 2				Shift 3			
	available	required	missing		available	required	missing		available	required	missing	
ST 1	40	40	0		26	39	0		0	0	0	
ST 2	45	45	0		33	44	0		0	0	0	

Figure 3.11: Production dashboard - station manager view

3.3.3.1 Report widget

This widget provides the user with quick production report about the work orders progress, see FigureFigure[3.12].

The report widget displays work orders sorted by their status i.e. finished, scheduled, in progress, blocked, delayed, etc., in the form of a status bar and as a table with all these categories. User can display all category by clicking on the display button or a respective part of the status bar. User can configure the time period he wants to display(daily report, cycle report) and object of the report(station, product).



Station 2 daily report | Station 2 cycle report

Work orders status: 36 | 5 | 1 | 1 | 1 | 64

Scheduled work orders:	105		
Work orders in progress:	5	-	4,7%
Finished work orders:	36	-	34%
Remaining work orders:	64	-	61%
Blocked work orders:	1	-	1%
Delayed work orders:	1	-	1%
Critical work orders:	1	-	1%

Figure 3.12: Report widget

3.3.3.2 Schedule widget

This widget provides thumbnail on the current production schedule, see Figure[3.13]. User can set which production subject and in what time range he wants the schedule to display. It also serves as an easy access link to the detail view of the displayed schedule. Production subjects can be an overall production, station schedule, product progress, resources workload or workers schedule. Workers schedule can be displayed according to the given team or shift. Time range of the displayed schedule can be set to shift, day, week, station cycle or month. Widget supports multiple tabs to display several different schedules. Thumbnail can be enlarged by moving the mouse cursor over it for displaying more details. Schedule information automatically updates over time.

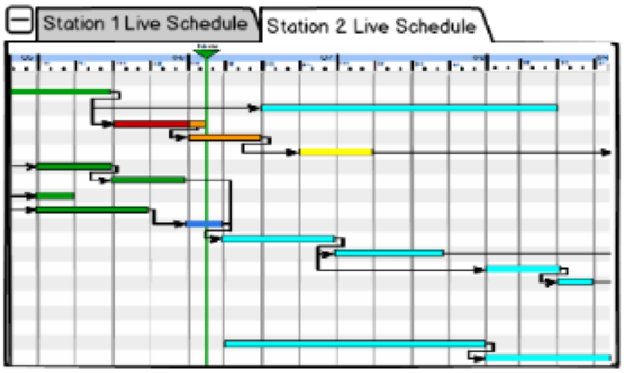
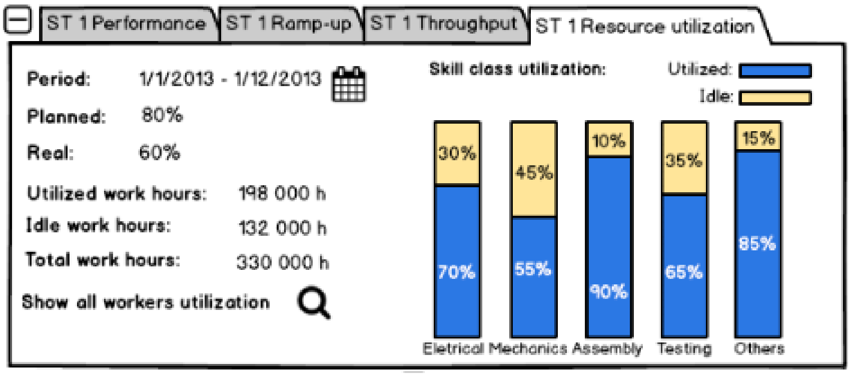


Figure 3.13: Schedule widget

3.3.3.3 KPIs widget

This widget shows key performance indicators, see Figure[3.14]. User can set what KPIs of which subject the (entire production, specific work station, specific products etc.) and in what time range he wants to display. The main KPIs is the Corner date, the Throughput and the Resource utilization. Clicking can change the time period on the calendar button, which will open time range option dialog. The widget also supports charts for displaying KPIs values.



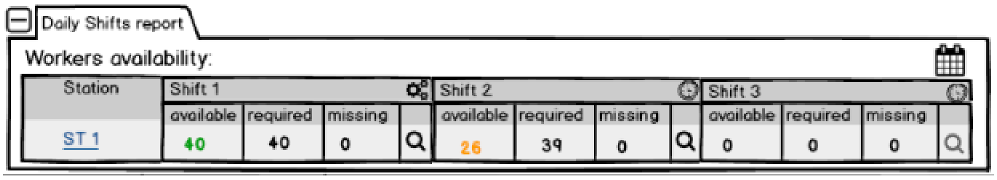
Period: 1/1/2013 - 1/12/2013
Planned: 80%
Real: 60%
Utilized work hours: 198 000 h
Idle work hours: 132 000 h
Total work hours: 330 000 h
Show all workers utilization

Figure 3.14: KPIs widget

3.3.3.4 Shifts widget

Shift widgets shows shifts overview and their workers availability, see Figure[3.15]. By default, it shows a daily report but can change the date by pressing calendar icon located in top right corner. Time range can be set to day, workstation cycle, week or month. Work progress table consists of "Station" column and one column for each shift, divided to the three sub-columns displaying available, required and missing workers. User can navigate to the workstation overview or the shift overview, by clicking on respective link in the table. Header of each Shift column contains one of the three shift status indicators:

- ✓ icon indicating that the shift is over.
- ⚙ icon indicating currently active shift
- 🕘 icon indicating a planned shift



Daily Shifts report

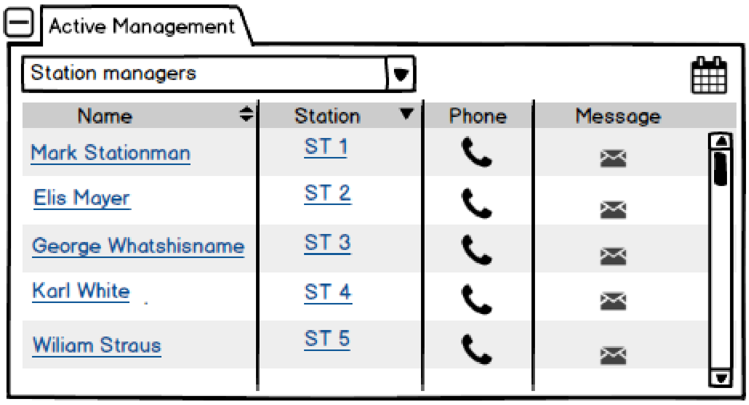
Workers availability:

Station	Shift 1			Shift 2			Shift 3		
	available	required	missing	available	required	missing	available	required	missing
ST 1	40	40	0	26	39	0	0	0	0

Figure 3.15: Shifts report widget

3.3.3.5 Active management widget

This widget show currently active management staff, see Figure[3.16]. User can select management category he wants to display (Station managers, Team leaders etc.). He can display phone number of selected person or send him a message by clicking on the respective icon. He can change the date by pressing the calendar icon in its top right corner (For example to see who will be active tomorrow).



Active Management

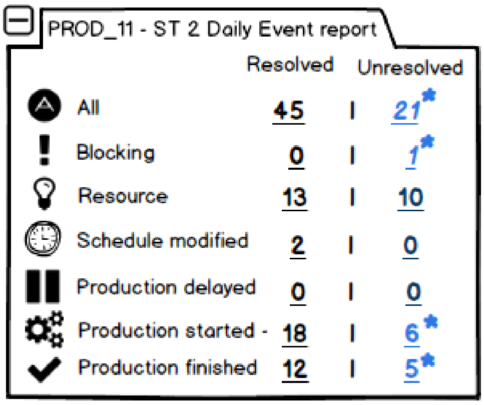
Station managers

Name	Station	Phone	Message
Mark Stationman	ST 1	☎	✉
Elis Mayer	ST 2	☎	✉
George Whatshisname	ST 3	☎	✉
Karl White	ST 4	☎	✉
Wiliam Straus	ST 5	☎	✉

Figure 3.16: Active management widget

3.3.3.6 Events report widget

This widget shows fast report about the incoming events, see Figure[3.17]. All events are sorted by category, easily recognizable by icons. Each event can be in one of three states: Resolved, Unresolved and New. Resolved event is the one that has been included to the schedule unlike the unresolved one. The New event is unresolved and user did not see its details yet. If there is a new event among the Unresolved ones, a star symbol appears at that category. User can quickly navigate to the list of events by clicking on the respective number.



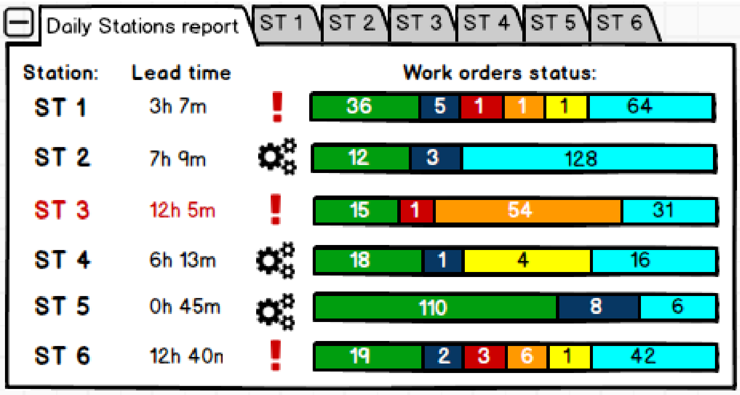
PROD_11 - ST 2 Daily Event report

	Resolved		Unresolved
All	45	I	21*
Blocking	0	I	1*
Resource	13	I	10
Schedule modified	2	I	0
Production delayed	0	I	0
Production started -	18	I	6*
Production finished	12	I	5*

Figure 3.17: Events report widget

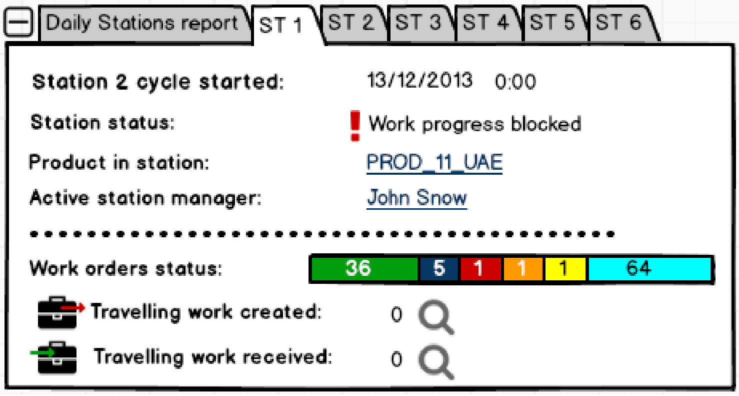
3.3.3.7 Workstations widget

This widget shows status and production progress of workstations in the production line. User can configure this widget to display all required workstations. The first tab displays a status report of all selected stations, see Figure[3.18], and for each station a report is displayed in its respective view, see Figure[3.19]. The stations status table in stations report consists of following columns: station name, station lead time, station status and station progress. User can quickly access the station detail, by clicking the station name. In Status column is an icon indicating the station status. If the user moves the mouse cursor over the icon a tooltip text with status description pops up. Lead time column can be configured to display station lead time in standard time format or IMS units. Critical station (station with the longest lead time) is highlighted with red font color. Progress column displays the station progress via progress bar. User can change the predefined time range in widget settings. The specific station tab contains the station cycle progress bar and production report: product in the station, information about traveling work, station status and active station manager.



Station:	Lead time		Work orders status:
ST 1	3h 7m	!	36 5 1 1 1 64
ST 2	7h 9m	⚙	12 3 128
ST 3	12h 5m	!	15 1 54 31
ST 4	6h 13m	⚙	18 1 4 16
ST 5	0h 45m	⚙	110 8 6
ST 6	12h 40n	!	19 2 3 6 1 42

Figure 3.18: Workstations widget



Station 2 cycle started: 13/12/2013 0:00

Station status: ! Work progress blocked

Product in station: PROD_11_UAE

Active station manager: John Snow

Work orders status: 36 5 1 1 1 64

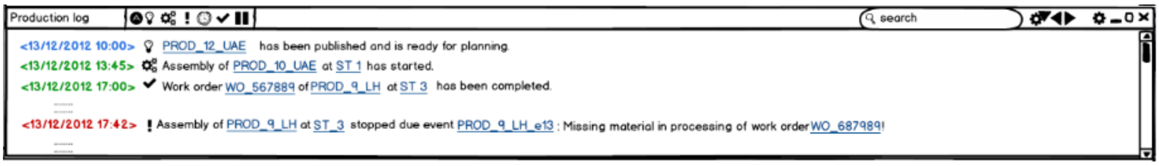
Travelling work created: 0

Travelling work received: 0

Figure 3.19: Workstations widget - station detail

3.3.3.8 Production log

A Production log is a component that displays events, messages and information incoming from production, see Figure[3.20]. Each message consists of a time stamp, its type, displayed by an icon, and text message with hyperlinks for easy access to its subjects.The timestamp is highlighted with predefined color depending on the type of the message. Production log support searching via search input located on the right side of its header. Messages can be filtered according to their type by activating/deactivating any of the filter type icon located on the left side of the log header. Icons of the messages and filters are same as in the list[3.5].

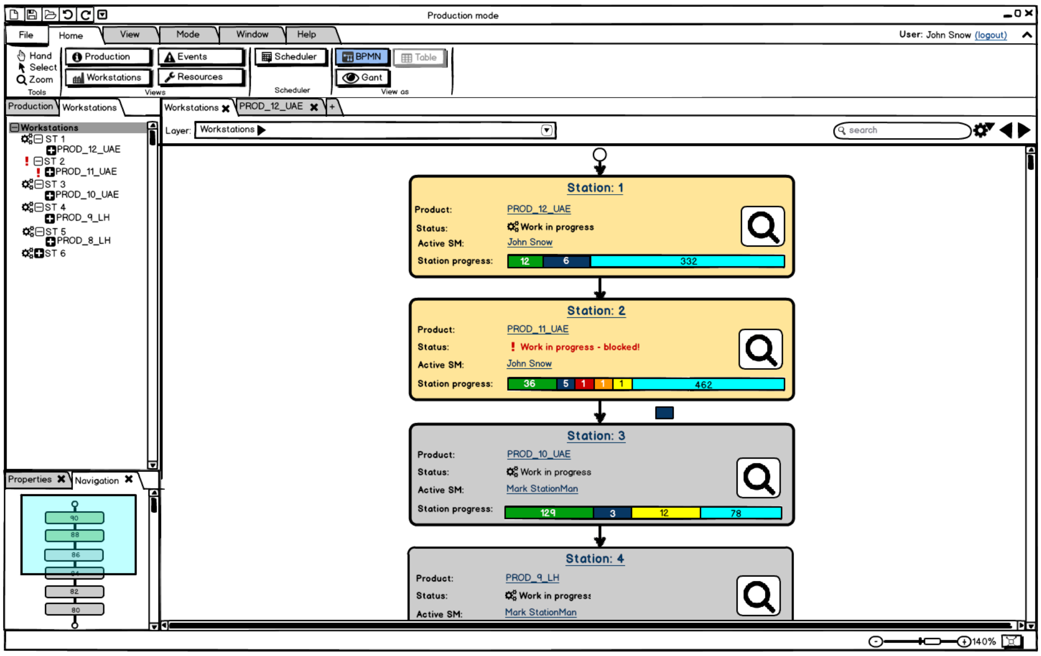


```
<13/12/2012 10:00>  PROD_12_UAE  has been published and is ready for planning.
<13/12/2012 13:45>  Assembly of PROD_10_UAE at ST 1 has started.
<13/12/2012 17:00>  Work order WO_567889 of PROD_9_LH at ST 3 has been completed.
      ........
      ........
<13/12/2012 17:42>  ! Assembly of PROD_9_LH at ST_3 stopped due event PROD_9_LH_e13 : Missing material in processing of work order WO_687989!
      ........
      ........
```

Figure 3.20: Production log

3.3.4 Work stations overview

This view shows the actual status of all workstations in a production line, see Figure[3.21]. It can display workstations flow in the BPMN notation where every station is represented as rounded-corner rectangle containing the most important information about the station status. It should contain Station name, name of the product in the station, the current state of the station and work orders progress bar. User can open a Work station details view[3.21] by selecting any station in the flow. Station flow should be also displayable as a table.



Station: 1
Product: PROD_12_UAE
Status: Work in progress
Active SM: John Snow
Station progress: 12 | 6 | 332

Station: 2
Product: PROD_11_UAE
Status: ! Work in progress - blocked!
Active SM: John Snow
Station progress: 36 | 5 | 1 | 1 | 1 | 462

Station: 3
Product: PROD_10_UAE
Status: Work in progress
Active SM: Mark StationMan
Station progress: 129 | 3 | 12 | 78

Station: 4
Product: PROD_9_LH
Status: Work in progress
Active SM: Mark StationMan

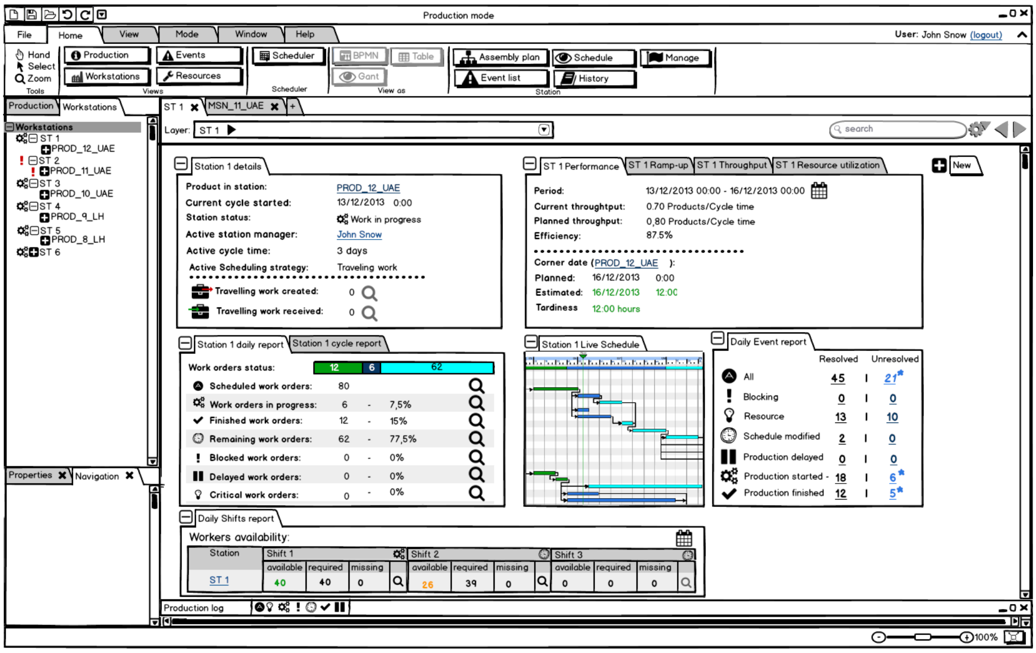
Figure 3.21: Workstations overview

3.3.5 Work station detailed view

This view shows detail information about selected workstation, see Figure[3.22].

Station widgets displays all information about the station. It can contain any widget regarding the workstations from the Production dashboard, see section Production mode dashboard[3.3.3] for the list of widgets. User can adapt the station view to his or her needs, for example, to see daily and cycle reports, KPIs, live schedule, etc.

All control buttons are located in the ribbon under the Station controls group. User can browse the part of the assembly plan of the product assigned to the selected station or open the workstation event list. He can also open the station schedule via the Schedule button or open Station assembly history via the History button.



Station 1 details

Product in station: PROD_12_UAE
Current cycle started: 13/12/2013 0:00
Station status: Work in progress
Active station manager: John Snow
Active cycle time: 3 days
Active Scheduling strategy: Traveling work
Travelling work created: 0
Travelling work received: 0

ST 1 Performance | ST 1 Ramp-up | ST 1 Throughput | ST 1 Resource utilization

Period: 13/12/2013 00:00 - 16/12/2013 00:00
Current throughtput: 0.70 Products/Cycle time
Planned throughput: 0,80 Products/Cycle time
Efficiency: 87.5%
Corner date (PROD_12_UAE):
Planned: 16/12/2013 0:00
Estimated: 16/12/2013 12:0C
Tardiness 12:00 hours

Station 1 daily report | Station 1 cycle report

Work orders status: 12 | 6 | 62

Scheduled work orders:	80		
Work orders in progress:	6	-	7,5%
Finished work orders:	12	-	15%
Remaining work orders:	62	-	77,5%
Blocked work orders:	0	-	0%
Delayed work orders:	0	-	0%
Critical work orders:	0	-	0%

Daily Event report

	Resolved		Unresolved
All	45	I	21*
Blocking	0	I	0
Resource	13	I	10
Schedule modified	2	I	0
Production delayed	0	I	0
Production started -	18	I	6*
Production finished	12	I	5*

Daily Shifts report

Workers availability:

Station	Shift 1			Shift 2			Shift 3		
	available	required	missing	available	required	missing	available	required	missing
ST 1	40	40	0	26	39	0	0	0	0

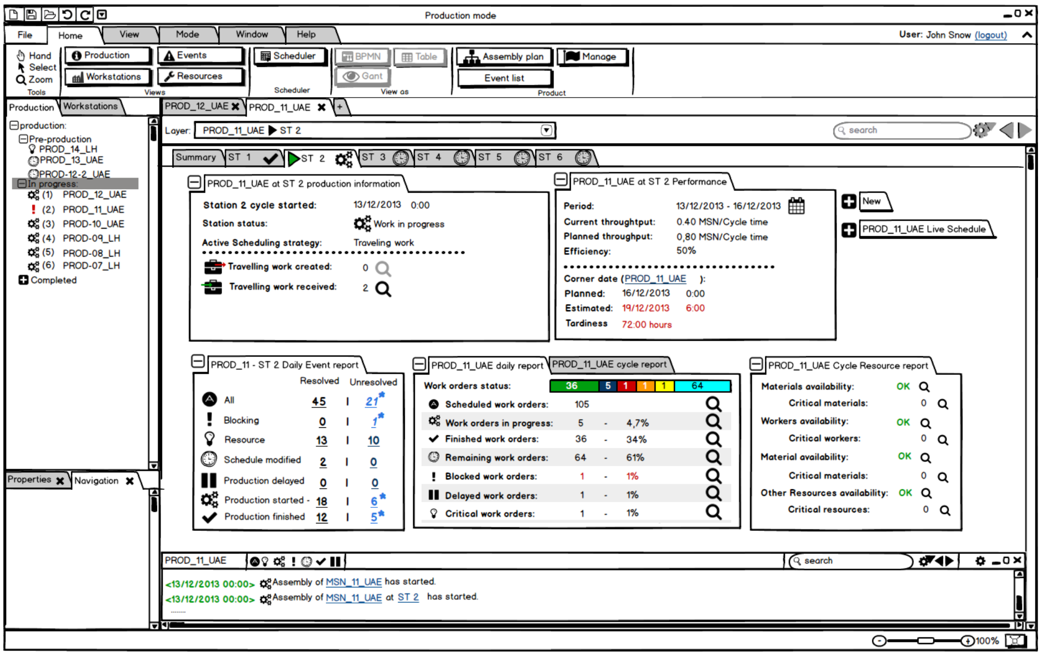
Figure 3.22: Work station details

3.3.6 Product detailed view

This view shows detail information about selected product, see Figure[3.23] User has to be able to browse through the whole product production process. Each workstation, the product is passing through, is displayed as a tab at the top side of the working panel. User can switch between those tabs to see all information about the product production on the selected station. The most left table shows product production summary. Each station tab header displays the production status icon, see list Production progress colors and icons[3.5]. The active station is marked with a green triangle.

Every tab can contain any widget regarding the product from the Production dashboard, see section Production mode dashboard[3.3.3] for the list of widgets. User can adapt the station view to his or her needs, for example, to see daily and cycle reports, KPIs, live schedule, etc.

All control buttons are located in the ribbon under the Product controls group. User can browse the part of the assembly plan of the product assigned to the selected station or open its event list.



production:
- Pre-production
  - PROD_14_LH
  - PROD_13_UAE
  - PROD-12-2_UAE
- In progress:
  - (1) PROD_12_UAE
  - (2) PROD_11_UAE
  - (3) PROD-10_UAE
  - (4) PROD-09_LH
  - (5) PROD-08_LH
  - (6) PROD-07_LH
- Completed

Layer: PROD_11_UAE ▶ ST 2

PROD_11_UAE at ST 2 production information

Station 2 cycle started: 13/12/2013 0:00
Station status: Work in progress
Active Scheduling strategy: Traveling work
Travelling work created: 0
Travelling work received: 2

PROD_11_UAE at ST 2 Performance

Period: 13/12/2013 - 16/12/2013
Current throughtput: 0.40 MSN/Cycle time
Planned throughput: 0,80 MSN/Cycle time
Efficiency: 50%
Corner date (PROD_11_UAE):
Planned: 16/12/2013 0:00
Estimated: 19/12/2013 6:00
Tardiness 72:00 hours

PROD_11 - ST 2 Daily Event report

	Resolved		Unresolved
All	45	I	21*
Blocking	0	I	1*
Resource	13	I	10
Schedule modified	2	I	0
Production delayed	0	I	0
Production started -	18	I	6*
Production finished	12	I	5*

PROD_11_UAE daily report | PROD_11_UAE cycle report

Work orders status: 36 5 1 1 1 64

Scheduled work orders:	105		
Work orders in progress:	5	-	4,7%
Finished work orders:	36	-	34%
Remaining work orders:	64	-	61%
Blocked work orders:	1	-	1%
Delayed work orders:	1	-	1%
Critical work orders:	1	-	1%

PROD_11_UAE Cycle Resource report

Materials availability:	OK
Critical materials:	0
Workers availability:	OK
Critical workers:	0
Material availability:	OK
Critical materials:	0
Other Resources availability:	OK
Critical resources:	0

PROD_11_UAE

```
<13/12/2013 00:00> Assembly of MSN_11_UAE has started.
<13/12/2013 00:00> Assembly of MSN_11_UAE at ST 2  has started.
```

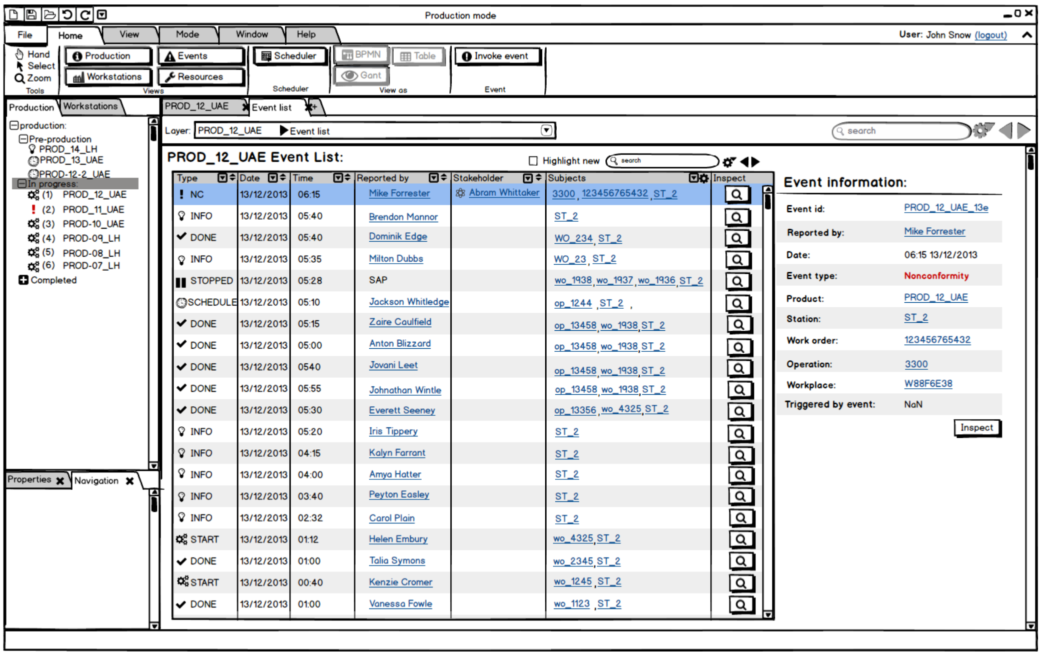
Figure 3.23: Product details

3.3.7 List of events

This view shows a list of events related to the selected production unit, for example, general list of events, product events, workstation events, etc. see Figure[3.24]. When user selects one, the event details will be displayed on the right side of the table, which contains the general information, event description and subjects description. Each event list table contains following columns:

Name	Sorting	Filters	Description
Type	yes	By event type	Type of the event – including icon
Date	yes	By date	Date of occurrence
Time	yes	By time	Time of occurrence
Reported by	yes	By source type (worker, SAP etc.)	Who reported the event
Subjects	no	By type (worker, product, workstation etc.)	Who reported the event
Inspect	no	no	Of what, the event relates to

Table 3.4: Events table columns



PROD_12_UAE Event List:

Type	Date	Time	Reported by	Stakeholder	Subjects	Inspect
! NC	13/12/2013	06:15	Mike Forrester	Abram Whittaker	3300, 123456765432, ST_2	
INFO	13/12/2013	05:40	Brendon Mannor		ST_2	
✓ DONE	13/12/2013	05:40	Dominik Edge		WO_234, ST_2	
INFO	13/12/2013	05:35	Milton Dubbs		WO_23, ST_2	
▮▮ STOPPED	13/12/2013	05:28	SAP		wo_1938, wo_1937, wo_1936, ST_2	
SCHEDULE	13/12/2013	05:10	Jackson Whitledge		op_1244, ST_2 ,	
✓ DONE	13/12/2013	05:15	Zaire Caulfield		op_13458, wo_1938, ST_2	
✓ DONE	13/12/2013	05:00	Anton Blizzard		op_13458, wo_1938, ST_2	
✓ DONE	13/12/2013	0540	Jovani Leet		op_13458, wo_1938, ST_2	
✓ DONE	13/12/2013	05:55	Johnathan Wintle		op_13458, wo_1938, ST_2	
✓ DONE	13/12/2013	05:30	Everett Seeney		op_13356, wo_4325, ST_2	
INFO	13/12/2013	05:20	Iris Tippery		ST_2	
INFO	13/12/2013	04:15	Kalyn Farrant		ST_2	
INFO	13/12/2013	04:00	Amya Hatter		ST_2	
INFO	13/12/2013	03:40	Peyton Easley		ST_2	
INFO	13/12/2013	02:32	Carol Plain		ST_2	
START	13/12/2013	01:12	Helen Embury		wo_4325, ST_2	
✓ DONE	13/12/2013	01:00	Talia Symons		wo_2345, ST_2	
START	13/12/2013	00:40	Kenzie Cromer		wo_1245, ST_2	
✓ DONE	13/12/2013	01:00	Vanessa Fowle		wo_1123, ST_2	

Event information:

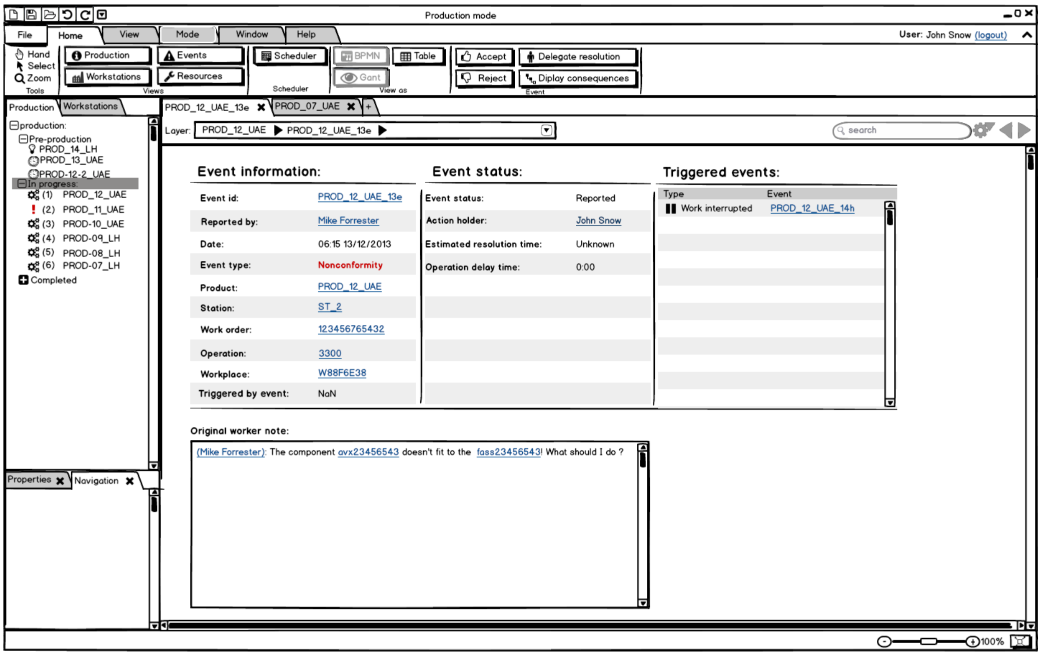
Event id:	PROD_12_UAE_13e
Reported by:	Mike Forrester
Date:	06:15 13/12/2013
Event type:	Nonconformity
Product:	PROD_12_UAE
Station:	ST_2
Work order:	123456765432
Operation:	3300
Workplace:	W88F6E38
Triggered by event:	NaN

Inspect

Figure 3.24: List of events

3.3.8 Event details

This view shows details about the selected event, see Figure[3.25]. This includes some general information about the event (ID, type, time stamp, event state, reported by), list of the event subjects, subject details, event specific details, event comments and list of related events. If the user is authorized to resolve the event, the option buttons will be displayed.



Layer: PROD_12_UAE ▶ PROD_12_UAE_13e ▶

Event information:

Event id:	PROD_12_UAE_13e
Reported by:	Mike Forrester
Date:	06:15 13/12/2013
Event type:	Nonconformity
Product:	PROD_12_UAE
Station:	ST_2
Work order:	123456765432
Operation:	3300
Workplace:	W88F6E38
Triggered by event:	NaN

Event status:

Event status:	Reported
Action holder:	John Snow
Estimated resolution time:	Unknown
Operation delay time:	0:00

Triggered events:

Type	Event
Work interrupted	PROD_12_UAE_14h

Original worker note:

(Mike Forrester): The component avx23456543 doesn't fit to the fass23456543! What should I do ?

Figure 3.25: Event details

3.3.9 Assembly plan production progress

This view uses the Assembly plan visualisation component, see section Assembly plans designer[3.2.3] for displaying production progress. It will label the work orders and jobs in assembly plan according to their actual state i.e. it will mark finished, blocked, currently processed, planned, and delayed tasks, see Figure[3.26]. Progress icons and colouring is as follows:

Icon	Name	Color	Description
❗	**Blocked task**	red	Task is blocked by an event. (NC or missing resource)
⚙	**Task in progress**	blue	Task is in progress without any problems.
✔	**Finished task**	green	Task has finished successfully.
⏸	**Delayed task**	orange	Start of the task production has been delayed due the problem at predecessor task or due the lack of resources.
🕘	**Scheduled task**	cyan	Task is scheduled and should start on time.
💡	**Problematic task**	yellow	There are some issues with this task (missing resource), which occurred before the task start time.
⛔	**Canceled task**	grey	Task has been canceled.
💼→	**Created travelling work**	none	Created travelling work. This task is passed to another station.
→💼	**Received travelling work**	none	Received travelling work. This task has been passed from another station.

Table 3.5: Production progress colors and icons

It is also possible to display the critical path by clicking on the Display critical path button at the main window ribbon.

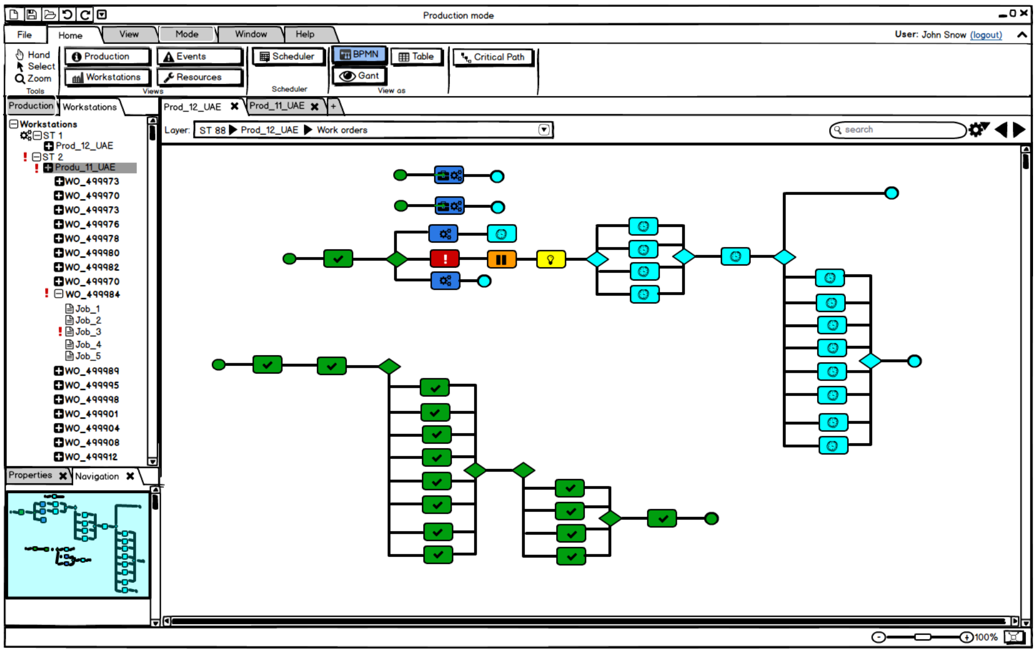
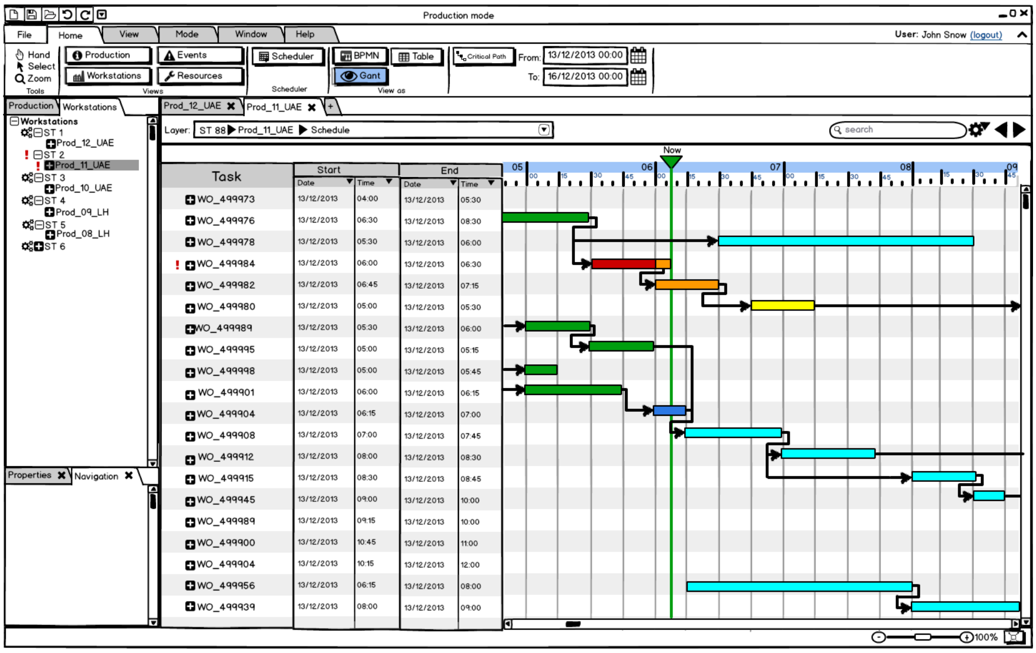


Figure 3.26: Assembly plan production progress

3.3.10 Gantt chart

Gantt chart is a standard graphical component for displaying schedule. All scheduled tasks are sorted in a table on the left side, together with their start/end timestamps, and their execution time and dependencies are displayed on the right side of the chart, see Figure[3.27].



From: 13/12/2013 00:00

To: 16/12/2013 00:00

Layer: ST 88 ▶ Prod_11_UAE ▶ Schedule

Task	Start		End	
	Date	Time	Date	Time
WO_499973	13/12/2013	04:00	13/12/2013	05:30
WO_499976	13/12/2013	06:30	13/12/2013	08:30
WO_499978	13/12/2013	05:30	13/12/2013	06:00
! WO_499984	13/12/2013	06:00	13/12/2013	06:30
WO_499982	13/12/2013	06:45	13/12/2013	07:15
WO_499980	13/12/2013	05:00	13/12/2013	05:30
WO_499989	13/12/2013	05:30	13/12/2013	06:00
WO_499995	13/12/2013	05:00	13/12/2013	05:15
WO_499998	13/12/2013	05:00	13/12/2013	05:45
WO_499901	13/12/2013	06:00	13/12/2013	06:15
WO_499904	13/12/2013	06:15	13/12/2013	07:00
WO_499908	13/12/2013	07:00	13/12/2013	07:45
WO_499912	13/12/2013	08:00	13/12/2013	08:30
WO_499915	13/12/2013	08:30	13/12/2013	08:45
WO_499945	13/12/2013	09:00	13/12/2013	10:00
WO_499989	13/12/2013	09:15	13/12/2013	10:00
WO_499900	13/12/2013	10:45	13/12/2013	11:00
WO_499904	13/12/2013	10:15	13/12/2013	12:00
WO_499956	13/12/2013	06:15	13/12/2013	08:00
WO_499939	13/12/2013	08:00	13/12/2013	09:00

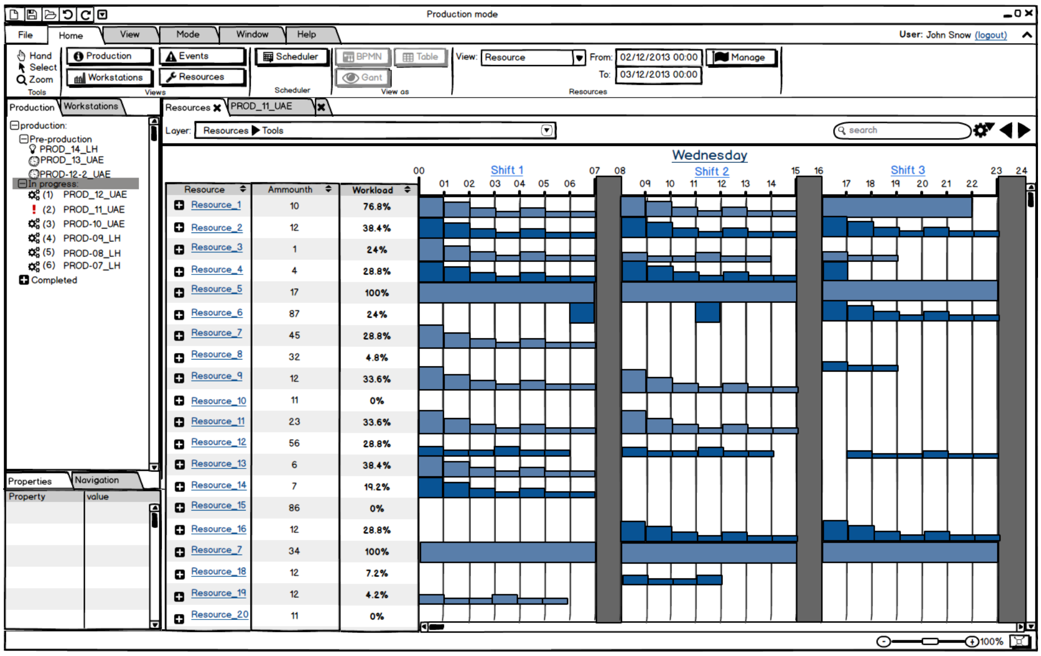
Figure 3.27: Gantt chart

3.3.10.1 Tasks schedule display

Gantt chart can be displayed for different production units i.e. specific workstation, specific product, across multiple stations, etc. It can display current time stamp (green horizontal line) and progress of the work orders, with the same colors as tasks in list Production progress colors and icons[3.5]. Each production unit/task can be expanded to its subtasks by clicking on the expand icon next to its name. Gantt chart should support zooming, searching, and displaying the critical path. Further design and implementation are the goals of Peter Mathia bachelor thesis.[14].

3.3.10.2 Resources workload display

Gantt chart can also be used for displaying the resource's workload and evaluate the critical resource, see Figure[3.28]. User can change the resource type and time range in Resources ribbon controls group. User can also change the period by clicking on the time links above the chart. When user moves his mouse cursor over the occupied time slot, a tooltip will be displayed with detailed information about what activity that has allocated the resource. Time slots highlighted with blue color indicates that the resource is allocated, white indicates that the resource is free to use, and grey slots mean that the resource cannot be allocated.

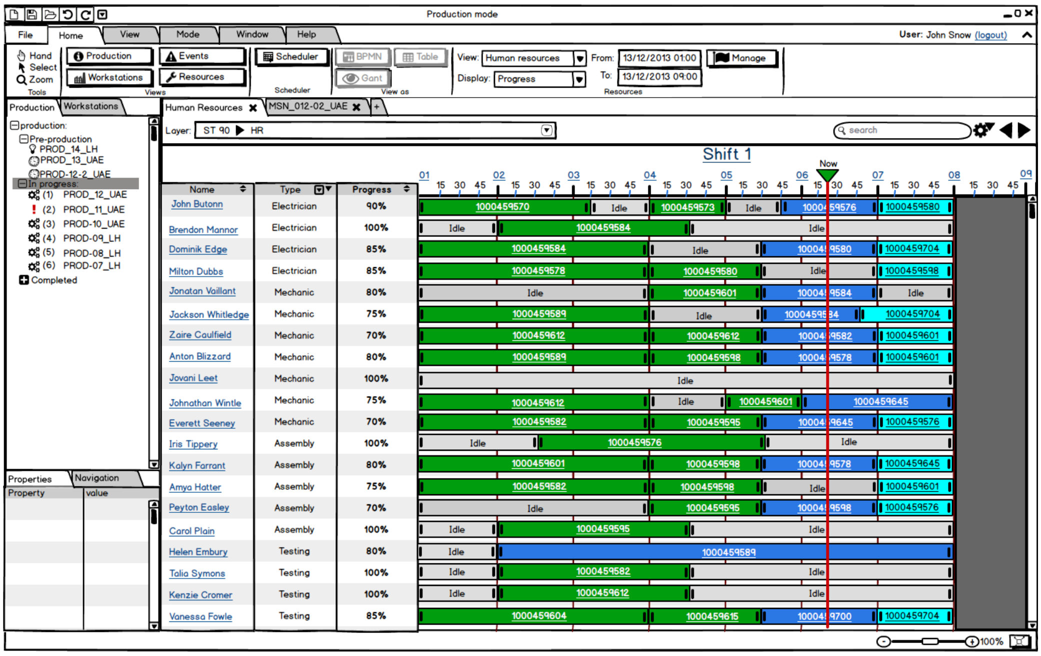


Resource	Ammounth	Workload
Resource_1	10	76.8%
Resource_2	12	38.4%
Resource_3	1	24%
Resource_4	4	28.8%
Resource_5	17	100%
Resource_6	87	24%
Resource_7	45	28.8%
Resource_8	32	4.8%
Resource_9	12	33.6%
Resource_10	11	0%
Resource_11	23	33.6%
Resource_12	56	28.8%
Resource_13	6	38.4%
Resource_14	7	19.2%
Resource_15	86	0%
Resource_16	12	28.8%
Resource_7	34	100%
Resource_18	12	7.2%
Resource_19	12	4.2%
Resource_20	11	0%

Figure 3.28: Resources workload chart

3.3.10.3 Human resources workload and schedule display

Displaying the HR workload and schedule is similar to the resource workload display, see Figure[4.1] Only difference is that when worker is idle, then the time slot is highlighted with yellow color, while when he is busy then the time slot is highlighted with green color. Unfilled time slot means that the worker is not at work.



Name	Type	Progress
John Butonn	Electrician	90%
Brendon Mannor	Electrician	100%
Dominik Edge	Electrician	85%
Milton Dubbs	Electrician	85%
Jonatan Vaillant	Mechanic	80%
Jackson Whitledge	Mechanic	75%
Zaire Caulfield	Mechanic	70%
Anton Blizzard	Mechanic	80%
Jovani Leet	Mechanic	100%
Johnathan Wintle	Mechanic	75%
Everett Seeney	Mechanic	70%
Iris Tippery	Assembly	100%
Kalyn Farrant	Assembly	80%
Amya Hatter	Assembly	75%
Peyton Easley	Assembly	70%
Carol Plain	Assembly	100%
Helen Embury	Testing	80%
Talia Symons	Testing	100%
Kenzie Cromer	Testing	100%
Vanessa Fowle	Testing	85%

Figure 3.29: Human resources workload and schedule

Chapter 4

Implementation

This chapter is dedicated to the implementation procedure used for application development. We explain what the Rich Client Platform (RCP) is, which RCP solutions are currently available and what are the main differences between them. Then, we further describe the Netbeans platform, which we choose for the application development and present some Netbeans modules and other libraries that we have used. Finally, we describe details about the implementation of proposed application.

4.1 Used platform and technologies

Previously described application design and requirements led the development to the decisive point, which platform and technologies will be used for its implementation. As shown in ARUM platform overview, application will be client connected to the ARUM platform via web service, so the choices were to either implement the application as a web based or as a desktop client. After considering pros and cons of both approaches, the desktop client solution has been chosen. The main reasons were that the divided development of web based client amongst multiple ARUM technology providers could be rather difficult. Also could appear potential browsers compatibility problems or documentation and code quality assurance issues. Also, AIB partner had a bad experience with this approach from previous projects.

The proposed application will be rather complex and developing it "from the scratch" in such a short time would be nearly impossible. Fortunately, stable solution, suitable for our case is available in the form of Rich Client Platform(RCP).

4.1.1 Rich Client Platform

Many desktop applications shares common features and behavior, like menus, toolbars, splash screen, internationalization, command line, contextual help and more. RCP is a program, which can be used as a basis of an application and provides an environment for its life cycle. It contains all regular parts of a desktop application, which can be used in developed application so there is no need of creating them again. RCP offers modular architecture, which results in a relatively loosely coupled relationship between independently functioning modules simplifying the dynamic extensibility of the application and the ability to swap its constituent parts.[9]

The main benefits of using the Rich Client Platform are:

- Platform independence
- Reduction in development time
- User interface consistency
- Autoupdating
- Reusability and reliability

There are two main Rich Clients Platforms currently available, with large community and developers support. There is also a Spring RCP, but community considers it as a dead project, since the last update release was in year 2007. The Eclipse and Netbeans platforms are very similar, the main differences are summarized in the following table.

	NetBeans Platform	**Eclipse RCP**
UI Toolkit	Standard Swing toolkit	SWT
UI Design	Matisse GUI Builder	Commercial alternatives
Module System	Standard OSGi module system or NetBeans-specific module system	Standard OSGi module system
Build System	Out of the box, Maven or Ant	Proprietary
JDK Support	VisualVM, a NetBeans Platform application,is in the JDK, so many JARs from the NetBeans Platform are in the JDK too	No equivalent support

Table 4.1: NetBeans and Eclipse RCP comparison [5]

We picked the Netbeans client platform for the further application development, because of its better knowledge across our development team. Also even the Netbeans supports the OSGi module system, unlike on the Eclipse platform, using it can be optional.

4.1.2 Netbeans RCP

The Netbeans RCP offers a large amount of frameworks, APIs and specifics, which we will use in development of our application. The most important ones are presented in following list:

- User interface framework
- Ribbon Bar
- Nodes API

- Visual Library API
- Internationalization
- Help system

4.1.2.1 User interface framework

The Netbeans user interface framework provides windows, menus, toolbars, browsers and other components. All components are based on AWT/Swing and the JavaFX is also supported. Integration of provided or user-created components into the application can be easily done via registering them into a layer.xml into the specified location among existing window system modes or, from NB version 6.9, via annotating the class with the Netbeans annotations. An example of registering an action via annotations, which is very similar to the component registration is shown in the following section.

4.1.2.2 Ribbon Bar

An open source extensible ribbon bar for NetBeans Platform applications is available.

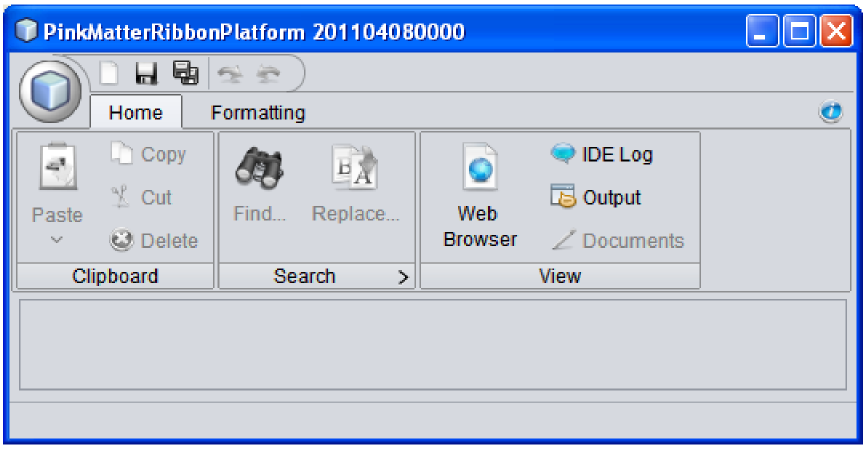


Figure 4.1: Netbeans Ribbon

Its integration into the application can be done simply by adding dependency on the PinkMatterPlatform module to the application and then by registering desired actions into it. Action is a class implementing ActionListener interface from standard Java API.[6]

```
@ActionID(category = "Draw", id = "org.draw.DrawAction")
@ActionRegistration(displayName = "#CTL_DrawAction", iconBase="org/draw/brush.png")
@ActionReferences({
    @ActionReference(path = "Ribbon/AppMenu", position = 0),
    @ActionReference(path = "Ribbon/AppMenuFooter", position = 0),
    @ActionReference(path = "Ribbon/TaskBar", position = 0),
    @ActionReference(path = "Ribbon/TaskPanes/Main/File", position = 0)
})
@Messages("CTL_DrawAction=Draw")
public final class DrawAction implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO implement action body
    }

}
```

Adding action into the Netbeans Ribbon code example

- **@ActionID** annotation defines unique identificator of the action and its category
- **@ActionRegistratio**n annotation defines action name that will be displayed and icon displayed in action button.
- **@ActionReference** annotation defines action button folder and position in the ribbon.
- **@Messages** annotation is used for internacionalization
- **actionPerformed** method will be called when user clicks on the action button

4.1.2.3 Visual Library API

Netbeans Visual Library API is a library for visualization of data structures, such as graphs. To use the Visual Library API, you need only define a dependency on the module. The components of the Visual Library API, like Swing, are structured and managed as a tree. The superclass of all graphic components is the Widget class. This library supports all features required for developing the Assebly plan designer component and Widgets in Production mode.

4.1.2.4 Localization

Localization is the process of providing of resources (text, icons, etc.) for a specific country or language. Well designed applications must be easily adaptable to the specific language or country. For this purpose, the Java API and the Netbeans Platform support localization at minimum effort. The best way for adding localization to the application on Netbeans platform is the use of @Messages annotations.

The @Messages annotation will generate a Bundle.java class and a Bundle.properties file. The Bundle.java class will include functions for localizing, and the Bundle.properties file contains the key-value pairs that determine the exact strings for the root locale.

In order to localize to another language, specific Bundle file can be added named Bundle_xx.properties where "xx" stands for the desired locale, for example, a Bundle_fr.properties contain the French localization.

Then, when the locale is set for the application, the Bundle.java class should use the correct Bundle_xx.properties file.

4.1.2.5 Help system

Help system in Netbeans is based on standard JavaHelp API. Netbeans platform provides module containing JavaHelp library and class for accessing it. After including the JavaHelp Integration module into the project, one can generate all necessary file via the "create new JavaHelp Help set" wizard. Wizard will generate skeletons of XML configuration files, which holds the topics of the help and java class, which will register help set into the Netbean's platform. Content of the help topic is a standard html page, which needs to be registered in respective configuration files.

4.2 Implemented module

ARUM platform development is still in an early stage and not all of its key services are ready to use. The Control service, which will handle requests from the application client via REST interface is also still missing. This means the client could not be integrated to the ARUM platform yet. Also, the Production mode design will be probably modified, based on the feedbacks from Iacobucci. The current implementation focused on the Designer mode, mainly on the assembly plan designer component, which currently supports plan visualization and editing. This component is a part of separated module and will be used in both modes. The implementation will continue, so the tool would support all described functionalities and be fully integrated into the ARUM platform by the second planned release D5.2.2[1] in November. The following section will show Application architecture and domain model. The Figure[4.5] presents a screenshot of the tool in current implementation state.

4.3 Application architecture

Current application implementation consist of the four following modules.

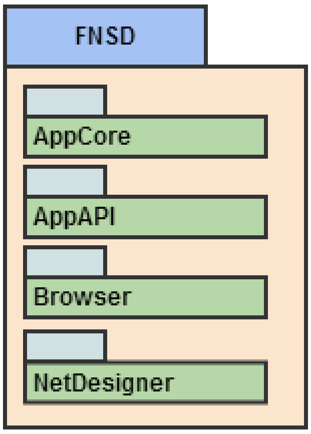


Figure 4.2: FNSD modules

- **Appcore** - module contains core packages of the application, such as basic application actions and ribbon layout.
- **AppAPI** - contains application domain model and api interfaces implemented by other modules
- **Browser** - contains implementation of the tree panel for browsing the assembly plans, scenes etc.
- **NetDesigner** - module contains component for displaying and designing of an assembly plan.

4.4 Domain model

The domain has been created based on the Core Ontology and Scene Ontology. The first Figure[4.3] shows a simplified overview of the first part of the domain model based on the Core Ontology, which represents product assembly plan. Designer mode mainly uses this part of the domain for creating product's assembly plans, which consist of tasks and their mutual dependencies, together with required input resources a resulting output resources. Figure[C.1] shows the second part of the domain model, which is based on Scene Ontology. Scene Ontology is an extension of the Core Ontology. It represents assignment of the product to the workstations and holds results of the scheduling process, job's schedule, assigned workers and their availabilities. It also holds workstation's flow and their resource pools.

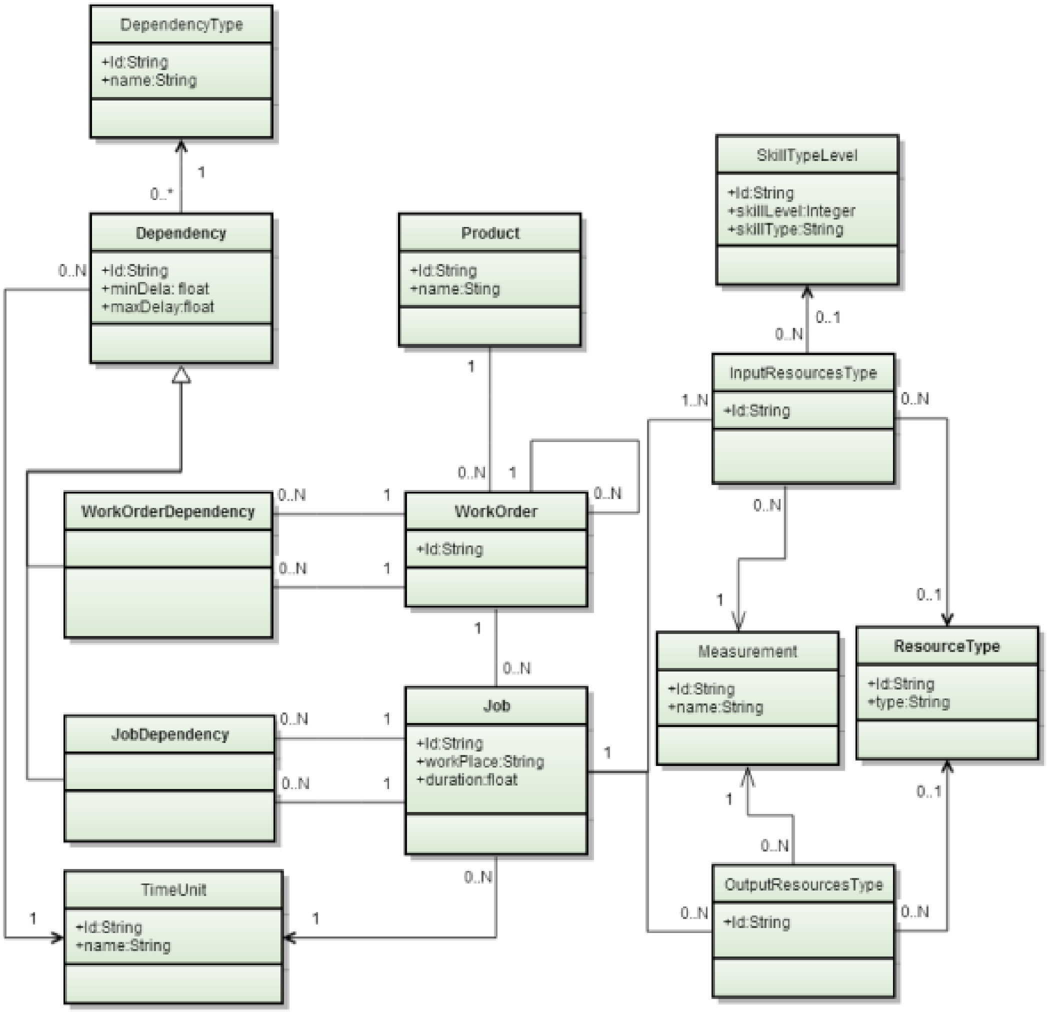


Figure 4.3: Core domain digram

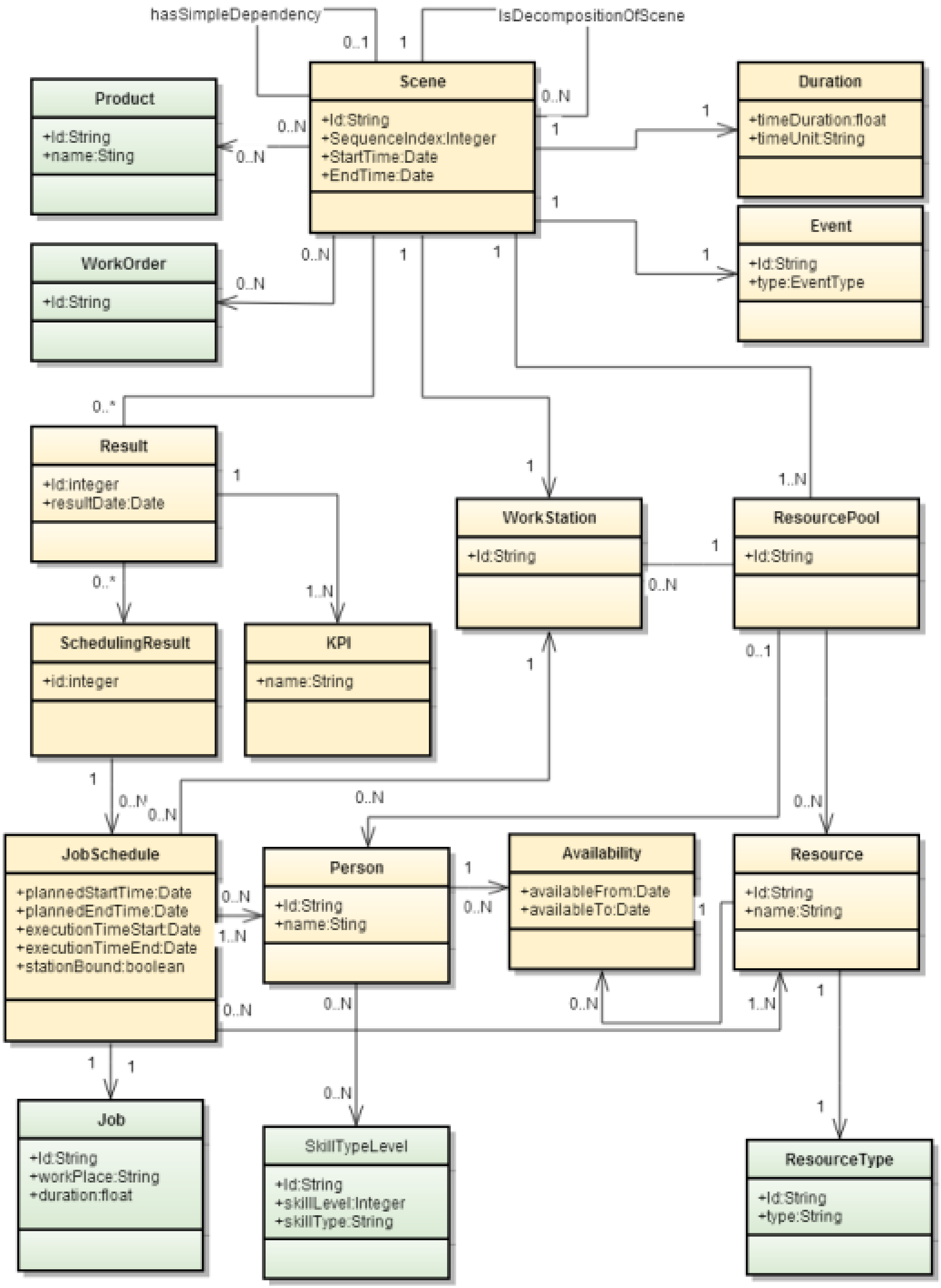


Figure 4.4: Scene extension of the Core domain digram

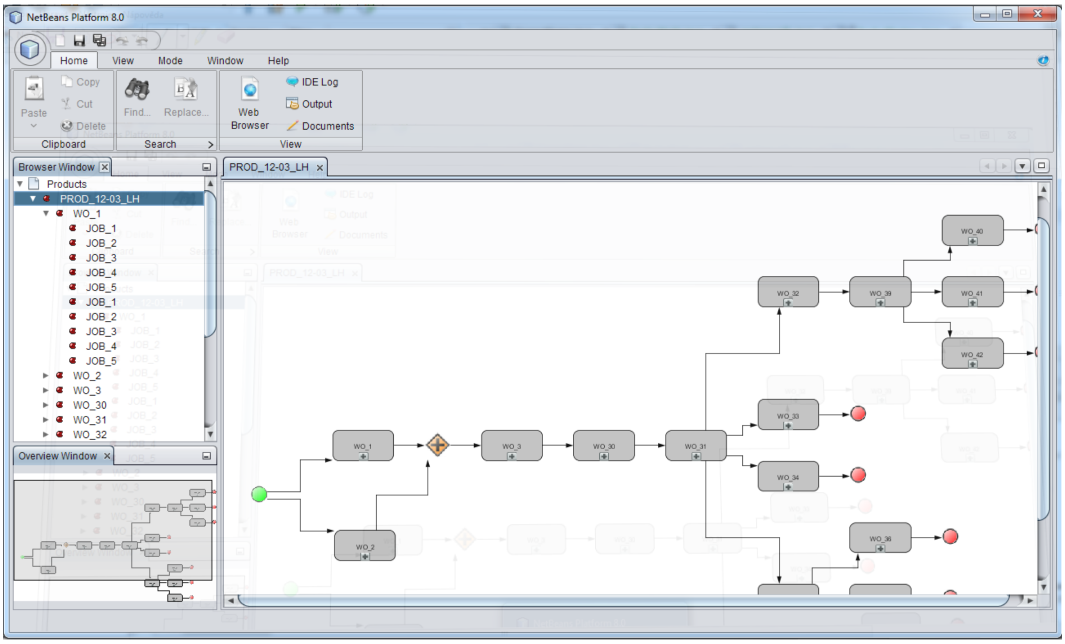


Figure 4.5: Screenshot of the FNSD implementation

Chapter 5

Testing

Application is now in an early state of development. A low-fidelity prototype covering the airbus use-case has been created for the testing purposes. This prototype has been discussed with our European partners from AIB and presented to potential future users within the two meetings in Hamburg. Objectives and results of the both meetings are presented in following sections.

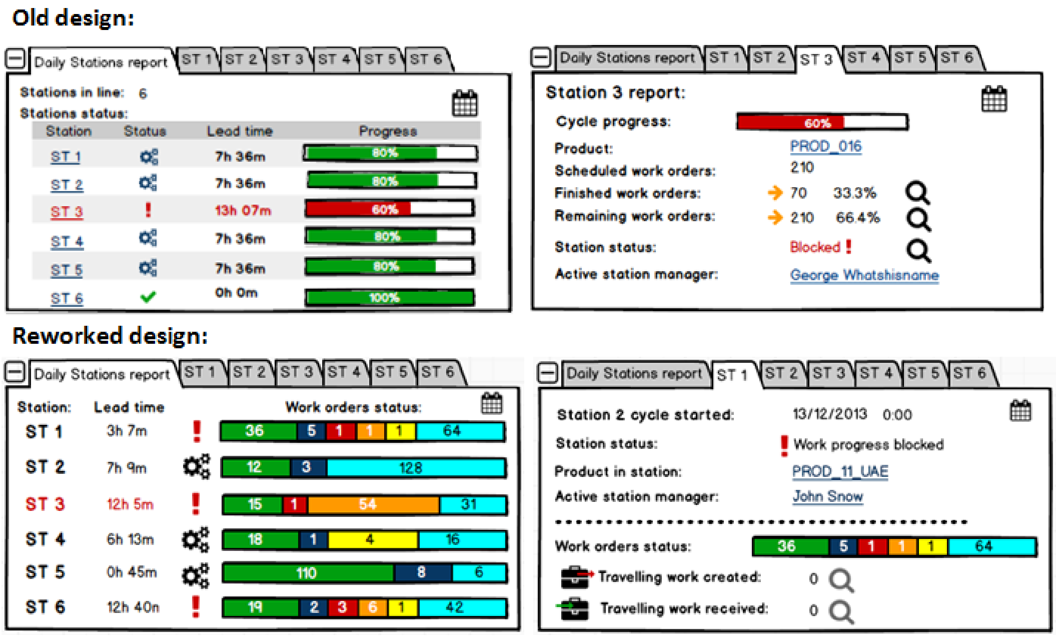
5.1 The first Airbus meeting in Hamburk

5.1.1 Objectives of the meeting

- Introduction of the main concepts of the scheduler in a presentation
- Definition of user roles (station manager, workers)
- Prototype walkthrough with an immediate response from AIB.
- Possible scenario modifications clarification of project demo (demo tuning)
- plan the first usability test with end-users for the next meeting
- Open questions clarifications

5.1.2 Results of the the user feedback

The main result of the user feedback was gathering of several new requirements mainly regarding the Production mode. All valid ones have already been added to the requirement's list described in the analysis chapter of this thesis. Major changes have been done on widgets of the production mode. As an example of design changes made, can be a complete redesign of the Stations report widget shown in the Figure[5.1]. The second, also important result of the meeting was that we made sure the design of the application is going the right way.



Old design:

Daily Stations report | ST 1 | ST 2 | ST 3 | ST 4 | ST 5 | ST 6

Stations in line: 6

Stations status:

Station	Status	Lead time	Progress
ST 1	⚙	7h 36m	80%
ST 2	⚙	7h 36m	80%
ST 3	!	13h 07m	60%
ST 4	⚙	7h 36m	80%
ST 5	⚙	7h 36m	80%
ST 6	✓	0h 0m	100%

Daily Stations report | ST 1 | ST 2 | ST 3 | ST 4 | ST 5 | ST 6

Station 3 report:

Cycle progress: 60%

Product: PROD_016

Scheduled work orders: 210

Finished work orders: → 70 33.3%

Remaining work orders: → 210 66.4%

Station status: Blocked !

Active station manager: George Whatshisname

Reworked design:

Daily Stations report | ST 1 | ST 2 | ST 3 | ST 4 | ST 5 | ST 6

Station:	Lead time		Work orders status:
ST 1	3h 7m	!	36 | 5 | 1 | 1 | 1 | 64
ST 2	7h 9m	⚙	12 | 3 | 128
ST 3	12h 5m	!	15 | 1 | 54 | 31
ST 4	6h 13m	⚙	18 | 1 | 4 | 16
ST 5	0h 45m	⚙	110 | 8 | 6
ST 6	12h 40n	!	19 | 2 | 3 | 6 | 1 | 42

Daily Stations report | ST 1 | ST 2 | ST 3 | ST 4 | ST 5 | ST 6

Station 2 cycle started: 13/12/2013 0:00

Station status: ! Work progress blocked

Product in station: PROD_11_UAE

Active station manager: John Snow

Work orders status: 36 | 5 | 1 | 1 | 1 | 64

Travelling work created: 0

Travelling work received: 0

Figure 5.1: Redesign of the Stations report widget

5.2 The second end-user meeting in Hamburk

5.2.1 Objectives of the meeting

The purpose of the user interface usability testing in Hamburg was to obtain user feedback on the developed application graphical interface (so far as an interactive prototype). The feedback was obtained from several potential future users and experts who represent primarily people working and organizing work at the assembly stations (Station managers, Production steering). A prepared testing scenario was presented to the each user and each user was asked to provide his opinion regarding the interface usability. The testing scenario, as well as the list of asked questions, are attached in the appendix of this thesis[B].

5.2.2 Results of the the user feedback

The results of the second meeting in the Hamburg can be summarized as follows. Some minor bugs in the prototype were found regarding the work orders, jobs or product naming and their labeling. These bugs were caused because of missing data and don't relate the prototype itself.

Additional user requirements were gathered during the testing, but some of them are strongly related to the Airbus production process and probably can't be used in Iacobucci case. An example of such Airbus related only requirement is to display work orders requiring level two and level three inspection, since the Iacobucci production process does not have leveled inspections. Solution to this could be a creation an extension module strictly related to each use-case, which would cover these specific requirements.

Some of the requirements would be nice to have, but could be potentially hard to implement. For example, the user was missing information about issues, which might become a traveling work after the station manager evaluates them. Problem is that it can be rather difficult for the system to determinate correctly if the work order affected by an event is suitable for creating the traveling work or not. It might require developing a potentially sophisticated reasoning system, therefore, this requirement will not be probably fulfilled in first or even in the second release of the application.

Finally, there were some valuable requirements, such as the user was missing information about work orders that are impacting (or could impact) the next station due to some disruptive event. Such information could be crucial for the production progress and can be easily obtained from the schedule. This requirement should be added to the requirement's list for the second application release.

Overall the users were satisfied with the presented solution and looked forward the next meeting.

Chapter 6

Conclusion

6.1 The thesis summarization

The Aim of this thesis was to design and implement managerial tool for operative monitoring of manufacturing processes, integrated to the platform ARUM. The ARUM project, as well as the ARUM platform architecture, has been described in Analysis chapter. ARUM user roles have been presented together with functional and nonfunctional requirements on the managerial tool.

The thesis guidelines declared that the tool will consists of the four modules. However, during the design phase of the tool has been decided, that first and second module from the guidelines will be merged into the one module called the Scenario Designer, which will be developed separately and will be described in the bachelor thesis of my colleague Luis Moreno.[15]

The other two modules the Designer mode and the Production mode has been described in the Design Chapter.

Design in the form of the low-fidelity prototype of those two modes, covering the Airbus use-case, was tested with our partners during the two meetings in Hamburg. Results are summarized in the Chapter Testing.

We choose the Netbeans Rich Client Platform, for the implementation. The description of the tool architecture together with its integration to the ARUM platform has been described. As a user manual serves the ARUM Delivery document D5.2.1[1] and contextual help in developed application.

ARUM platform development is still in an early stage and not all of its key services are ready to use. Also, the Production mode design will be probably modified, based on the feedbacks from Iacobucci. So the current implementation is focused on the Designer mode, mainly on the assembly plan designer component, which currently supports plan visualization and editing. The implementation will continue, so the tool would contain all described functionalities and be fully integrated into the ARUM platform by the second planned release D5.2.2[1] in November.

6.2 Future work

The future work on the tool will mainly focus on its implementation and integration to the ARUM platform. More requirements or design changes will be gathered based on the Iaccobuchi feedback on the low-fidelity prototype. But it should be only minor changes.

Also, a new mode called the Strategic planner will be added, which will be used for production long term planning.

The Production mode could be extended by one more view, displaying something with the working name "production matrix." This matrix could display all work stations in an assembly line and products they currently process, processed and will process. It could also provide functionality such as moving products between stations, see the content of all workstations at a given time (present, past, future) or display the impact of fault events affecting more than one workstation. Further research in this matter is needed.

Bibliography

[1] ARUM project Deliverables. http://arum-project.eu/index.php/en/results/deliverables, 2014.

[2] Balsamiq Mockups. http://balsamiq.com/products/mockups/, 2014.

[3] Factories of the Future. http://goo.gl/coa5gM, 2014.

[4] Seventh Framework Programme (FP7). http://cordis.europa.eu/fp7/home.html, 2014.

[5] NetBeans and Eclipse RCP comparison. http://goo.gl/OYNGjX, 2014.

[6] NetBeans Platform Ribbon Bar Integration Tutorial. https://platform.netbeans.org/tutorials/nbm-ribbonbar.html, 2014.

[7] ALEXANDER BIELE, E. I. W. D2.1.1 – Use-case Definition (use-cases 1 and 2). http://ec.europa.eu, 2013.

[8] ARLOW, N. I. J. *UML 2 And the unified process: Practical object-oriented analysis and design, 2nd Edition.* Pearson Education, Inc., 2007. ISBN 0321321278.

[9] BCK, H. *The Definitive Guide to NetBeans™ Platform 7.* The expert's voice in Java. Apress, 2011. ISBN 9781430241027.

[10] EGGER, F. N. Lo-Fi vs. Hi-Fi Prototyping: how real does the real thing have to be?, 2014.

[11] INDEN, U. et al. Towards an Ontology for Small Series Production. In MAříK, V. – LASTRA, J. – SKOBELEV, P. (Ed.) *Industrial Applications of Holonic and Multi-Agent Systems*, 8062 / *Lecture Notes in Computer Science.* Springer Berlin Heidelberg, 2013. s. 128–139. doi: 10.1007/978-3-642-40090-2_12. ISBN 978-3-642-40089-6.

[12] KARTSON, D. et al. *Modelling with Generalized Stochastic Petri Nets.* New York, NY, USA : John Wiley & Sons, Inc., 1st edition, 1994. ISBN 0471930598.

[13] MARIN, C. et al. A Conceptual Architecture Based on Intelligent Services for Manufacturing Support Systems. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, s. 4749–4754, Oct 2013. doi: 10.1109/SMC.2013.808.

[14] MATHIA, P. Tool for decision support in production planning and scheduling. Bachelor's thesis, CVUT FEL, 2014.

[15] MORENO, L. Tool for production processes operational monitoring. Bachelor's thesis, CVUT FEL, 2014.

[16] OBJECT MANAGEMENT GROUP, I. Business Process Model and Notation (BPMN). http://www.omg.org/spec/BPMN/2.0.2/, 2014.

[17] SOMMERVILLE, I. – SAWYER, P. *Requirements Engineering: A Good Practice Guide*. New York, NY, USA : John Wiley & Sons, Inc., 1st edition, 1997. ISBN 0471974447.

[18] WIEGERS, K. E. *Software Requirements*. O'Reilly Media, Inc., 2009. ISBN 9780735618794.

Appendix A

Glossary

AIB	-	Airbus
ARUM	-	Adaptive Production Management (Name of the European Union project)
DEP	-	dependency
EU	-	European Union
ESB	-	Enterprise Service Bus
FNSD	-	Factory Network Scenario Designer
ICT	-	Information and communications technology
IT	-	Information technology
IHF	-	Iacobucci
KPI	-	Key Performance Indicator
MAS	-	Multi-agent systems
MR	-	Missing Resource events
NC	-	Non conformity
ND	-	Network designer
RAO	-	Operative Area Responsible
RAT	-	rationality
Ramp-up stage	-	Phase of transition between product development and maximum capacity utilization.
RCP	-	Rich Client Platform
RDF	-	Resource Description Framework
REST	-	Representational State Transfer
SD	-	Scenario Designer
Task	-	Generic class of Job or Work order
Technological process	-	equivalent to term assembly plan
Traveling work	-	unplanned transfer of job to another station
UI	-	user interface
WBS	-	Work breakdown structure

Appendix B

End-user meeting

B.1 List of questions

The user was asked to answer the following questions:

- Is the information presented easy to understand?
- Does the presented information match your currently used formats/forms?
- Are you missing some additional information in the presented view/component?
- Do you know what individual items represent and what do you expect the controls would do?
- Did the controls do what you expected?
- Does the tool cover your needs?
- Is the accessibility of the user interface satisfactory?

B.2 Testing scenario

A station manager (you) comes to the morning shift and wants to review the performance of the currently running production at his/her station(s).

Dashboard

1. Identify the KPIs of the production
   - What is the percentage of finished/remaining work orders/processes?
   - Is the production delayed?
2. What is the current situation at your stations?

   - What is the current progress of each station?
   - Which station is blocked? Why?

3. How many blocking events there are?
   - How many of them are new?
4. What is the current situation regarding the shifts?
   - How many workers are available at ST 90 for shift 1?
   - How many workers are required at ST 88 for shift 2?

Assembly plan/Gantt chart

5. Which work orders / processes are now running?
6. Which work orders / processes are finished?
7. Which work orders / processes are in trouble?
   - Why?
8. Which processes (work orders) are being done at my station that did belong to other sta-tions?
9. Which processes (work orders) originally planned to be done at my station are delayed and transferred to other stations?
10. Display the critical path

Resources

11. What workers should be in the current shift for ST 90?
12. What workers are currently working?
13. What workers are available?
14. What work orders / processes are assigned to worker Jackson Whitledge?

Events

A new event occurs – reported by one of the workers.

15. Identify what happened – what type of event occurred
16. Filter the events – display only the blocking events
17. Browse details of a blocking event
   - Who reported the event?
   - On which station the event occurred?
   - When was the event reported?

- Who is currently responsible for solving the event?

Rescheduling

You have decided to create a new schedule, based on the current situation.

18. Run the rescheduling

19. Setup the scheduling parameters

    - Setup the scheduling strategy: traveling work
    - Setup the preferred optimization: corner date
    - Setup the scheduling time range: current day

20. Display the results

21. How would you compare the KPIs of the results?

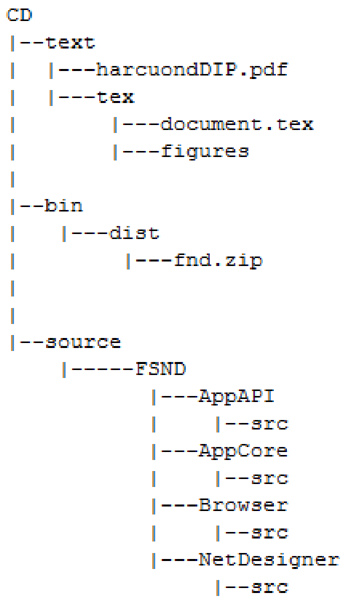
22. What is the best result regarding the maximal throughput?

23. Compare two Gantt charts

24. Apply your selected schedule to the current production

Appendix C

Content of attached CD



```
CD
|--text
|  |---harcuondDIP.pdf
|  |---tex
|       |---document.tex
|       |---figures
|
|--bin
|   |---dist
|        |---fnd.zip
|
|
|--source
    |-----FSND
           |---AppAPI
           |    |--src
           |---AppCore
           |    |--src
           |---Browser
           |    |--src
           |---NetDesigner
                |--src
```

Figure C.1: Content of attached CD

- **text** - pdf and tex sources of this thesis
- **bin** - runnable NB apllication distribution
- **source** - source codes of the application