FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Persona Adaptable Visualization Scheduling in Supply Chain Management for an ERP System

Antero Guimarães Pacheco da Silva

Report of Project/Dissertation
Master in Informatics and Computing Engineering

Supervisor: Luis Paulo Gonçalves dos Reis (Auxiliar Professor)

17th July, 2008

Persona Adaptable Visualization Scheduling in Supply Chain Management for an ERP System

Antero Guimarães Pacheco da Silva

Report of Project/Dissertation

Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: Doutor Raúl Fernando de Almeida Moreira Vidal (Professor Associado da FEUP)

External Examiner: Doutor José Manuel de Castro Torres (Professor Auxiliar da Universidade Fernando Pessoa)

Internal Examiner: Doutor Luís Laulo Gonçalves dos Reis (Professor Auxiliar da FEUP)

17st July, 2009

Abstract

The technological evolution over the past century has spread to every aspect of our daily life; in the business management world the development of *Enterprise Resource Planning* (ERP) system has also revolutionized the way companies perceive and manage their businesses. The integration and centralization of all the departments' data within a company has increased efficiency of processes, making it possible to detect behavioral patterns from customers and suppliers, and bringing companies closer to the market's needs, resulting in increased profits.

The core investment of ERP development companies has been towards *Business Inteligence*. This field's goal is to analyze data gathered from all over the system, locating problems or finding new ways to improve certain processes.

Another field of research in ERPs is the enhancement of graphical user interfaces, through charts that aid in visualizing the enormous amount of the system's data. Although there has been a constant investment in visualization solutions, this is a field where the first steps are being taken.

The development of a visualization system for aiding the user's planning, relating orders delays with specific business concepts, would represent an asset for ERP systems.

Microsoft, market leader in operating systems' development, has a solid presence in the ERP development for the small and mid-sized companies, through the Microsoft Dynamics AX software.

This project, developed within the *Dynamics AX Manufacturing* department, aims to analyze the current state of the art status in the visualization field, also looking into other ERPs' approaches in this area. This research will be the basis for the development of a tool that aids in *Manufacturing* planning, focusing in *Production Planning* and problem solving by lack of resources.

Relying on highly developed human cognitive abilities, capable of visually absorbing several different concepts at once, this system will provide the user with new perspectives, being one step ahead of the current table views. It will integrate fundamental planning aspects into one single view allowing the user to relate order delays with the lack or overload of business' resources.

The details and concepts of *Production Planning* within *Manufacturing* will be outlined, following the results of the visualization research. The project will define solutions to solve the problems at hand, presenting justifications for the chosen methodologies. Due to the technical nature of this thesis, the software's architecture, design and implementation details are also described, along with the technologies used in its development, followed by a demonstration on how the developed software aids *Production Planning*.

This thesis final chapter will provide conclusions over the success of the developed software, based on the outlined requirements, also stating possible improvements, or other

fields where the same methods could be applied.

Resumo

A evolução tecnológica do último século faz-se sentir em todas as vertentes do quotidiano, sendo que na área de gestão empresarial o desenvolvimento dos sistemas de *Enterprise Resource Management* revolucionou a forma como as empresas vêem e gerem os seus negócios. A integração e centralização dos diversos sectores de um empresa, tornou possível aumentar a eficiência dos processos assim como detectar padrões comportamentais de clientes e fornecedores, permitindo uma maior aproximação às necessidades dos mercados, resultando no aumento dos lucros.

O investimento principal das empresas de desenvolvimento de sistemas ERP tem-se direccionado para a *Business Intelligence*. Esta área tem como finalidade analisar os dados recolhidos de todas os componentes do sistema, localizando possíveis problemas, ou sectores que possam ser melhorados.

Outra das vertentes de investigação em ERPs tem sido a melhoria dos interfaces disponibilizados ao utilizador, através de gráficos que auxiliem a visualização da imensidão de dados do sistema. Embora tenha havido um constante investimento em soluções de visualização, é uma área onde ainda se estão a dar os primeiros passos.

O desenvolvimento de um sistema de visualização que auxilie o utilizador a efectuar o seu planeamento, relacionando atrasos de encomendas com conceitos específicos de negócio, representaria uma mais-valia para os sistemas ERP.

A Microsoft, líder de mercado no desenvolvimento de sistemas operativos, tem uma presença sólida no desenvolvimento de ERPs para pequenas e médias empresas, através do software Microsoft Dynamics AX.

Este projecto, inserido no departamento de *Dynamics AX Manufacturing*, visa analisar as tecnologias de topo existentes na área de visualização, aliado a uma pesquisa de mercado aos sistemas existentes noutros sistemas ERP. A investigação servirá de base para o desenvolvimento de uma ferramenta, que possa auxiliar o sector de *Manufacturing*, focando-se no planeamento de producão e resolução de problemas por falta de recursos (capacidade ou material).

O software proporcionará novas perspectivas ao utilizador, estando um passo à frente dos actuais sistemas tabelares. Baseando-se nas elevadas capacidades cognitivas de absorção de diferentes conceitos visuais de uma vez só, do ser humano, integrará os aspectos fundamentais do planeamento numa vista só permitindo ao utilizador relacionar o atraso de encomendas com sobrecarga ou falta de recursos no negócio.

Serão apresentados em detalhe os conceitos de *Planeamento de Produção* em *Manufacturing*, seguido dos resultados da investigação em visualização. São posteriormente definidas soluções para a resolução do problema em questão, apresentando uma justificação para a sua escolha. Devido à natureza tecnoloógica desta tese, são apresentados detalhes de arquitectura, design e implementação do software, assim como as tecnologias utilizadas para o seu desenvolvimento. Segue-se uma demonstração de como o software desenvolvido auxilia o *Planeamento da Produção*.

A parte final desta tese, retira conclusões acerca do sucesso do software em relação aos requisitos delineados, assim como, possíveis melhorias, ou áreas nas quais as mesmas metodologias poderiam ser aplicadas.

Acknowledgements

The acknowledgment has to start in my studying days; in my household there was always a strong influence to aim for the top. Looking back to those days I need to thank my parents and sister's support throughout the several difficult stages of my life, without them none of it would be possible. Especially my Dad for putting me into sports, music, and english at an early age which definitely provided a strong basis for building character and turning me into what I am today.

I would like to thank all my teachers from the first to the twelfth grade for never giving up on a student that enjoyed, maybe too much, playing football during the recess and not focus on classes.

The breakthrough in my life was when I joined my University, and the exact degree I aimed for, after one week of studying 15 hours a day for the Mathematics and Physics exams, which was deeply supported by my sister, to whom I also have to thank for maintaining me on track every time I thought studies were becoming impossible.

I would like to thank my University friends, for the support during my studies, for the long nights working at the University, for the coffees, for the Red Bull, for the opportunity to grow.

I also have to thank my family's support, when in September 2007, I was given the opportunity to take one year off from my studies and go to Ireland for one year and work in Sun Microsystems.

I have to thank all my close friends for helping me throughout my life, their support during rough times kept me going, thank you for the good times, for the dinners, for the music festivals, for the trips around Europe, for the long debates on politics or football, for the moments, for existing and being there.

I would also like to thank my project coordinator Luís Paulo Reis, and Hans Jorgen Skovgaard that interviewed me in Portugal and gave me the opportunity to join Microsoft. After my arrival I received much support from the Manufacturing team, specially Alexey Ovsyannikov for his support throughout the project. A special thanks to Jens Olesen Lund for the brain storming sessions, filling me in about several business details, making it possible for me, to determine what were the important aspects of *Manufacturing*.

In the beginning of this project a mailing list was created with several people that were finishing the project this semester, a special thanks to the VMT guys, for making it easy to sit in front of the computer for endless hours.

I started by thanking three important people in my life, and will end, by thanking them again, my sister, Joana Guimarães Pacheco da Silva, my Mother, Zália Manuela Lemos Guimarães Pacheco da Silva, and my Father, Antero de Magalhães Pacheco da Silva.

Antero Guimarães Pacheco da Silva

Contents

List of Figures

List of Tables

Abbreviations

2D	Two Dimensions
3D	Three Dimensions
API	Application Programming Interface
APS	Advanced Planning System
ASP	Active Server Pages
BOM	Bill Of Materials
CLI	Common Language Infrastructure
CLR	Common Language Runtime
CLS	Common Language Specification
COM	Component Object Model
CPU	Central Processing Unit
CRM	Customer Relationship Management
CRP	Capacity Requirements Planning
CTS	Common Type System
DBMS	Database Management System
DPI	Dots-Per-Inch
DRP	Distribution Requirements Planning
ERP	Enterprise Resource Planning
GC	Garbage Collector
GDI	Graphics Device Interface Plus
GDI+	Graphics Device Interface Plus
GUI	Graphical User Interface
IDE	Integrated Development Environment
IBM	International Business Machines Corporation
JIT	Just-In-Time
KPI	Key Performance Indicators
LINQ	Language Integrated Query
MDCC	Microsoft Development Center Copenhagen
MFC	Microsoft Foundations Class
MITS	Micro Instrumentation and Telemetry Systems
MPR	Materials Requirements Planning
MPC	Manufacturing Planning and Control
MPS	Master Production Schedule
MRP	Materials Requirements Planning
MSDN	Microsoft Developer Network
MSIL	Microsoft Intermediate Language
OS	Operating System
PHP	Personal Home Page
PLM	Product Lifecycle Management
RSS	Really Simple Syndication

ABBREVIATIONS

SAP	Systems Applications and Products
SCM	Supply Chain Management
scRGB	wide color gamut RGB
SOP	Sales and Operations Planning
USA	United States of America
UX	User Experience
VB	Visual Basic
WinForms	Windows Forms
WPF	Windows Presentation Foundation
XAML	Extensible Application Markup Language

Chapter 1

Introduction

"Whenever you are asked if you can do a job, tell them, 'Certainly I can!' Then get busy and find out how to do it.."

Theodore Roosevelt

The goal of this thesis is to provide innovative approaches on visualization, based on the ERP logic of Microsoft Dynamics AX, by portraying several scheduling and planning overviews, allowing bottleneck detection and providing some level of interaction, with the intent of allowing users to optimize company processes and increase efficiency. Due to these goals this project will be referred to as *Interactive Visualization* project.

This thesis fundamentals is to research the state of the art in visualization techniques, providing different views over the latest developments in representing large amounts of aggregated data. A requirement is a functionality which allows users to transverse from the top level of aggregated data, down to the smallest detail, i.e. a system that allows the user to adjust the data to what is required for solving a determined problem.

This project was developed in Microsoft Development Center in Copenhagen, and is the result of eighteen weeks of research and development.

This chapter contextualizes the project within the company and the ERP Manufacturing business, followed by presenting its motivations and objectives.

1.1 Microsoft

1.1.1 About

Microsoft Corporation is an American multinational computer technology company, leader in the development of operating systems. Based off Redmond, Washington, USA, Microsoft was founded in April 4th 1975 by William Henry ('Bill') Gates III and Paul Garner Allen in Albuquerque, New Mexico. When the Altair 8800 microcomputer was launched by MITS (Micro Instrumentation and Telemetry Systems) in 1975, Bill Gates and Paul Allen approached the manufacturers with the idea to create a programming language. After a successful demonstration of the product, the BASIC operating language was deployed on the platform.

In 1980, Microsoft released their first operating system, the Xenix, based on the Unix operating system. The first release of Microsoft's own operating system followed, the MS-DOS, which was still based on a text console, providing ease of use to users all around the world. The first approach to the graphical interface operating systems was launched in 1985, the Microsoft Windows 1.0; running on top of MS-DOS it extended its functionalities and provided the user interaction by the use of a Mouse.

The consolidation of Microsoft as a market leader came with the release of Windows 3.0 in 1990, followed by Windows 3.1, NT, 95/98, ME, 2000, XP and Vista. Their simplicity, user-friendliness, widespread distribution and ability to run on almost every genre of CPU, put Microsoft's name on top of the operating system business [6] [7].

1.1.2 Microsoft Dynamics

Microsoft Dynamics is a group of software business management solutions, which facilitates decision making by reducing risks and allows companies to integrate their entire business model in one application. Dynamics solutions include applications and services for retailers, manufacturers, wholesale, distributors, and service companies, doing business domestically or in multiple countries.

The main focus is on three critical areas:

- Financial Management
  - Microsoft Dynamics AX
  - Microsoft Dynamics GP
  - Microsoft Dynamics NAV
  - Microsoft Dynamics SL
  - Microsoft Small Business Financials
- Customer Relationship Management (CRM)

  - Microsoft Dynamics CRM
  - Microsoft Dynamics AX for sales and marketing
  - Microsoft Dynamics GP for field service
  - Microsoft Dynamics NAV for sales and marketing
  - Microsoft Dynamics NAV for service management
  - Microsoft Dynamics SL for field service

- Supply Chain Management (SCM)
  - Microsoft Dynamics AX
  - Microsoft Dynamics NAV
  - Microsoft Dynamics GP
  - Microsoft Dynamics SL

Dynamics AX and NAV are the broadest solutions overlapping CRM and SCM. These software bundles have everything to compete for a share of the market with softwares from the largest companies in the business, Oracle and SAP. One of the advantages of Dynamics is that all tools are fully integrated, not sold in different customizable software packages; furthermore, although everything comes within the same package it is still modular and customizable.

1.1.3 Microsoft Development Center Copenhagen (MDCC)

MDCC emerged after Microsoft acquired Navision-Daamgard, a Danish company focused in Enterprise Resource Planning (ERP) solutions (later renamed Microsoft Dynamics AX and Microsoft Dynamics NAV) development. It is currently the largest development center in Europe and outside the United States, employing 650 people from all over the world. As to be expected from one of the largest companies in the world, MDCC has meets the workers' every need, providing not only excellent working conditions, but also organizing events and other leisure activities [8].

1.1.4 Manufacturing Department

This project was developed for Microsoft Dynamics AX's Manufacturing Department at MDCC, with the collaboration of Microsoft Dynamics User Experience (UX) experts.

The Manufacturing Department is responsible for integrating Dynamics AX with all logic associated with the manufacturing management process, integrating the latest ERP concepts.

Dynamics UX is a department at MDCC, which sole purpose is to improve the aspects of visualization in Microsoft Dynamics techonologies. This is a very broad area of development, from new ways of presenting data to the user, down to how an icon looks on the screen or if a label correctly named.

1.1.5 Microsoft Dynamics AX

The nature of the Enterprise Resource Planning business, relies on integrating several systems within one manageable platform, allowing users interaction with every aspect of their business in one tool, simplifying work flows and paperwork, providing common data, common procedures, and real-time data availability to facilitate decision making in globally dispersed organizations. This integration comes at a cost; the software must be highly customizable so that the user is able to model their business around it, increasing the tool's development complexity and its price. ERP systems are very expensive and hard to integrate into a working business, taking a few years to fully model and adjust.

Dynamics AX is an ERP business management solution highly customizable and complex. Originally named Axapta, it had been developed from 1983 by Damgaard Data A/S; in 2000 the company was merged with Navision Software A/S to form Navision-Daamgard. By 2002 the company had reached a size and revenue level that interested Microsoft, allowing it to enter the ERP market business. Dynamics AX's goal is to increase productivity efficiency and easily standardize processes across multiple locations and countries, simplifying the visibility of the business as a whole by providing consolidated reports and feedback. Unlike most high-end ERPs, Dynamics AX targets not only big companies, but also small businesses that are trying to get an edge on the market [9].

1.2 Manufacturing Planning and Control

Due to this project's focus on scheduling and planning within manufacturing, this section is seen as an introduction to required concepts, allowing the reader to understand the problem at hand.

The traditional manufacturing company model no longer exists; in today's global markets and fast paced society, even the smallest manufacturing company has customers in different countries, making it important to keep up-to-date on every area and follow up on trends, adapting to the market's needs.

The evolution of *Manufacturing Planning and Control* (MPC) has lead ERPs systems to integrate a solution into their software packages, attempting to integrate every business area into one complex but manageable system. The MPC system's main goals are to plan and control all aspects of manufacturing, including managing material, suppliers, customers and scheduling.

Before business globalization, a manufacturing company would typically own one or more factories, located near their target markets; in today's global economy, manufacturers have a different approach, relying on complex supply chains, with suppliers and customers around the world. The businesses have shifted their main focus to product design and marketing, outsourcing the manufacturing aspect of the business to third-parties, taking advantage of the specialization and low manufacturing costs regardless of the geographical location. This means that a product may have some parts developed in Taiwan, others in China, be assembled in Greece and shipped for the whole European Market, providing the developing company the ability to plug into any supplier around the globe.

With the growth of the global market concept and with the rise in competition between companies, the need to innovate and provide a better customer service arises, making it important not only to reduce costs but also improve delivery time response. Customization levels have increased, and although some products have the same basis, the final stage of the manufacturing process sees an adjustment for the target market.

The integration of all components in this complex chain lead to the development of the MPC concept, in order to manage the supply chain or the supply network systems. With the improvement of ERPs, it became an essential tool to gather all aspects of the MPC business, providing integration with other areas of the company's business and also with geographically dispersed and culturally different organizational units [10].

1.2.1 Planning

The complexity of MPC demands rigorous planning, following strict constrains and complex strategies to keep everything in place and make sure demand and schedules are delivered on-time. There are several planning strategies and steps which must be considered, which are explained in this chapter.

1.2.1.1 Sales and Operations Planning

In the dawn of manufacturing and sales the customer would seek a product from a certain company that would fulfill a given need, but nowadays the model has switched - companies now try to fulfill customers' needs. In this sense, *Sales and Operations Planning* (SOP) is an important aspect of any successful business, trying to balance demand and supply, while considering their volume and mix. This plan links strategic production goals with other business plans, such as marketing planning, financial planning, operations planning and so forth.

In sales planning it becomes obvious that there is a need to determine the market's needs, fulfilling demand. When this principle is not met and demand exceeds supply, service levels lower and the company may endure some trust issues with its customers. On the other hand, if supply exceeds demand other problems may emerge, inventories increase, possibly resulting in workforce and products' value reduction. Therefore, continuous planning becomes important, in order to maintain a good balance between supply and demand. The other two variables in the system are volume and mix; the first concerns the decision about how much of each available product families should be manufactured in a given time period, while the latter details which products, within each family, are going to be manufactured and in what sequence.

The operations section of the SOP defines the mission which manufacturing must accomplish to meet the company's strategic objectives; detailed planning, expressed in meaningful and aggregated units, easy to understand, providing guidelines for other planning stages. It is not a forecast; it clearly defines what is to be produced, at a high level, in every product family. It also assures the company's strategies are to be met, for example, if a sales forecast provides information on a certain product's demand, it may be a company strategy to keep supply much lower than demand, to keep the company's image of exclusivity and high prices in their products. The operations plan will also state strategies and guidelines for demand peaks and product seasonality.

Sales and Operations Planning is usually seen as the link between top management and manufacturing, between the business level and the actual production, providing visibility of the critical interactions between sales, marketing, production and finance [10].

1.2.1.2 Master Production Scheduling

The *Master Production Schedule* (MPS) is the next step between the SOP to manufacturing, stating what is planned for the future. It relies on several inputs: forecast demand, production costs, inventory costs and levels, customer orders, supply, economic lot size, production lead time, capacity limitations, other resource considerations and the sales and operations plan. It transforms the SOP from its aggregated and strategic levels into a scheduled plan of how products will be supplied to meet demand, specifying products quantities and timing.

The MPS is stated in product specifications known as *Bill Of Materials* (BOM), outlining which parts are needed to produce each product. It measures products in currency (such as dollars or euro), or any other global units which don't relate directly to the product, so the measures are in number of units or number of parts of a product, which may be assembled later.

Due to MPS's nature as a build schedule and considering capacity limitations and other factors, the MPS may specify build quantities that exceed the current demand, in order to meet future order deadlines. If future demand is higher than the available build capacity for the required amount of products, advancing the start of production to meet the needs of the market is sometimes one solution, also taking into consideration the inventory costs.

By specifying the exact production output, in terms of products, dates and quantities, the MPS becomes the core plan for making sure a company will be able to fulfill customer demand. In case there is the need to reduce or increase one product's production, the MPS allows the planners to determine which item's production should be reduced in order to obey capacity limitations [10].

1.2.1.3 Material Requirements Planning

The *Materials Requirements Planning* (MPR) details material planning in the manufacture of parts and their assembly into finished products. It is a plan used by companies that take the batch production process approach on manufacturing and its goal is to make sure the right number of parts are available at the right time, to meet customer demand.

Along with the MPS, which provides information on the number of parts that need to be produced and when, MRP requires two other inputs which allow the system to adjust its production levels: 1) the BOMs details, so its aware of what is required to build product A, B or C, and 2) inventory status and product shelf life time. Other useful requirements are: lot size, safety stock and safety lead time; these three inputs allow MRP to improve the output schedule by considering other variables in its planning.

The MRP system outputs a *Recommended Production Schedule*, which details the schedule to meet demand for the current orders, and a *Recommended Purchase Schedule* which states what items should be purchased and when in order to keep the schedule on the right track. Other outputs are recommendations for cancelling, delaying, increasing or speeding up existing orders [10] [11].

1.2.1.4 Distribution Requirements Planning

The *Distribution Requirements Planning* (DRP) relates current inventory status, forecasts and other predictions about demand, to manufacturing production scheduling and material planning. Through effective understanding of the business, current sales and inventory levels, it tries to predict the demand requirements for production. By reporting updates about sales and forecasting, MPS and MRP may be adjusted to meet the current orders' status, in volume and mix, therefore reducing transportation and inventory costs and keeping the system working at a level that meets demand.

A key to success in a DRP system is to have it set up as close to the marketplace as possible, ideally at key customer locations. By directly integrating with customer systems, value knowledge is acquired, providing feedback to manufacturing on what should be produced. With a continuous link between customer and manufacturing, DRP establishes a more stable and effective supply chain, reducing global costs and increasing service levels [10] [11].

1.2.1.5 Just-In-Time

Just-In-Time (JIT) is a modern approach on material planning and is key to the success of some businesses. Unlike MRP, which relies on replenishing items, inventory levels for production based on demand, trying to make sure stocks are not ruptured, JIT bases its philosophy on minimizing waste in manufacturing by assuring that the materials are only delivered iat the "*exact moment*" they are needed.

The main developer of this concept was *Toyota*, introducing JIT into their supply chain, and thereby reducing lot sizes and inventory levels, while increasing constant replenishment of stock. However, in order to work, the parts' suppliers must have a *no defect* policy, approaching as close to 100% quality assurance as possible; if these levels drop, the chain stops due to lack of parts.

JIT's use reduces complexity over items tracking, because the time between the arrival of parts and their transformation into finished goods very short, removing the need to track their progress in the system and providing a continuous flow to the system, at whatever working level is required to meet demands.

MPR is based on the *push* system concept, which relies on delivering items to other parts of the chain; if they are not required at the time, they will be put into inventory, making balanced inventory levels the key element. This system has some fluctuations because parts may be defective which often requires the supplier to resend new items, resulting in a time variance in supplier delivery date. JIT differs from this approach, and it is known as a *pull* system, which reports to the suppliers its requirements and dates. The key elements of this chain are the suppliers' no defect policy and the assurance that the items will be delivered on schedule [10] [11].

1.2.1.6 Capacity Planning and Utilization

One of the planning problems is to assure capacity constrains are met, while still being able to deliver orders; this is where *Capacity Planning and Utilization* (CPU) comes into place. Another problem is to consider the balance between production and capacity utilization, trying to achieve a level where demand is met and capacity is not underutilized.

Due to the difficulty in capacity balancing, the use of capacity planning techniques is an important aspect of the business. The main goal of these techniques is to accurately estimate capacity requirements, to be able to meet orders' delivery times. Another objective is plan execution, making sure it is executed flawlessly to avoid capacity underload which deteriorates delivery performance, work-in-progress inventories and lowers manufacturing personnel's moral, and also avoid excess capacity which leads to greater expenses.

There are several ways to develop a capacity plan, for long range, mid range and short range planning time periods. While short range planning considers the actual job scheduling for every different resource, mid and long range are based in aggregated time periods and aggregated capacities, and use *Capacity Requirements Planning* (CRP).

CRP takes into consideration several variables for its planning: actual lot sizes, lead times for scheduled orders, current capacity status of all work-in-progress, demand for service parts, bills-of-materials, and routing. CPR uses MRP information to calculate the capacity required to complete the MPS, which is achieved by calculating capacity requirements for open shop orders and planned orders.

The CPR calculations are based on aggregated working times for each work center. For example: if one work center has a working time of 8 hours per day and 5 working days per week, it's week working time is 8 x 5 = 40 hours / week; an order that requires 10 hours of working time from this work center may be scheduled within this week. These calculations only look at aggregated loads and do not consider that for actually scheduling this order there are several constrains. This planning is for long-mid range periods of time, for small periods of time *Job Scheduling* comes into place. In CPR there is the concept of *Finite Capacity*, which means that the company may restrict overbooking in a certain time period. If there is a possible solution, as simple as increasing the number of working shifts, for a work center in this period, it is useful to use infinite capacity. If however the company knows it is unable to cope with even the slightest overbooking level, finite capacity is used for planning.[10]

1.2.2 Supply Chain Management

The *Supply Chain Management* (SCM) is an evolution from the traditional *Manufacturing Planning and Control* system, while MPC is mainly focused on intrafirm integration to an interfirm integration. SCM's goal is to coordinate flows across multiple companies, achieving major improvements by overall coordination. With SCM integration there is need to implement several systems across all companies in the business, achieving a single chain, much like trying to get multiple companies to be integrated into a ”merged company” view.

SCM determines complete support between all working parties in the supply chain, integrating supply needs from customers and reporting them to suppliers. Sometimes a *Vendor Inventory Management* is used, whereby the supplier immediately knows when to replenish customer inventory levels, and takes full responsibility in performing this task [10] [11].

1.3 Visualization

The study of information representation includes several fields: data representation, information graphics, knowledge visualization, scientific visualization and visual design.

Since the introduction of data graphics in the late 1700's, information visualization studies have been seeking to improve how people look at data by relying on the human cognitive capability to visually absorb large amounts of aggregated and abstract data. Besides the interesting aspect of making it possible to squeeze in as much information as possible into one single view, the human brain is also able to visually detect some data patterns and behaviors - something that today is an advanced computer science area, trying to automate what humans have achieved with evolution [12] [13].

Throughout history there has been an evolution in visual representation, from the prehistoric days of cave painting to the high-tech solutions of today. In the past century, viewing data in the form of charts or graphs became usual, displayed in an organized fashion, easy to understand. Today, with the use of computers, designers and engineers have been able to create other complex types of views. The development of *static views* which by the click of a button, animate or adjust to represent the data, transversing through different time periods in an automated way and allowing the user to define the detail level [14] [15].

In an Era where the amounts of new information per year exceeds all historical facts before the twentieth century, it becomes hard to provide aggregated overviews of a certain problem, making data and information visualization one of the main technological problems of this century. Another problem is relating aggregated data with the ability to drill down to the smallest detail.

With the introduction of computer graphics and new technologies, different perspectives on the same data have become available, trying to innovate and advance the visual interaction [16] [17] [13].

1.4 Motivation and Objectives

The key in ERP systems is customization and therefore, adaptable visualizations development is one of today's main investments in every vendor, allowing the user to setup the system according to what his job requires and helping the manager to schedule and plan, by increasing his grasp over how the business is progressing.

The development of a tool, that provides a planning overview and allows the user to filter data, would meet one of the main planning aspects in production planning: which orders are being delivered on-time. After the order lateness is known, the user still needs to find out where the bottleneck resides; providing a view on resource capacity load levels, will facilitate this resolution.

This project's visualization approach is innovative and relevant for Microsoft Dynamics AX evolution in the ERP market, a proof-of-concept of what can be achieved when cutting edge technology and new ideas are incorporated, to an old business, possibly altering perspective, and way of looking at planning in the *Enterprise Resource Planning* business. The objectives of this project are:

1. Research what is the latest in visualization approaches and techniques, not only ERP, but in other areas, attempting to reach an innovative solution for visualization by incorporating different concepts and ideas;
2. Define what are the target *personas* and planning environments of the project;
3. Determine what each *persona* regards as valuable data to be displayed;
4. Design visualization mock-ups to address the *personas*' planning goals;
5. Design an interactive visualization prototype that is innovative, intuitive, appealing and, at the same time, able to resolve most of the personas issues;
6. Integrate the visualization solution with Microsoft's *Job Scheduling Engine* used in Microsoft Dynamics AX;
7. Use Microsoft technologies for implementation, allowing easy integration with the Microsoft Dynamics AX ERP system;

The project's working proceedings follow an iterative development process. After each cycle, feedback is gathered from simple user experience tests regarding new features introduced to the visualization mock-ups and prototypes. The test subjects were mainly people involved in the project from the start, although some random feedback was obtained from co-workers at Microsoft, providing valuable intelligence for taking the next steps.

1.5 Thesis Overview

This document is organized in the following manner:

Chapter 1 - Introduction
Contextualization of the project within Microsoft's environment, introducing the location where it was developed and the target business and software. It continues by presenting concepts and basis to the understanding of the project, its motivations and objectives.

Chapter 2 - Problem Description
A detailed project description, providing a solid understanding over the problem, presenting requirements and what the project aims to achieve. The final section of this chapter

outlines the project schedule and deliverables.

Chapter 3 - State of the Art
A research on the what is cutting edge in visualization, introducing common and innovative techniques, solutions and views, inside and outside the ERP business. It details what aspects should be considered for the project and which are the common techniques.

Chapter 4 - Solution Specification
Based on the problem requirements and the state of the art research, a specification will be detailing how to develop the software. A justification of the proposed solution will be presented, stating why this was the best alternative compared to the rest of the research.

Chapter 5 - Design And Implementation
This chapter describes how the application was designed and implemented, in order to achieve the proposed requirements, solving the problem at hand. It details how specific problems were solved to achieve the overall goal.

Chapter 6 - Evaluation of The Solution
To assess a solution's success it is necessary to evaluate it from several perspectives; proposed functional and non-functional requirements will be compared with the final software. An user experience will also be conducted to assure that the project is what customers want, and could be used in Dynamics AX.

Chapter 7 - Conclusion and Future Work
This final chapter draws conclusions on the project and provides some insight into other features that could be introduced.

Chapter 2

Problem Description

> *"Any problem, no matter how hard it may be, will only remain unsolved from the moment all attempts are abandoned."*
>
> Inácio Dantas

This chapter presents a detailed overview of the problem this project aims to solve and how the project adjusts to the ERP business. It details requirements, objectives, schedule and deliverables.

2.1 Persona Adaptable Visualization Scheduling in Supply Chain Management for an ERP System

With ERP business logic stabilization, the path points towards business intelligence research, and although this goal seems somewhat separate from visualization enhancements, it is highly linked - visualization innovation provides planners with new perspectives, increasing understanding, insight and grasp of business flows, ultimately aiding the decision making process.

One of the reasons for Microsoft's success is their focus on usability, aiming to develop software that anyone can easily interact with. Concepts such as: intuitive software, user-friendly environments, simplicity, visual appealing software, are no strangers to Microsoft. With this in mind, one focus of Dynamics AX is to develop interactive visualization systems. The current visualization resides in charts and statistics, along with

Gantt-Chart planning for specific areas, such as job scheduling. Each Microsoft Dynamics department assisted by the Dynamics User Experience department tries to increase the visualization potential of their software.

This visualization project's goals go hand-in-hand with the planners way of thinking, not trying to complicate the planning task by introducing complex and unusable systems, but by gathering specific business concepts which are familiar, into one tool. The visualization itself will introduce new ideas and techniques to achieve an useful, intuitive and visually appealing tool, adding new perspectives to aid the planner with bottleneck detection and problem solving. Due to the large amount of data in ERP systems, the application must provide aggregated data, the ability to apply filters and have different perspectives over the planning, while maintaining usability.

2.1.1 Planning Environment

Microsoft Dynamics AX is a complex and vast system, integrating different business concepts into one platform. This project's integration with Microsoft Dynamics AX Manufacturing department provided a target business area: Supply Chain Management.

In Section 1.2 different types of planning were presented, each having its own purposes and goals. Supply Chain Management comprehends several planning concepts and strategies to support and maintain flow throughout the supply chain. Due to the broadness and complexity of this scope, the focus was shortened to visualization within Capacity Planning and Utilization, also providing some insight on Material Requirements Planning.

2.1.2 Capacity Planning and Utilization Concepts

For understanding the purpose and challenge of Capacity Planning, several other concepts must be acquired. This description will provide concept definitions and their relations, the required knowledge for understanding the concepts behind planning [18]. The main keyword capacity relates to the load on the available resources, for example, if there are 200 hours of possible working time and 50 are occupied, this represents, in percentage, 25% *capacity load*. Capacity may also be grouped by *Resource Type*, *Resource Group* or *Capability*, concepts explanation follows.

- **Bill Of Materials:** Every product must have an associated BOM, describing what components and quantities are required for its manufacturing process. If the manufacturer produces its own components and introduces them into other products, another concept arises, the BOM level. Each BOM may require inputs from other BOMs built by the manufacturing company; if this is the case, this is a level 1 BOM. A simple BOM level explanation is displayed in Figure 2.1. There are three levels in this picture; BOM 1 has material A and BOM 4 as inputs, and BOM 4 has material E, F and BOM 5 as inputs.

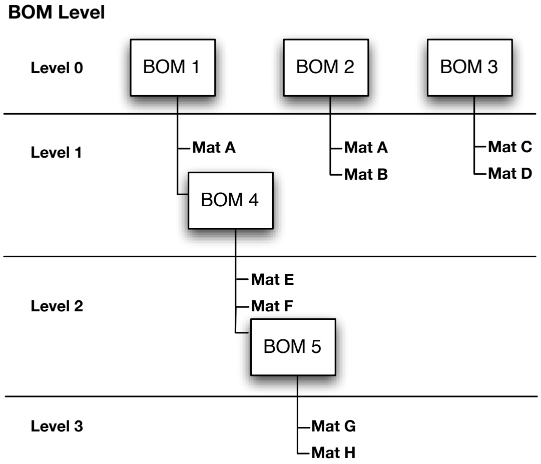


Figure 2.1: BOM Level Example

- **Route:** For a BOM to be executed a sequence of operations must be followed, a sort of manufacturing recipe; this specification is known as Route. Microsoft Dynamics AX provides a complex Route system, allowing the creation of more than a simple sequence. Routes may have networked operations and also secondary operations.

  In Figure 2.2 an simple example of a network route is displayed - Operation 30 only starts after Operations 20 and 21 finish.

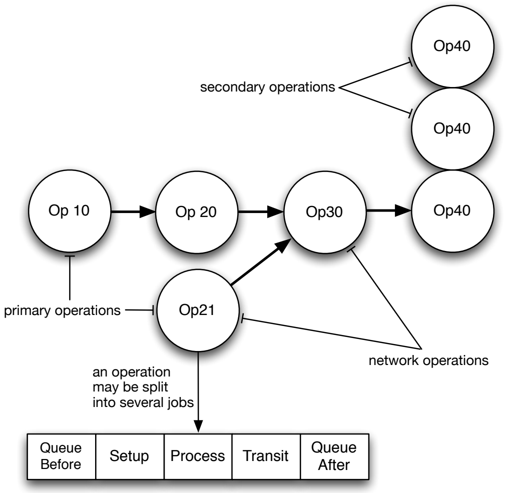


Figure 2.2: Network Route Example

- **Resource:** A resource provides a function in order to perform a task. It is defined by a *Resource Type*, associated with *Resource Group* and may have one or more

Capabilities.

- **Resource Group:** Every resource must be associated with a resource group. Usually it is a logical entity for organizational purposes, comprehending resources that are similar or able to perform the same task.
- **Resource Type:** Each resource is associated with a type: Person, Machine, Tool or Vendor.
- **Capability:** A resource has one or more associated Capabilities, providing information on what tasks the resource is able to perform, or which purposes fit it best. A resource of type person and another of type tool may have the capability paint, and may be both required to complete a job of type process, providing means to perform a painting task.
- **Operation:** An operation defines an action by one or more resources to complete a task. Dynamics AX defines that an Operation may be split into several Jobs.
- **Job:** There are several types of Jobs: queue before, setup, process, transit and queue after, each defined by a duration. For execution there are a number of associated resources capable of performing the Job. When a Job is scheduled it is defined by a start time, end time and a resource. In order to schedule a Job the resource must meet the *Capability* or *Resource Group* requirements, whichever is defined.
- **Scheduling: Backward vs Forward** The concept of scheduling has two strategies, Backward and Forward Scheduling. The first approach relies on scheduling from the due date backwards in time, while the latter approach schedules forwards a certain time to make sure the delivery is on the specified date.

  In Figure 2.3 both methodologies are displayed. In Forward Scheduling as soon as resources are available the Route is scheduled; in Backward Scheduling, the goal is to make sure the actual delivery date is as close to required delivery date as possible. The latter approach may be useful if a company has no available stock for finished products, or if the nature of the product doesn't allow it.
- **Infinite vs Finite:** In capacity and materials planning there are two ways of scheduling: Finite and Infinite. Infinite capacity assumes resources are always available, and that the company will never run out of components for manufacturing purposes. These concepts may sometimes be useful; for example: if a company relies on nails for building tables, and there is always wood and nails available due to constant availability of the materials, scheduling with infinite materials is common.

In manufacturing, a company must define well, all BOMs and Routes and every resource, along with its type, resource group and capabilities. From a capacity perspective

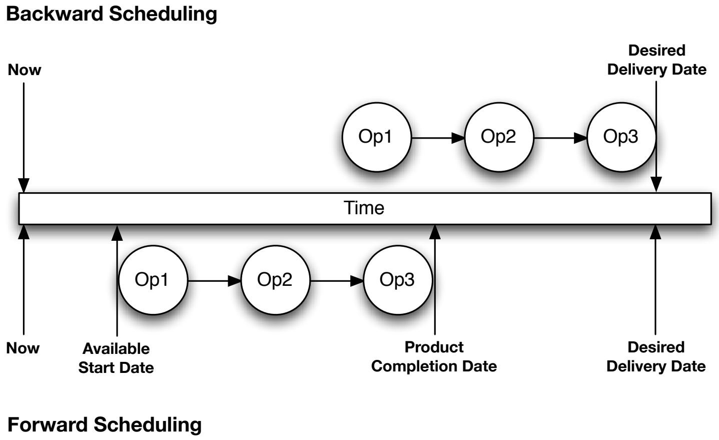


Figure 2.3: Back vs Front Scheduling

these are the required concepts for scheduling operations and jobs. There are two different types of capacity planning; usually, for long term planning (more then 1 month) the approach is Operation Scheduling; for short term planning, Job Scheduling is the solution.

Operation Scheduling

From a long term capacity planning perspective, details become irrelevant because there is much uncertainty of what will be the exact available resources in a few months. This is due to issues that will arise in a manufacturing environment: machine breakdown, unavailable employees, urgent orders and so forth, making long term scheduling *to the minute* usually impossible.

Due to the complexity of long term planning, operation scheduling is calculated using aggregated values.

Job Scheduling

In short term planning horizons, there is much more accuracy about the resources availability and so scheduling *to the minute* is possible. Job scheduling doesn't use aggregated resource availability for planning, instead it is a detailed schedule of the jobs' sequence in the manufacturing environment. It provides an actual start time, a finish time and a resource for every job, making sure they are able to be scheduled in the precise order.

2.1.3 Target Personas

Based on research and input from business customers, Microsoft has created a company model for everyone working on the software to relate with the actual people using the ERP system [19]. This model is a tool to document, capture, visualize and share how people work within departments, and how that drives performance across organizations. There is detailed information about every area of the business along with the possible job positions in the company. The positions themselves are filled by personas with a detailed description on their day-to-day, interactions with other personas, skills, objectives, primary and secondary roles, core activities, pain and stressors, and so forth.

This project, focuses on four personas, with planning responsibilities within a manufacturing environment:

- **Vince - Operations Manager** Vince's job is to make sure that everything is moving along and everyone is communicating efficiently, as well as planned correctly. Vince has to review orders, job assignments and readiness to begin work with machine operators, the shop floor supervisor, or production management. Vince sorts out emergencies stemming from jobs that have not been completed, communicates delivery times to customers, plans long term operations and rearranges future and urgent orders.
- **Tony - Production Manager** Tony is responsible for overseeing production of goods. He ensures that the necessary resources are in place and that the production plan is being carried out properly. Tony is also responsible for halting jobs when needed, due to an emergency. He provides quality control and makes sure production is going on as planned. Tony's job is to make sure everything within product development is working, so he needs to get a clear overview of what has been planned and how the routes/operations/jobs are working. Tony's two main requirements are: a graphical tool for scheduling purposes and a tool to clearly see what happens when he reshuffles an order.
- **Eduardo - Production Planner** Eduardo needs to reshuffle existing orders to make room for urgent orders on a regular basis. His goals are to make a plan that meets the customers' needs, ensuring there is enough capacity to produce the order within the lead time and aligning a production order for a new item with an existing item type. Eduardo is responsible for firefighting production order problems, prioritizing orders, identifying bottlenecks and moving orders around if there is not the necessary capacity or available inventory. He fine tunes the schedule every day and needs to know if there are any delays so he can make changes to the plan. Eduardo also coordinates scheduling issues with the machine operators. Eduardo's main pain is that is difficult to get a good overview of the production orders and their status. He

requires a tool that allows him to get a good overview of production orders, so he can rearrange urgent orders, execute what-if scenarios, and locate capacity bottlenecks is highly desirable.

- **Lars - Shop Supervisor** Lars knows the business due to his experience working in a shop floor, but his understanding of computers is limited. He is used to working with Gantt charts due to the training he had in the company. Lars's main goals are: ensure on-time delivery, respond flexibly to scheduling changes and maintain existing machinery and purchase/request new machinery and tools. Lars finds new strategies for keeping the shop floor going and monitors throughput reports, forecasts, and order status reports. He solves problems related to absence, late orders, break downs, missing or bad materials, or missing or bad technical instructions. His main pain is machine breakdowns. Lars requires a tool that allows him to assign jobs to different machine operators and helps with work reshuffling when a machine goes down or an employee is unavailable.

These personas all share concerns about not having a visualization tool that provides a good overview of what is the status of production orders, and what is the actual plan. A tool capable of running what-if scenarios and actively help with planning and locating bottlenecks.

2.2 Project Requirements

The focus of this project is to develop a visualization tool capable of providing a solid overview of planned production orders. Focusing on production orders planning and capacity planning, the overall project requirements are:

1. **Visual overview of Production Orders** - Design a tool for visually displaying a production plan.

2. **Transverse through different timelines** - Due to the large amount of data in an ERP system, its difficult to display every production order in one single view. To resolve this issue a control that enables users to transverse through different time periods is required;

3. **Filtering** - Due to the large amount of data, and even with time frame filtering, other filters must be considered, reducing visual cluttering and allowing the user to target the desired focus groups;

4. **Visual Capacity Utilization overview** - After the user has selected the desired time frame in the production orders visual overview, the user must be able to see what is their capacity impact on the available resources.

5. **Aid in capacity problems and bottleneck detection** - The capacity utilization overview should provide different filters over which resource types and capabilities are used, allowing the user to gather valuable data for bottleneck detection.

6. **Provide a tool for comparing different planning scenarios** - After a possible problem has been detected the user must be able to setup a new schedule and compare it with the original schedule;

7. **Provide Key Performance Indicators (KPI) over the plans** - When the user devises different schedules, KPIs allow to compare both plans' performance.

2.2.1 Interactive Visualization - Functional Requirements

The interactive visualization tool will provide overviews of the planned production orders, their reserved capacity and allow the user to compare it with different scheduling solutions. For a complete problem definition (from the requirements presented in the previous subsection) use cases were defined along with features. The use cases present the main required functionalities detailed in the project requirements in Section 2.2. From these, the following views and controls were defined:

1. **Production Plan Orders List:** This view is meant as an introduction to the current production plan, in the form of a list;

2. **Production Plan Orders Visualization Overview:** This view will provide a visual overview of the current production plan;

3. **Production Plan Orders Visualization Capacity Overview:** This view will provide a visual overview of resource capacity load status;

4. **Date Selector Control:** This control will allow the user to select the displayed time frame in the *Production Plan Orders Visualization Overview*. It will also provide a mechanism for linking this overview with the *Production Plan Orders Visualization Capacity Overview*. This correlation is required for analyzing what is the capacity utilization impact of the production orders in the selected time frame.

5. **Interactive Visualization GUI:** This system is the integration platform on which all previous controls will be presented.

The following list presents more details on the use cases displayed in Figure 2.4, including description, goals and priorities, pre-conditions, post-conditions and dependencies where applicable. The use cases are itemized by the system they incorporate, representing the project's functional requirements.

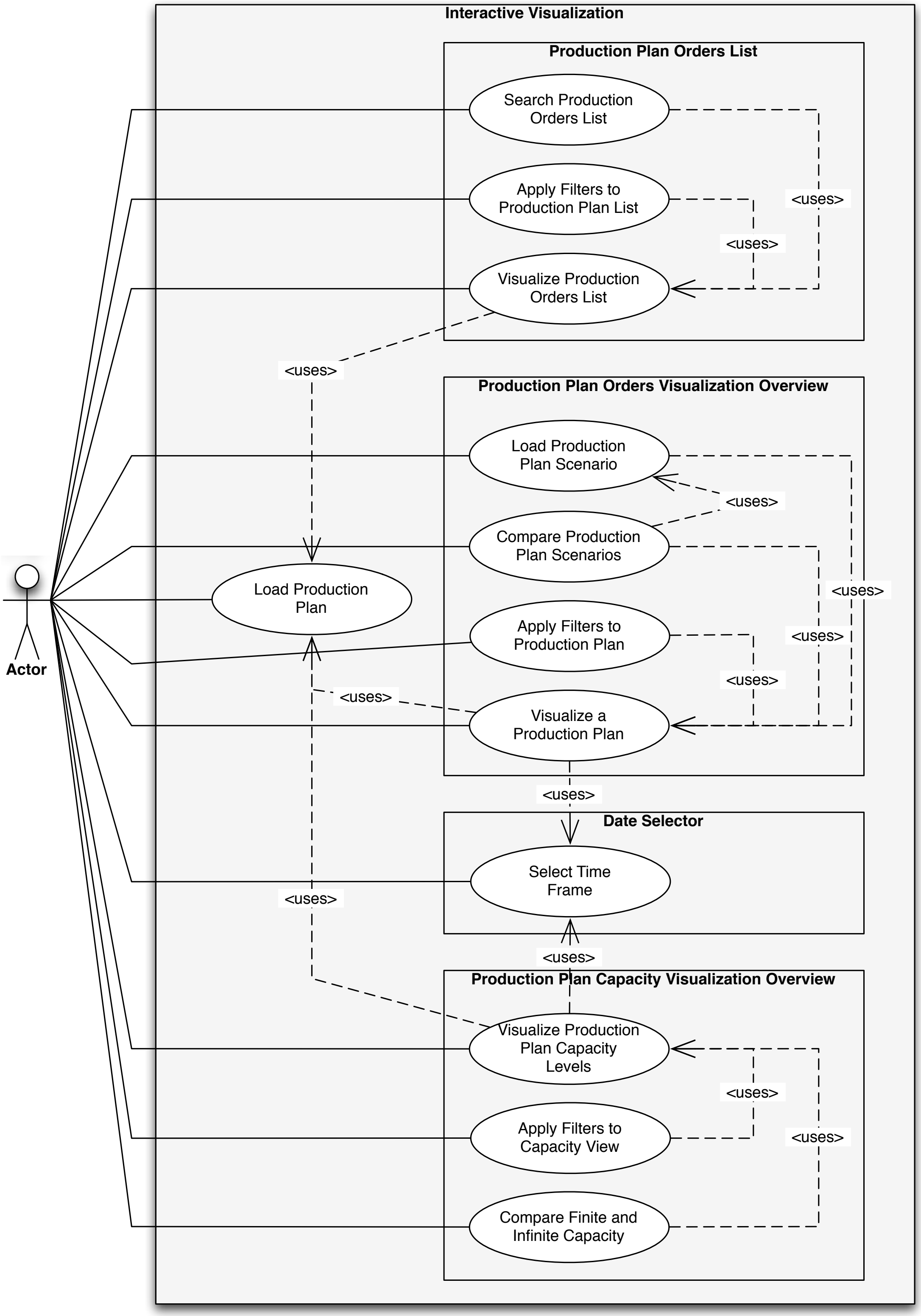

Interactive Visualization
Production Plan Orders List
Search Production Orders List
Apply Filters to Production Plan List
Visualize Production Orders List
<uses>
<uses>
<uses>
Production Plan Orders Visualization Overview
Load Production Plan Scenario
Compare Production Plan Scenarios
Apply Filters to Production Plan
Visualize a Production Plan
<uses>
<uses>
<uses>
<uses>
Load Production Plan
<uses>
Actor
<uses>
Date Selector
Select Time Frame
<uses>
<uses>
Production Plan Capacity Visualization Overview
Visualize Production Plan Capacity Levels
Apply Filters to Capacity View
Compare Finite and Infinite Capacity
<uses>
<uses>

Figure 2.4: Visualization Use Cases

- **Interactive Visualization System**

IV-UC01: Load Production Plan

It must be possible to load a Production Plan from the database system; the user will select the plan he/she setups for displaying.

Goal: With the selected plan, all objects will be populated and all the corresponding statistics will be calculated, making the production plan visualization possible.

Priority: High

Pre-Conditions: There must be at least one schedule in the database system.

- **Production Plan Orders List**

IV-UC02: Visualize Production Plan Orders List

After the loading of the production plan is complete the user is presented with a list of which production orders have been scheduled.

Goal: Provide an overview of the scheduled production orders in this plan.

Priority: High

Pre-Conditions: A schedule must have been loaded.

Dependencies: This use case depends on the Data model loaded in IV-UC01.

IV-UC03: Search Production Plan Orders List

The user must be allowed to search through the production plan orders list.

Goal: Provide a search field for the user, allowing the tracking of specific production orders.

Priority: Medium

Pre-Conditions: A schedule must have been loaded, and there must be at least two production orders.

IV-UC04: Apply Filters to Production Plan List

There must be several types of filters related with the production orders' fields.

Goal: Reduce the number of production orders in the plan, providing a good overview of what the customer wants to analyze.

Priority: High

Pre-Conditions: A schedule must have been loaded.

- **Production Plan Orders Visualization Overview**

IV-UC05: Visualize a Production Plan

The user must be able to switch from the list view defined in IV-UC02 to a visual overview of the production plan.

Goal: Provide a better way do display data, instead of using a table like view the user will be able to absorb more information and faster.

Priority: High

Pre-Conditions: A schedule must have been loaded.

Dependencies: This use case depends on the Data model loaded in IV-UC01 and in the time frame selected in IV-UC12 (detailed below).

IV-UC06: Apply Filters to Production Plan

There must be several types of filters. The first filtering feature must be a date filtering (through IV-UC12), showing only the time period the user has selected. Other filters must also be applied for other specific order data fields.

Goal: Reduce the number of production orders in the plan, providing a good overview of the time frame the customer wants to analyze.

Priority: High

Pre-Conditions: A schedule must have been loaded.

IV-UC07: Load Production Plan Scenario

The user must be able to select different production plan scenarios.

Goal: The different scenarios (where the resources are altered) will be compared with the original plan, demonstrating the capabilities of bottleneck and problems detection, as well as their resolution.

Priority: High

Pre-Conditions: A schedule must have been loaded, and there must be at least one scenario available for selection.

Dependencies: This use case uses the Data model loading system from IV-UC01.

IV-UC08: Compare Production Plan Scenarios

The user must be able to distinguish scheduling differences from visually analyzing the planning overview. This must be seen along with KPI, switching between the original plan and the planned scenario.

Goal: Provide a visualization feature for comparing schedules; along with the KPIs, this will provide a simple and easy way to absorb large amounts of information.

Priority: High

Pre-Conditions: A schedule and a planned scenario must have been previously loaded.

- **Production Plan Orders Visualization Capacity Overview**

IV-UC09: Visualize Production Plan Capacity Levels

The user must be able to switch from the list view defined in IVUC-05 to a visual overview of the capacity takes by the production plan at a selected time period.

Goal: The resource capacity load view will provide information about how the different types of resources, capabilities and resource groups are loaded, allowing the user to figure out if there is closeness to full capacity loading.

Priority: High

Pre-Conditions: The user must have loaded a schedule and selected a time period for displaying loaded resource.

Dependencies: This use case depends on the Data model loaded in IV-UC01 and in the time frame selected in IV-UC12.

IV-UC10: Apply Filters to Capacity View

The user must be able to filter the capacity view by selecting the resource type, resource group or capability loads to be displayed.

Goal: Provide a mechanism for analyzing different aggregated capacity levels, therefore increasing the likelihood of discovering bottlenecks and capacity problems.

Priority: High

Pre-Conditions: The user must have loaded a schedule and selected a time period for displaying loaded resources.

IV-UC11: Compare Finite and Infinite Capacity

The user must be able to switch between viewing finite capacity scheduling and infinite capacity scheduling.

Goal: By switching between finite and infinite capacity scheduling, the user will have a view of what are the actual resource capacity loads and what the capacity levels would have to be in order to deliver everything within the provided time period.

Priority: High

Pre-Conditions: The user must have loaded a schedule and selected a time period for displaying loaded resources.

- **Date Selector**

IV-UC12: Select Time Frame

The user must be able to select the desired production plan time frame.

Goal: By selecting a time frame, the user may focus on a specific period for analysis.

Priority: High

Pre-Conditions: The user must have loaded a schedule.

The use cases detailed in this list have been split up into features, which will constantly be updated during the *Solution Specification* and *Design and Implementation* stages of the project. Due to this project's iterative process the features list will increase from the starting use cases; Appendix A provides a complete updated features list.

2.2.2 Interactive Visualization - Non-Functional Requirements

2.2.2.1 Technological Requirements

A technological requirement of the project is the use of Microsoft tools for development, providing easy integration with Microsoft Dynamics AX. There are different technologies for achieving this goal, and so a study over Microsoft's development platforms is required.

2.2.2.2 Usability Requirements

For a Graphics User Interface (GUI) to achieve its visual objectives, best practices must be followed. Section 3.2, of the *State Of The Art* chapter, details how data should be presented to the user, and what are the requirements for providing a visually appealing application.

Microsoft Dynamics AX GUI was designed to be user friendly, always being enriched by user feedback to the User Experience Team. In designing new user interfaces, the developer should follow Microsoft's documentation on *User Experience Guidelines for Dynamics AX 2009* [20]. Figure 2.5, displays an example of the Dynamics AX GUI.

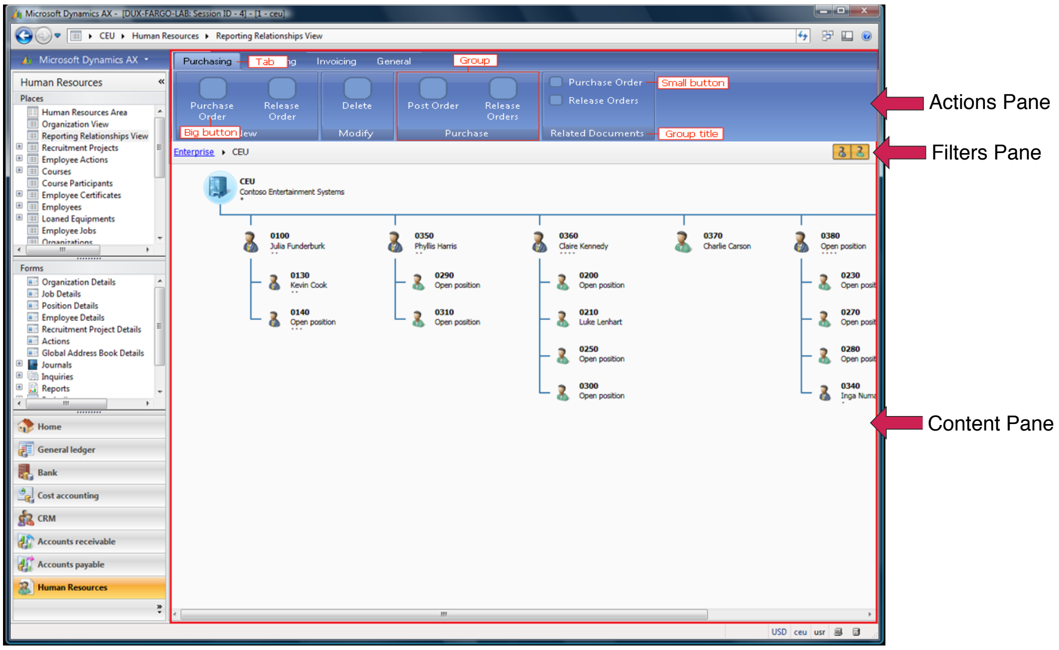


Figure 2.5: Microsoft Dynamics AX GUI

The Dynamics AX user interface must follow the defined guidelines:

- **Content Pane:** The center of the screen should display the main content;
- **Action Pane:** Presented at the top of the screen, this pane should include every action that changes what the user is viewing;

- **Filter Pane:** This pane should appear below the actions pane, providing filters for the current content. The difference between filters and actions is that filters change the user's perspective while actions change the content;
- **Fact Box:** Interaction with objects in the content pane should be presented in a fact box, displayed to the right or left of the screen.

These guidelines provide a standard frame for developing user interfaces, in order to be compliant with Microsoft Dynamics AX GUI. Following these steps, the visualization software should provide a similar user experience to that of Dynamics AX.

2.3 Schedule and Deliverables

When developing any software it is important to set milestones and deadlines. This assures that the project moves forward and meets its goals. Defining a schedule does not always mean, it has to be strictly followed, because a particular stage may take longer than planned. It is necessary to define priorities for the goals, as well as deadlines for tracking their progress. This project's schedule is detailed in Appendix E.

This project was separated into 5 stages:

1. Research
2. Brainstorm and Specification
3. Development
4. Testing and Evaluation
5. Documentation

2.3.1 Research

This stage is meant as an introduction to the project. It started with a study on SCM and *Manufacturing*, which were essential aspects to understand the problem and be able to define a solution. This was followed by a research on what is the current status of visualization within ERP systems, as well as other innovative visualization approaches, that might provide insight on how to build the required views.

2.3.1.1 Brainstorm and Specification

A brainstorm session followed the research stage, with the objective of providing ideas for the project's views, which were focused on *Production Planning* and *Capacity Utilization Planning*. It was quite difficult to come up with an idea that would change the way users look at production plans, and there were several failed approaches. Finally the idea of the *Bubble Chart* came to life, and everything started to fit together, soon followed by the 3D chart view for the capacity plan.

The final challenge was to create a new way to integrate both charts. Although the *Date Selector* already existed on the project's ideas, it only provided means for selecting a dynamic timeline, but for the capacity view it was still necessary to select different time frames. The complete breakthrough was the 2D bar chart, which was eventually developed and allows the user to select a time frame, still be able to look into other points in time, and select different time frames (displayed as weeks) for the *Capacity Utilization Planning GUI*.

At first, two views were outlined; the developed software, and a Gantt view for *Job Scheduling*. This latter view was considered to be redundant and would not add value to Dynamics AX, and was therefore disregarded.

From the brainstorming sessions, ideas started to be fully clear, which made it possible to define a software specification.

2.3.1.2 Development

This stage started by defining a design and architecture, based on the defined specification. It was decided the software would be self-contained, and not integrated into Dynamics AX; therefore a database was created, following Dynamics AX's *Production Planning* logic able to represent *Resources*, *Resource Types*, *Capabilities*, *Resource Groups*, and so forth.

Because Microsoft .NET was set as a requirement, there were only two possible APIs for developing this software: Windows Forms, and Windows Presentation Foundation. The implementation started by research on this technology, which proved to be quite powerful, with its visual and business logic separation. Its development progressed efficiently, achieving a full feature completion.

2.3.1.3 Testing and Evaluation

When developing any software it is necessary to perform testing, to make sure the software meets its goals. To complete this task, a set of different plans and scenarios was ran; two of these examples are available in the report in the chapter 6. For a final evaluation a presentation was performed before an audience, mainly composed of Program Managers, who provide a close link to customers reasoning and way of thinking. The reactions were very positive and the tool a success, with the software being seen as representing an asset for future integration in Dynamics AX.

2.3.1.4 Documentation

While previous stages were sequential, this documentation was written overlapping *Development* and *Testing and Evaluation*.

2.4 Summary

This chapter started by presenting the problem which lead to this thesis research and development, including the target ERP planning environment, and the target personas and their specific requirements. Specific manufacturing concepts were introduced, allowing a better problem understanding, and how this project aims to aid in its resolution. This section and correspondent subsections presented the project's functional requirements based on personas requirements, as well as technological and usability requirements. The final section of this chapter introduced the project's schedule and deliverables, outlining its working procedure.

The *State of The Art* chapter will perform a research on what is currently available in visualization in Microsoft's ERP system, and in its competitors, also looking at other innovative approaches for displaying data. This will provide a basis for defining a specification and a solution to address the problem at hand.

Chapter 3

State of The Art

"The outcome of any serious research can only be to make two questions grow where only one grew before."

Thorstein Veblen

This chapter begins by analyzing the current state of the art in visualization, innovative approaches, advantages and disadvantages, also researching the current visualization techniques status within ERP systems. An understanding of what is the latest in visualization will provide an insight for developing a tool that will aid production plan development, bottleneck tracking and resolution. The final segment will relay an overview over the technologies used for this project.

3.1 Data Visualization

Data visualization studies the representation of data in a concise, aggregated and simplified way, differing from information visualization in the sense that it has numerical values, instead of abstract concepts.

The traditional way of visualizing data is displaying tables with all available values, in an absolute or aggregated way. With chart evolution, data representation has moved to different visualization techniques, providing new ways of relying information through graphical means. One of the problems of these representations is that sometimes the design evolves to another level, but the goal of communicating more information is not achieved [16] [17] [13].

3.1.1 Charts

With the need of visually representing data, several charts have been developed, according to the needs of different areas and businesses. There are some common charts, used by every spreadsheet application, and others that have evolved by merging abstract concepts into data visualization [14].

3.1.1.1 Scatter Chart

The Scatter Chart consists of a collection of points represented in a two dimensional chart, as shown in Figure 3.1. A scatter plot may facilitate pattern detection with a certain confidence level, by providing correlations between two variables. A value in the horizontal axis may have different representations in the vertical axis, so valuable information may be extracted from a group of points in a common area in the chart [14] [21].

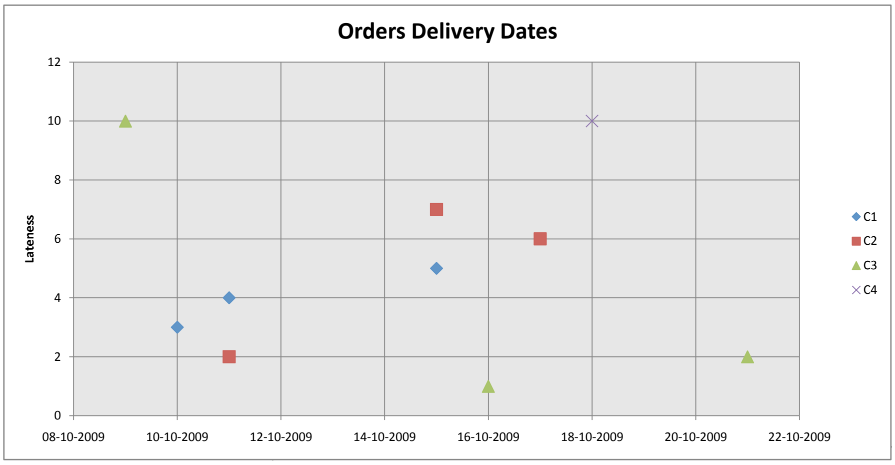


Figure 3.1: Microsoft Excel's Scatter Chart

3.1.1.2 Line Chart

The line chart is a simplified version of the scatter chart and was the first two dimensional chart to be used for data representation, as seen in Figure 3.2. The difference between the line and the scatter charts, is that the first only has one y-axis value for each x-axis value and the latter represents events, which may have different y-axis values for a given x-axis value. The line char is a simple two axis graph which is able to relate two variables in absolute or aggregated views. An increase in complexity is the use of series which represent different groups of points that share the same axis variables. Usually the horizontal axis is chronological, representing a function variation over time [14].

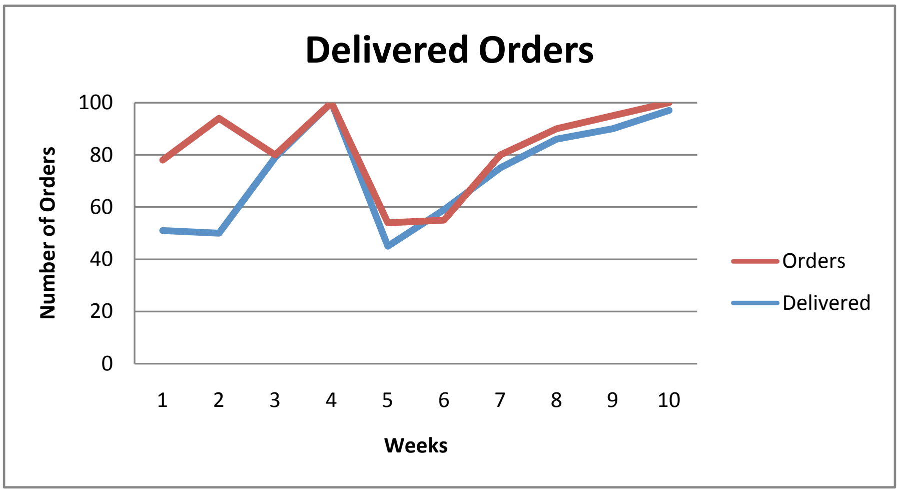


Figure 3.2: Microsoft Excel's Line Chart

3.1.1.3 Bar Chart

The bar chart is composed of several bars varying in length, representing value; their orientation can be vertical or horizontal. Several bars may be grouped to represent different concepts within the same variable value. Although visually different, it communicates the same information as a line chart, which is able to achieve the grouping effect through several series. What differs in these charts is the granularity, line charts may represent data varying in time (or any other variable), while in bar charts data is usually aggregated. The bar chart is mainly used to compare different occurrences of grouped characteristics of data in amounts or frequency.

Figure 3.3 displays a bar chart developed using Microsoft Excel, representing two stacked series, *number of orders* and *number of delivered orders*. It is a simple way to present data which allows the user to have a quick overview of what happened during a few weeks [14].

3.1.1.4 Pie Chart

The pie chart is a circular chart, divided into segments, used for representing magnitudes or frequencies. It is a somewhat limited chart, with each segment representing a certain quantity, and it is used mainly for representing percentages.

As displayed in Figure 3.4, a pie chart is able to provide a quick overview of the frequency and impact of a certain event, or group of events, in the full 100% the pie represents [14].

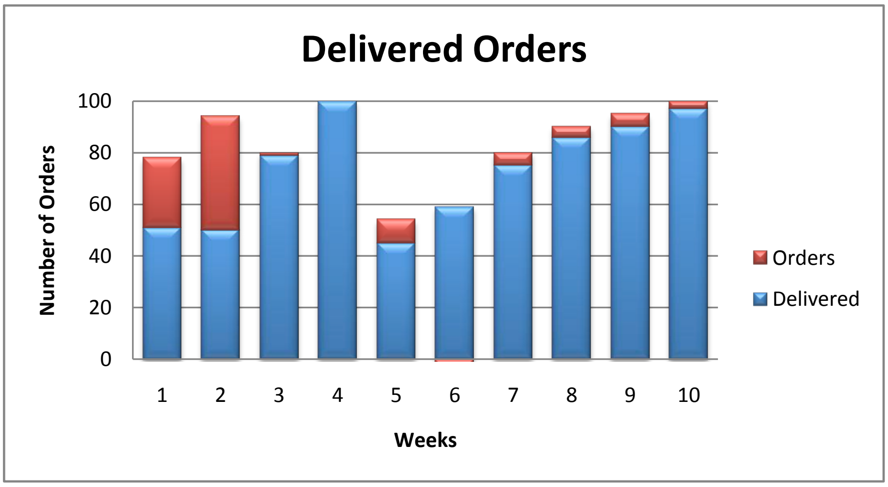


Figure 3.3: Microsoft Excel's Bar Chart

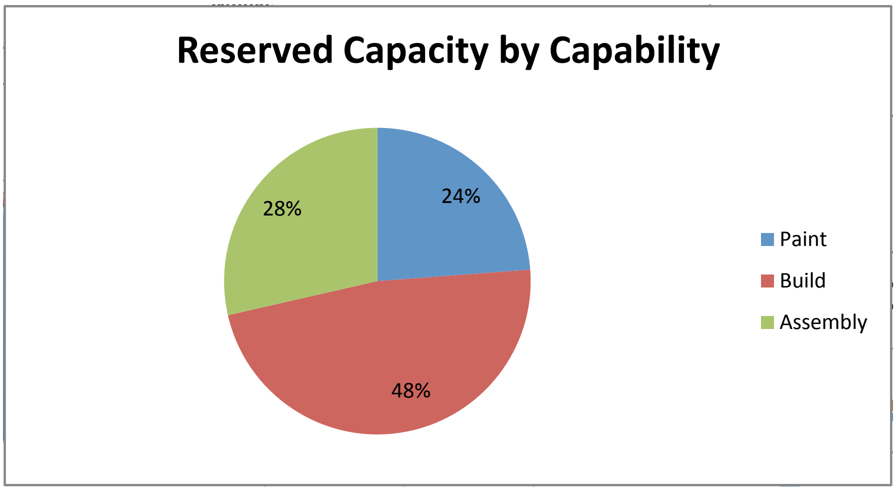


Figure 3.4: Microsoft Excel's Pie Chart

3.1.1.5 Area Chart

The area chart, which is based on the line chart, is composed of different functions that make up an area between their values. There are two possible ways of displaying data. In the quantitive way, each line in the horizontal axis has an area between its values and the previous line values, represented in the vertical axis. The other view is a layered area, where each line has an area in the vertical axis and may intersect another line's area; this may be useful for viewing ranges of intersection between both functions, possibly extracting useful information [14]. The latter view is shown in Figure 3.5.

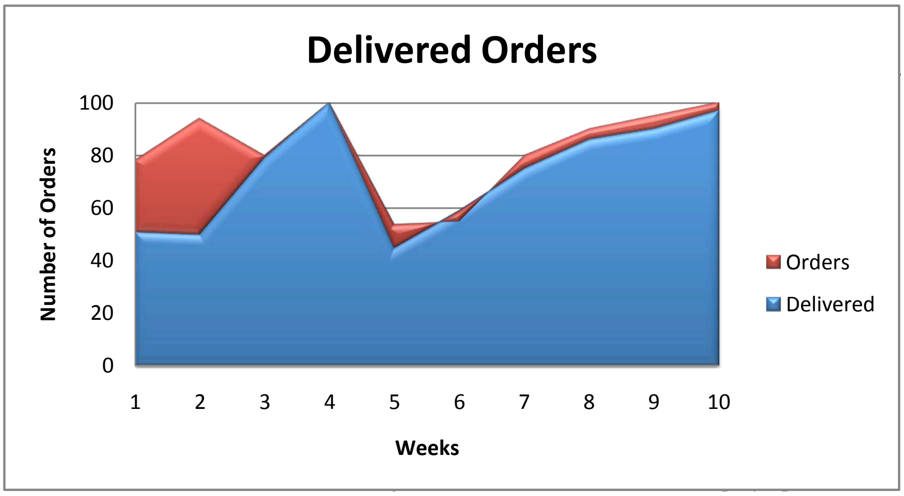


Figure 3.5: Microsoft Excel's Area Chart

3.1.1.6 Timeline and Gantt Chart

The timeline chart provides a graphical representation of a chronological sequence of events. The time granularity changes according to the need of the business. A Gantt chart is similar to a timeline chart; usually the horizontal axis provides a time reference of year, month, week, day, minute and second.

Timeline charts are used for providing views over chronological events, with time represented in the horizontal axis; events are represented by points or lines along this axis, which may also also be displayed in segments of time.

Gantt charts use the horizontal axis as a time reference and the events are represented with bars, defining start and end. They are used mainly for displaying schedules, representing tasks along time in the vertical axis, which have lines connecting the several events, represented with bars [14] [22].

3.1.1.7 Radar Chart

The radar chart, also known as spider chart, allows the visual representation of multivariate data. The center of the chart represents the start point, and the end represents the maximum value of this variable. Several series may be represented, which allows for easy comparison between variables which have different values (some may be numerical, like 1,2,3,..., and others just concepts, like bad, medium, good, very good, ...).

In figure 3.6 is the representation of a radar chart. It facilitates comparing of the several weeks performance in a easy manner [14].

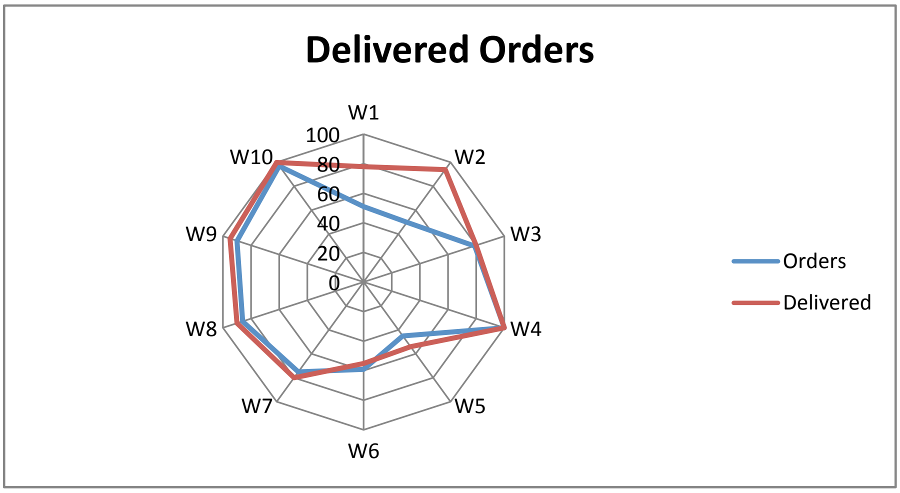


Figure 3.6: Microsoft Excel's Radar Chart - Delivered Orders per Week

3.1.1.8 Bubble Chart

The bubble chart is the most complex of the common charts, due to its several dimensions. It is an evolution of the scatter chart, with two extra variable concepts. The scatter chart has points, which are the same size and usually the same color, distributed along an horizontal and a vertical axis. In a bubble chart the axis concept is the same but the size of the points changes, transforming them into bubbles which assume different colors. These four dimensions (two axis, bubble size and color) provide a concise and impressive way of displaying data, making the perceiver assimilate a high quantity of information in an instant.

Figure 3.7 displays the full potential of a bubble chart. The amount of information instantly perceived is much higher than with other charts. This representation using Microsoft Excel's charting tool, shows the number of orders planned for this week, along the horizontal timeline, and the lateness they incur in the vertical axis. The other two dimensions, size and color, are used from representing value of an order and customer [14] [21].

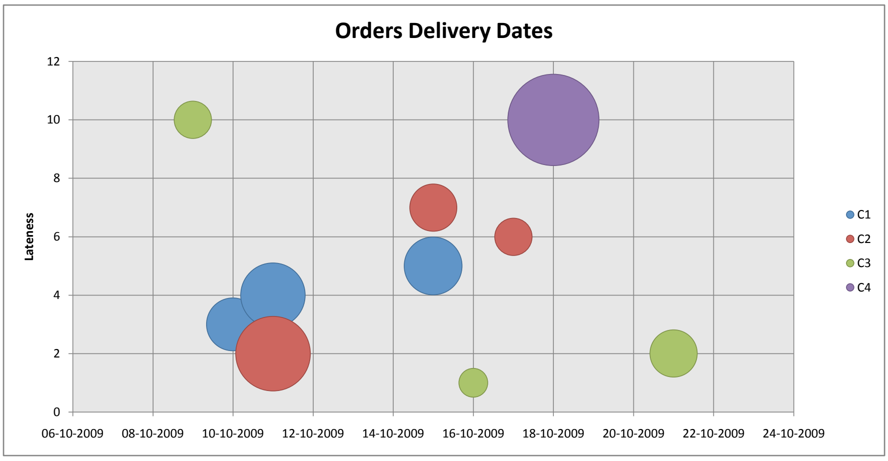


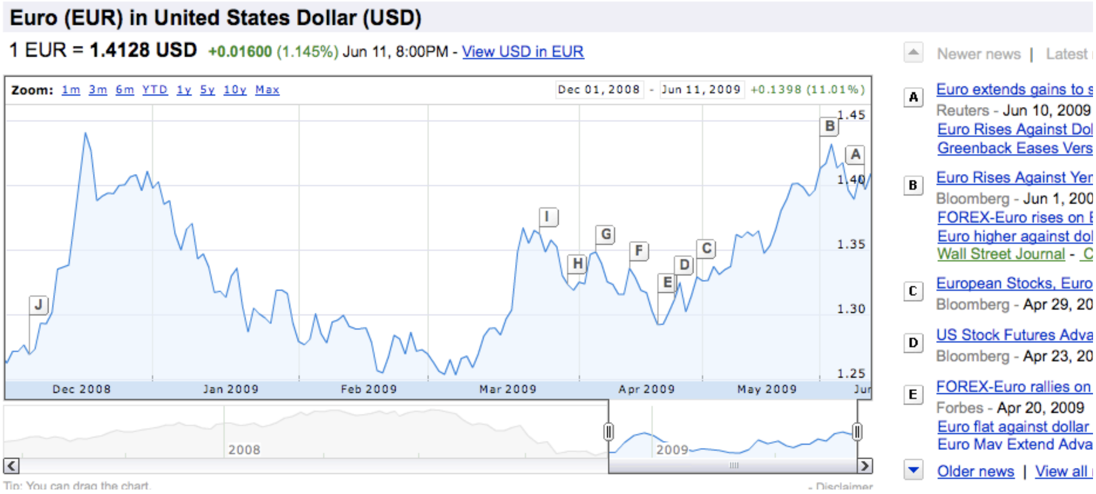
Figure 3.7: Microsoft Excel's Bubble Chart

3.1.2 Innovative Approaches

Due to chart limitations for displaying abstract concepts, innovative techniques have been introduced in different businesses. By using the latest technologies available, design and visualization have been able to advance and find creative ways of presenting whichever concepts. One of the main limitations of charts is the fact they are static, the use of cutting edge techniques, creates new perspectives over the same data [23] [24].

3.1.2.1 Google Finance

Google Finance presents an innovative way of displaying currency changes, as seen in Figure 3.8. The chart, which is an advanced *Line Chart*, provides a value reference for the stock variation by using a line in the vertical axis, while time is represented in the horizontal line [1].


Euro (EUR) in United States Dollar (USD)

1 EUR = **1.4128 USD** +0.01600 (1.145%) Jun 11, 8:00PM - View USD in EUR

Newer news | Latest

A Euro extends gains to s
Reuters - Jun 10, 2009
Euro Rises Against Dol
Greenback Eases Vers

B Euro Rises Against Yen
Bloomberg - Jun 1, 200
FOREX-Euro rises on E
Euro higher against dol
Wall Street Journal - C

C European Stocks, Euro
Bloomberg - Apr 29, 20

D US Stock Futures Adva
Bloomberg - Apr 23, 20

E FOREX-Euro rallies on
Forbes - Apr 20, 2009
Euro flat against dollar
Euro Mav Extend Adva

Older news | View all

Tip: You can drag the chart.

- Disclaimer

Figure 3.8: Google Finance, Displaying USD-EUR variations [1]

The chart also displays some main events which caused the currency to vary, by using labels pointing at the specific value/time, and showing the news on the right side. Another interesting innovation is the time period selector, which allows the user to select a dynamic time frame. It also provides a less detailed version of the main chart, increasing navigation by allowing the user to look ahead into other time periods without actually selecting them.

3.1.2.2 Web Trends

Information Architects Japan presents Web Trend Map 4.0, a representation of the most successful websites on the internet, and their owners. The representation is based on Tokyo's Metro Map, each domain is evaluated based on traffic, revenue, age and the company. Its usage is very simple; the station's height provides a reference to its success, based on traffic, revenue and trend; its width represents the company's stability; and the station's location is the category or business it is in.

The companies' traffic is directly mapped with Tokyo's Metro traffic, as seen in Figure 3.9.

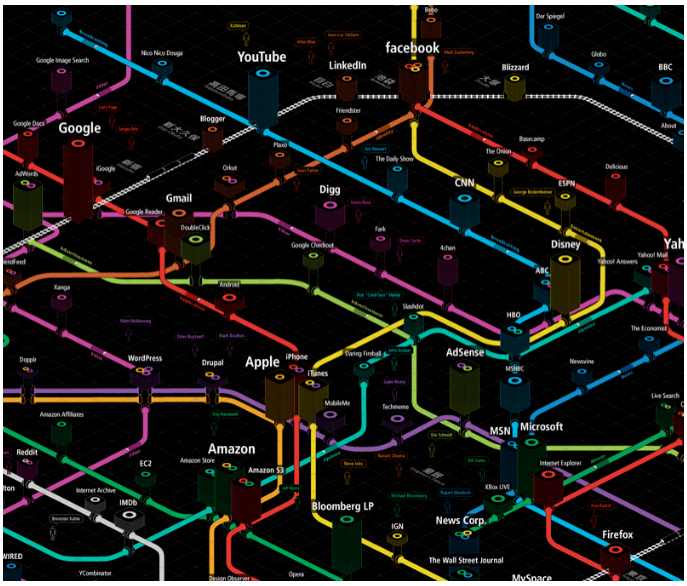


Figure 3.9: Web Trend Map 4.0 - Zoomed Version displaying Microsoft, Google and Apple

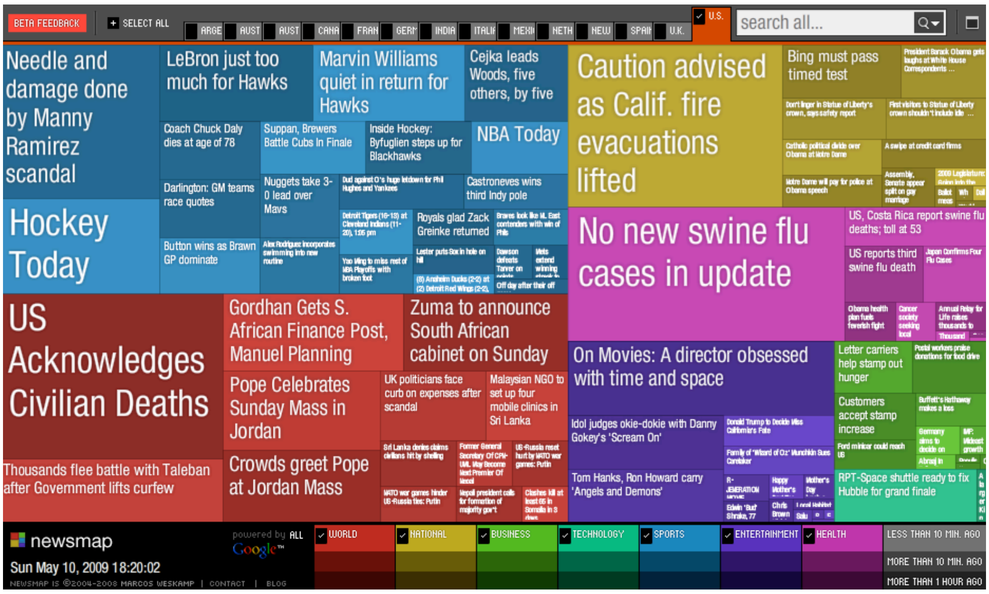
"For example, Twitter is located in Shibuya this year, as Shibuya is the station with the biggest buzz. Google is placed in the busiest, most highly trafficked train station in the world: Shinjuku. The New York Times, the "Old Gray Lady" is located in Sugamo - a shopping paradise for Tokyo's grandmothers."

WebTrends [25]

Although at first it seems somewhat complicated or nonsense, once you start reading and become familiar with the companies displayed, you understand not only the category grouping but also their closeness or line crossing. A simple, yet innovative, way of creating a map of *places to visit* and *how they are linked* in the internet.

3.1.2.3 Newsmap

Newsmap is a website that displays news in rectangles, differing in color and size, as seen in Figure 3.10. The color indicates the type of news it represents (World, National, Business, Technology, Sports, Entertainment and Health), and it also dims from a bright to a darker version to represent how long it has been posted (*less than 10 min. ago*, *more then 10 min. ago* and *more then 1 hour ago*). The size of the rectangle reflects the popularity of the news.



Needle and damage done by Manny Ramirez scandal

LeBron just too much for Hawks

Marvin Williams quiet in return for Hawks

Cejka leads Woods, five others, by five

Coach Chuck Daly dies at age of 78

Suppan, Brewers Battle Cubs In Finale

Inside Hockey: Byfuglien steps up for Blackhawks

NBA Today

Darlington: GM teams race quotes

Nuggets take 3-0 lead over Mavs

Dud against O's huge letdown for Phil Hughes and Yankees

Castroneves wins third Indy pole

Hockey Today

Detroit Tigers (16-13) at Cleveland Indians (11-20), 1:05 pm

Royals glad Zack Greinke returned

Braves look like NL East contenders with win of Phils

Button wins as Brawn GP dominate

Alex Rodriguez incorporates swimming into new routine

Yao Ming to miss rest of NBA Playoffs with broken foot

Lester puts Sox in hole on hill

Dawson defeats Tarver on

Mets extend winning

(8) Anaheim Ducks (2-2) at (2) Detroit Red Wings (2-2),

Off day after their off

Caution advised as Calif. fire evacuations lifted

Bing must pass timed test

President Barack Obama gets laughs at White House Correspondents ...

Don't linger in Statue of Liberty's crown, says safety report

First visitors to Statue of Liberty crown shouldn't include idle ...

Catholic political divide over Obama at Notre Dame

A swipe at credit card firms

Notre Dame will pay for police at Obama speech

Assembly, Senate appear split on gay marriage

2009 Legislature:

No new swine flu cases in update

US, Costa Rica report swine flu deaths; toll at 53

US reports third swine flu death

Japan Confirms Four Flu Cases

Obama health plan fuels feverish fight

Cancer society seeking local

Annual Relay for Life raises thousands to

Thousand

US Acknowledges Civilian Deaths

Gordhan Gets S. African Finance Post, Manuel Planning

Zuma to announce South African cabinet on Sunday

Pope Celebrates Sunday Mass in Jordan

UK politicians face curb on expenses after scandal

Malaysian NGO to set up four mobile clinics in Sri Lanka

Sri Lanka denies claims civilians hit by shelling

Former General Secretary Of CPN-UML May Become Next Premier Of Nepal

US-Russia reset hurt by NATO war games: Putin

Thousands flee battle with Taleban after Government lifts curfew

Crowds greet Pope at Jordan Mass

NATO war games hinder US-Russia ties: Putin

Nepal president calls for formation of majority gov't

Clashes kill at least 65 in Somalia in 3

On Movies: A director obsessed with time and space

Idol judges okie-dokie with Danny Gokey's 'Scream On'

Donald Trump to Decide Miss California's Fate

Family of 'Wizard of Oz' Munchkin Sues Caretaker

Tom Hanks, Ron Howard carry 'Angels and Demons'

R-JENERATION

Happy Mother's

Mother's Day

Edwin 'Bud' Shrake, 77

Chris Brown

Letter carriers help stamp out hunger

Postal workers praise donations for food drive

Customers accept stamp increase

Buffett's Hathaway makes a loss

Germany aims to decide on

IMF: Mideast growth

Ford minicar could reach US

RPT-Space shuttle ready to fix Hubble for grand finale

Sun May 10, 2009 18:20:02

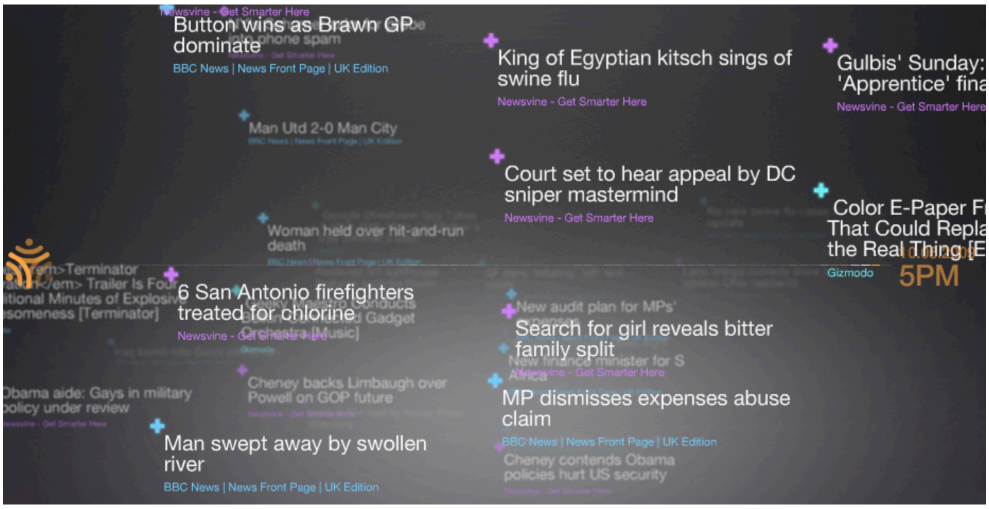
NEWSMAP IS ©2004-2008 MARCOS WESKAMP | CONTACT | BLOG

Figure 3.10: Newsmap - Representing USA News on Sunday May 10, 2009

It is an easy way to catch up on the latest news, and find out what are the most popular topics, which usually means the ones everyone should be up-to-date on. There is also a new filter feature which allows users to select the country they want to receive news from [26].

3.1.2.4 Voyage - RSS feed reader

Voyage uses RSS feeds from different media around the world as a news source. It displays news in a *gravity area* - the news that relate with the selected viewing time are bright and in front, while older news appear out of focus in the back, as seen in Figure 3.11. The user is able to transverse through the news by scrolling with the mouse, moving up goes back in time, and down goes forward. The color on the news also shows from which feed it comes from [27].



Newsvine - Get Smarter Here
Button wins as Brawn GP dominate
BBC News | News Front Page | UK Edition

King of Egyptian kitsch sings of swine flu
Newsvine - Get Smarter Here

Gulbis' Sunday: 'Apprentice' fina
Newsvine - Get Smarter Here

Man Utd 2-0 Man City

Court set to hear appeal by DC sniper mastermind
Newsvine - Get Smarter Here

Color E-Paper F That Could Repla the Real Thing [E
Gizmodo

Woman held over hit-and-run death

Terminator Trailer Is Four Additional Minutes of Explosive Awesomeness [Terminator]

6 San Antonio firefighters treated for chlorine
Newsvine - Get Smarter Here

New audit plan for MPs'

Search for girl reveals bitter family split
Newsvine - Get Smarter Here

New finance minister for S Africa

Obama aide: Gays in military policy under review

Cheney backs Limbaugh over Powell on GOP future

MP dismisses expenses abuse claim
BBC News | News Front Page | UK Edition

Man swept away by swollen river
BBC News | News Front Page | UK Edition

Cheney contends Obama policies hurt US security

Figure 3.11: Voyage Gravity Area

3.1.2.5 American Population Density

Time Magazine has a chart displaying the density of the American population within the USA. The base is a map of the country with every city having a hill associated with its population, as displayed in Figure 3.12. This is a simple and innovative view, allowing users to relate straight away and easily locate whatever city they are searching and find out its density compared with the rest [28].

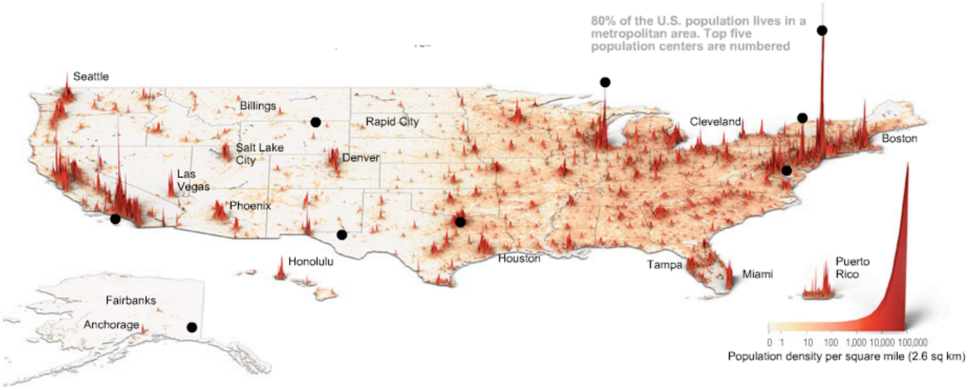


Figure 3.12: Time Magazine - American population density

3.1.2.6 Network Tag Visualizer

Fidg't Visualizer allows the user to interact in a different way with Flickr and LastFM tags. It allows users to create *magnets* which they can distribute along the Visualizer viewing area and network members will arrange accordingly. The more the relation the users has with a certain *magnet* tag, the closer they will be to it [29], as seen in Figure 3.13.

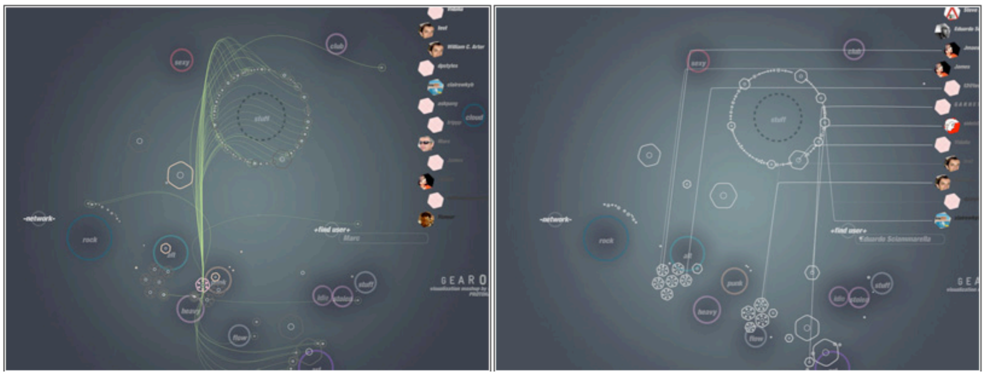
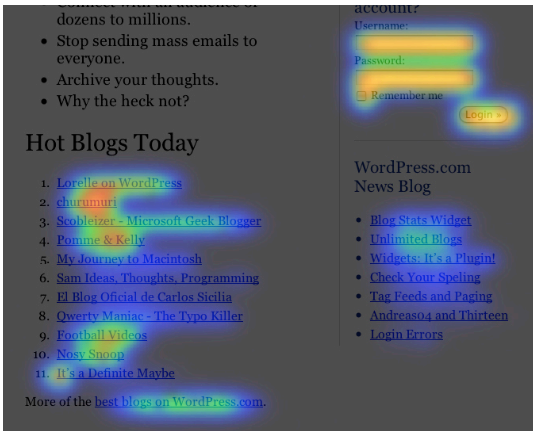


Figure 3.13: Fidg't Visualizer Area

3.1.2.7 Heat Map

Crazy Egg provides a tool for website owners that allows users to view the website in a heat map. The warmer the displayed area, the more the users have clicked on it, as seen in Figure 3.14. A new and interesting way to get to know what is visited the most within a website, unlike the traditional spreadsheet with a sum of page views [30].



dozens to millions.
- Stop sending mass emails to everyone.
- Archive your thoughts.
- Why the heck not?

Hot Blogs Today

1. Lorelle on WordPress
2. churumuri
3. Scobleizer - Microsoft Geek Blogger
4. Pomme & Kelly
5. My Journey to Macintosh
6. Sam Ideas, Thoughts, Programming
7. El Blog Oficial de Carlos Sicilia
8. Qwerty Maniac - The Typo Killer
9. Football Videos
10. Nosy Snoop
11. It's a Definite Maybe

More of the best blogs on WordPress.com.

account?

Username:

Password:

□ Remember me

Login »

WordPress.com News Blog

- Blog Stats Widget
- Unlimited Blogs
- Widgets: It's a Plugin!
- Check Your Speling
- Tag Feeds and Paging
- Andreas04 and Thirteen
- Login Errors

Figure 3.14: Crazy Egg - Heat Map

3.2 Usability Best Practices

There have been several studies over user experience trying to find out what are the best ways to visually organize concepts. For developing GUIs some guidelines and best practices are followed, increasing the user experience and providing an overall product satisfaction [21].

However well software is architectured, every use case and application behavior will fail to communicate the appropriate concepts and ideas if the GUI is not design to meet users' needs [31]. The human brain has cognitive abilities we are only now discovering such as, the ability to aggregate data, find patterns and visualize large amounts of data in one single view, which can nonetheless be overwhelming. A final key for success in developing any interface is to know the players, be aware of what users want and what the business is all about.

According to Edward R. Tuffle [13], there are two important problems in information design :

1. It is difficult to display multidimensional information on a two-dimensional surface.
2. The resolution of the display surface is often not high enough to display dense information. Computers present a particular challenge - although they can add motion and interactivity, computer displays have low information density compared to that of paper.

Although both points are true, the developer can leverage the information density from papers to displays by introducing one key aspect: interactivity. Computers simplify the interaction process, compared with regular print outs.

Also according to Edward R. Tuffle [13] [31], visually displayed information should :

- **Enforce visual comparisons**

  There should be a comparison feature, allowing the user to compare related variables or switch between different scenarios.

- **Show causality**

  Clearly identify and clarify causality and its effects in the visualization.

- **Show multiple variables**

  Allow the user to have different perspectives over the same data; switching variables allows the user to achieve a better understanding over the data.

- **Integrate text, graphics and data in one display**

  Diagrams, graphics and charts should be organized in a matter that allow text and data integration, removing the users' hassle from switching between diagrams and table listings, and aggregating concepts.

- **Ensure the quality, relevance, and integrity of the content**

  Information should be well thought of before being introduced into any interface. Unreliable or poor-quality information will damage the users' trust in the products' content.

- **Show content adjacently in space, and not stacked in time**

  For showing different time periods, following the regular x-axis view for time displaying is usually the correct approach.

- **Don't de-quantify quantifiable data**

  Numbers should always be available to the user; it may be useful to create a different perspective and allow data to be hidden from the visual representation, but it should always be at *the click of a button* in case the user requires it.

A methodology by Ben Fry is also available in his book *Visualizing Data*[21], outlines *Why Data Display Requires Planning* and defines *The Seven Stages of Visualizing Data*. While developing a graphical user interface, one should consider the following:

1. **Too Much Information:**

   Large quantities of information make it extremely difficult to understand the "*big picture*";

2. **Data Collection:**

   Understand what can be done with the collected data and define methods to understand and communicate this information;

3. **Thinking About Data:**

   Understand what the data means and what the user values as useful information;

4. **Data Never Stays the Same:**

   Build software that is updated in real-time because data is always changing;

5. **What is The Question?**

   Be specific about what the visualization is supposed to show; the more specific one is the clearer the visual result will be;

6. **A Combination of Many Disciplines:**

   To be able to display data there must be an understanding of several fields: statistics, data mining, graphic design, and information visualization.

7. **Process:**

   Data must be processed and worked to provide a solid output to the user; the following steps should be performed: Acquire, Parse, Filter, Mine, Represent, Refine and Interact.

These are some guidelines and best practices for creating an user-friendly visual interface. Following these premises should provide a solid and consistent application, that users will easily relate with.

3.3 Enterprise Resource Planning

Enterprise Resource Planning (ERP) is a complex software solution bundle, which aggregates data and is modeled to the customer's business. If correctly implemented into a company, one of its main advantages is the merge of all of the business's data into one single database. This allows the customer to elaborate plans to drive the business, fully aware of all areas, such as: finance, manufacturing and supply chain management, logistics, sales and marketing, human resources and so forth.

Through its complex database, an ERP solution will provide an overview of what is happening in the company, what areas are working on schedule and according to predictions, keeping track of every movement within the company, therefore increasing efficiency and productivity and allowing the company to focus on its core business.

In order for the systems integration to be a success, an ERP system must obey fundamental rules:

1. The system must be correctly modeled to the business itself, integrating day-to-day currency used in the company, in whichever units and measurements. This will provide a better understanding, for the people using this tool, and also an easier ground for planning.

2. Every aspect of the business should be modeled, integrated and tracked by the system. The smallest detail may ruin a business's success and cause an implementation failure, which usually results in bankruptcy due to the elevated costs of integration. Another key aspect is to leave no legacy systems behind, running unreported routines, outside the system; if unachievable, then full integration, with a purpose built interface is required.

3. Modularity is a key factor in an ERP system; it should be able to scale and adjust to the company's needs, also providing easy expanding with other business modules, as they are required.

4. The software must prove to be a facilitator. If a system proves to be more of a hassle then it is worth it will cease to be used, or it will be incorrectly used.

The ERPs goal is to help manage a company by providing the information required to turn decision making into a simpler task, including several decision support tools into their package [10] [11].

3.3.1 Planning And Visualization

With the integration of key areas of companies' businesses, ERP systems have evolved into enormous software bundles. With the advantage of integration and ability to overview the full length of the business in *real-time* a question emerged: ***How can we display this amount of data?***. One of the fundaments of planning is analyze and understand information, turning it into useful knowledge; this is where visualization techniques come into place.

The first versions of ERP systems provided tabular views and simple charts, which become hard to read when the length of the business increases. With the evolution of computer graphics, it became clear visualization was an area of focus, and ERPs have been trying to move forward in these regards, integrating several different solutions into their individual packages (financial, logistics, manufacturing, etc.).

3.3.1.1 SAP

Systems Applications and Products (SAP) was founded in 1972 by Dietmar Hopp, Hans-Werner Hector, Hasso Plattner, Klaus Tschira, and Claus Wellenreuther, five former IBM engineers. The goal was to develop a standard application software for real-time business processing. Following a rapid growth in the 1980s and with the introduction of new technologies in the 1990s, SAP has grown to 75,000 customers in 120 countries by 2008. Delivering 25 industry-specific business solutions, SAP became the third largest independent software vender worldwide [32] [33].

SAP has been one of the first in ERP systems development and is currently leader in implementation and deployment around the world. In the beginning of ERPs development, customization was difficult, which turned these systems into expensive investments, only attainable by big companies. With the systems' evolution and with reduction of the ”big company replenishment” it was clear that mid-sized and small companies represented an important niche. With this new approach in mind, trying to cover as much of the market as possible, SAP started developing affordable and generic solutions, with some customization, expanding the number of possible customers [32] [33].

Today, SAP provides fully customized software solutions for big companies, and generic bundles for mid-sized and smaller companies. The main modules of SAP Business Suite, the big company package solution, are:

- SAP Customer Relationship Management
- SAP ERP (enterprise resource planning)
- SAP Product Lifecycle Management
- SAP Supply Chain Management
- SAP Supplier Relationship Management
- SAP Manufacturing

Every company may purchase whichever package fits their business, or integrate several packages due to the modularity of ERP systems. The process of integration and customization of the software modules is the core of the ERP implementation and companies have these packages modeled around their business [34].

For smaller companies, the solutions are based on the main software packages. Less customized, very interesting and still very powerful, containing:

- SAP Business All-in-One
  This bundle includes all the solutions from the SAP Business Suite, such as CRM, ERP, PLM, SCM and SRM, but adjusted to smaller companies requirements.

- SAP Business ByDesign
  ByDesign is a good approach for small businesses. Based on SAP ERP main system and relying on its modularity, it allows customers to introduce whichever modules they require at their own pace.

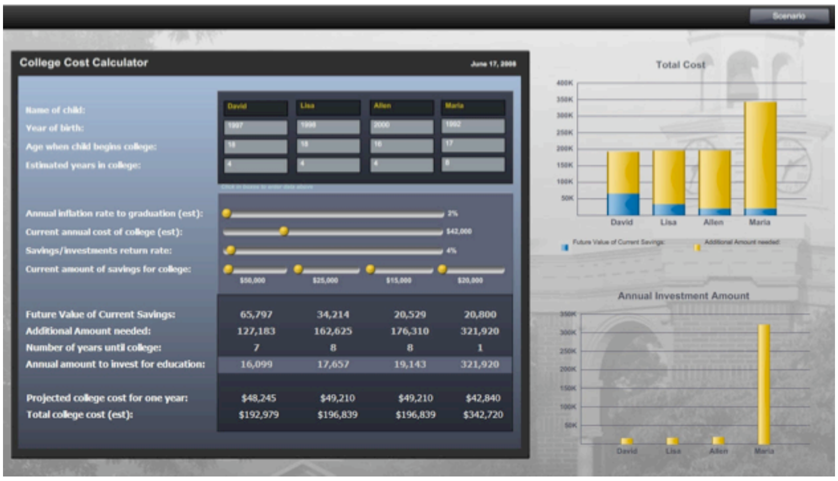
- SAP Business One
  Unlike the previous bundles, Business One is not a derived solution from SAP ERP. It is fully targeted to small companies, containing all required tools for managing a small business with efficiency. It provides several features as: ”Accounting and Financials”, ”Customer Relationship Management”, ”Operations and Distribution” and ”Administration and Reporting” [34].

SAP Visualization

If the purpose is planning, an Advanced Planning System (APS) is required for organizing and presenting concepts, data and information. Within SAP's helper tools for business planning and control, SAP Business Objects, there are two customizable visualization tools, Xcelsius Engage and Xcelcius Present, capable of creating dashboards and statistics, allowing customers to create views of whatever they feel is necessary. The potential of Xcelsius's tools is unquestionable, representing a step forward in fully customizable environments [35] [36].

Xcelsius provides functionalities for analyzing plans (Figure 3.15), cost analysis (Figure 3.16) and run different what-if scenarios (Figure 3.17)

Views are based on statistics from the ERP *real-time* data, helping planners analyze data from several perspectives, run different scenarios and make better decisions.



College Cost Calculator

June 17, 2008

Name of child:	David	Lisa	Allen	Maria
Year of birth:	1997	1998	2000	1992
Age when child begins college:	18	18	16	17
Estimated years in college:	4	4	4	8

Click in boxes to enter data above

Annual inflation rate to graduation (est): 2%

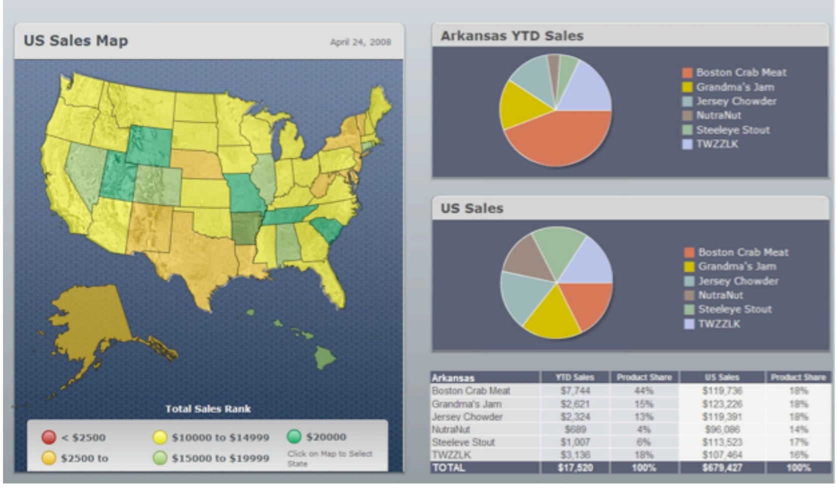
Current annual cost of college (est): $42,000

Savings/investments return rate: 4%

Current amount of savings for college: $50,000 $25,000 $15,000 $20,000

Future Value of Current Savings:	65,797	34,214	20,529	20,800
Additional Amount needed:	127,183	162,625	176,310	321,920
Number of years until college:	7	8	8	1
Annual amount to invest for education:	16,099	17,657	19,143	321,920
Projected college cost for one year:	$48,245	$49,210	$49,210	$42,840
Total college cost (est):	$192,979	$196,839	$196,839	$342,720

Figure 3.15: Xcelsius Dashboard - Plan Analysis



Arkansas	YTD Sales	Product Share	US Sales	Product Share
Boston Crab Meat	$7,744	44%	$119,736	18%
Grandma's Jam	$2,621	15%	$123,226	18%
Jersey Chowder	$2,324	13%	$119,391	18%
NutraNut	$689	4%	$96,086	14%
Steeleye Stout	$1,007	6%	$113,523	17%
TWZZLK	$3,136	18%	$107,464	16%
TOTAL	$17,520	100%	$679,427	100%

Figure 3.16: Xcelsius Dashboard - Cost Analysis

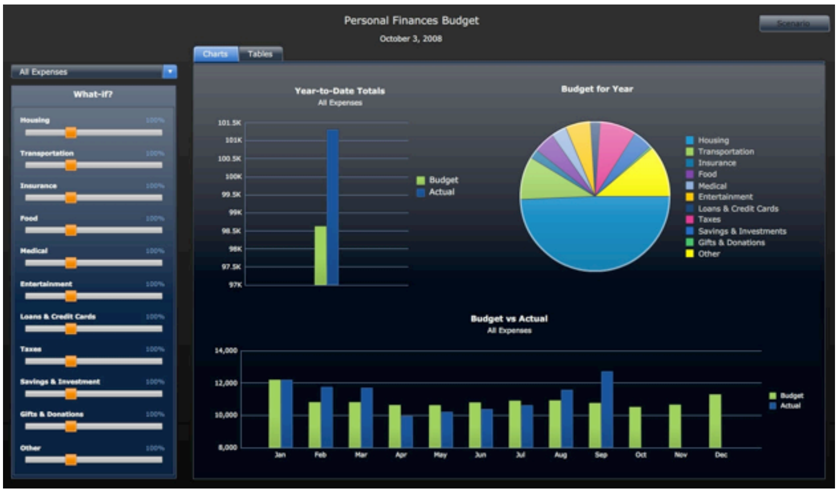


Figure 3.17: Xcelsius Dashboard - What-if Scenarios

3.3.1.2 Oracle

Oracle was founded in 1977 by Larry Ellison, Bob Miner and Ed Oates, after realizing the market niche in advanced and powerful relational database models. Today Oracle is one of the main companies in the database market, offering advanced solutions to their customers. After the database business settled down, Oracle started to invest in several solutions which rely on advanced database systems, such as ERPs, CRMs, Banking Software, SOA, Financial Management, and so forth, having 320,000 customers in 145 countries [37]. This business' turn and evolution was mainly due (as in most large companies) to the purchase of several mid-sized companies with good representation in their targeted software area [38].

The E-Business Suite is Oracle's milestone in ERP development and one of the biggest in the business, targeting the large companies. As with other ERPs, Oracle's solution is divided into several modules, fully integrable, which are acquired as they are required, and fitted into one big tool. Its main modules are:

- Asset Lifecycle Management
- Customer Relationship Management
- Enterprise Resource Planning
  - Channel Revenue Management
  - Financial Management
  - Human Capital Management
  - Project Management
- Procurement
- Product Lifecycle Management
- Supply Chain Management
  - Supply Chain Planning
  - Logistics & Transportation Management
  - Order Management
  - Price Management
- Manufacturing

E-Business Suite is an extensive and modular system, with its main goal being to cover most of the customer's needs.. Oracle provides other ERP packages and solutions, such as PeopleSoft Enterprise, Siebel, Primavera and JD Edwards EnterpriseOne [39].

Along with SAP and other top ERP vendors, there was the need to supply software for mid-sized companies, and so Oracle developed an E-Business Suite version for this market. [40] [41].

Oracle Visualization

Oracle also provides visualization tools for their products, fully customizable to fit customers' needs and improve the customization experience. The knowledge of the business is the key for creating interesting views, with statistics and KPIs.

This tool was introduced with Oracle's advanced relational database models and is integrated in Oracle's database systems, allowing dashboard creation and different views over data, as displayed in Figure 3.18.

Another tool for planning and visualization, which integrates this module, is Oracle's Advanced Planning Command Center [42], already setup with several views as shown in Figure 3.19, also including other useful dashboards such as *Sales and Operations Planning Dashboard* and *Supply Chain Analyst Dashboard*. Oracle's visualization package provides an easy way of overviewing all possible statistics over data.

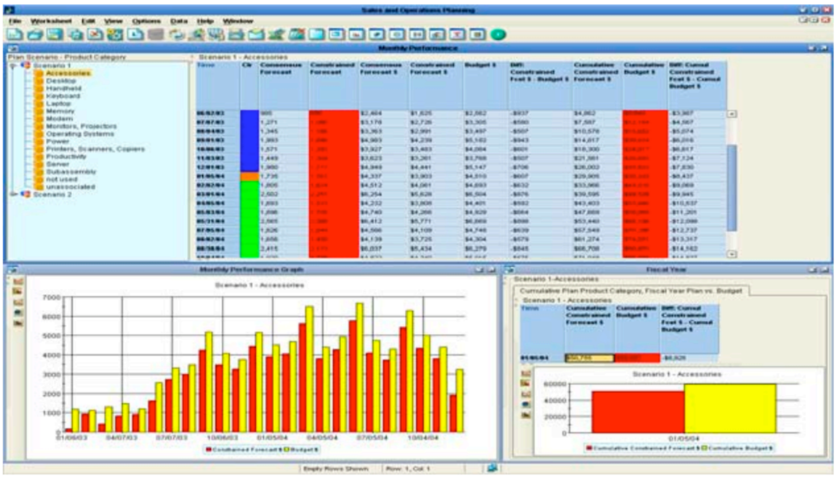
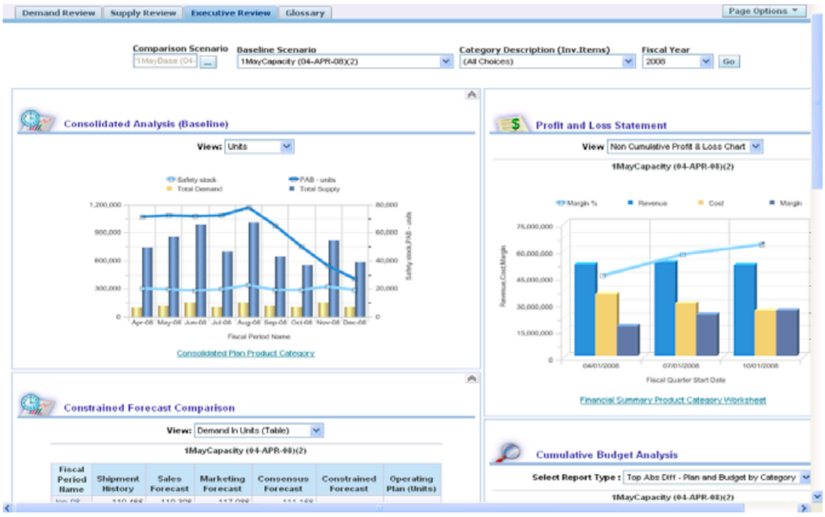


Figure 3.18: Oracle Dashboard Example

3.3.1.3 SAGE Group

SAGE was founded in 1981 in North East England by Graham Wylie, David Goldman and Paul Muller, for developing accounting software for small companies. Today SAGE is one of the largest in software development for business management for mid-sized companies, employing 14,500 people and with revenues around £1,300 million.

SAGE developed two main ERP systems, SAGE Accpac for small- and mid-sized companies, and SAGE X3 for mid-to-large businesses. Accpac is meant to help mainly



Comparison Scenario: 1MayBase (04-

Baseline Scenario: 1MayCapacity (04-APR-08)(2)

Category Description (Inv.Items): (All Choices)

Fiscal Year: 2008

View: Units

View: Non Cumulative Profit & Loss Chart

1MayCapacity (04-APR-08)(2)

View: Demand In Units (Table)

1MayCapacity (04-APR-08)(2)

Fiscal Period Name	Shipment History	Sales Forecast	Marketing Forecast	Consensus Forecast	Constrained Forecast	Operating Plan (Units)

Select Report Type : Top Abs Diff - Plan and Budget by Category

1MayCapacity (04-APR-08)(2)

Figure 3.19: Oracle Advanced Planning Command Center – Sales Planning Review

in accounting by organizing everything into one manageable system, while X3 targets the business intelligence aspect of ERP systems.

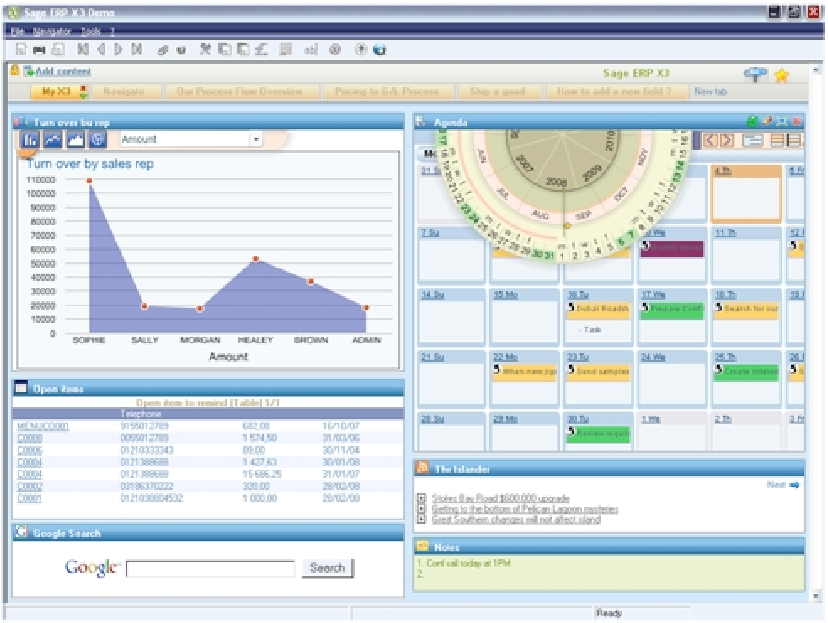
X3 targets for larger businesses and therefore aims to achieve real-time system management. It is split into six main groups:

- Manufacturing
- Sales
- Purchasing
- Inventory
- CRM
- Finance

SAGE Visualization

SAGE X3 ERP platform includes the same dashboard creation capabilities as seen in SAP's and Oracle's ERPs, allowing the user to setup his own working environment as he/she sees fit. By simple drag-and-drop the user may configure views to support whatever data fields he/she finds interesting to work on, as displayed in Figure 3.20.

SAGE X3 also contains some views already setup for aiding the planner in the decision making process, Figures 3.21 and 3.22 demonstrate some of these views.



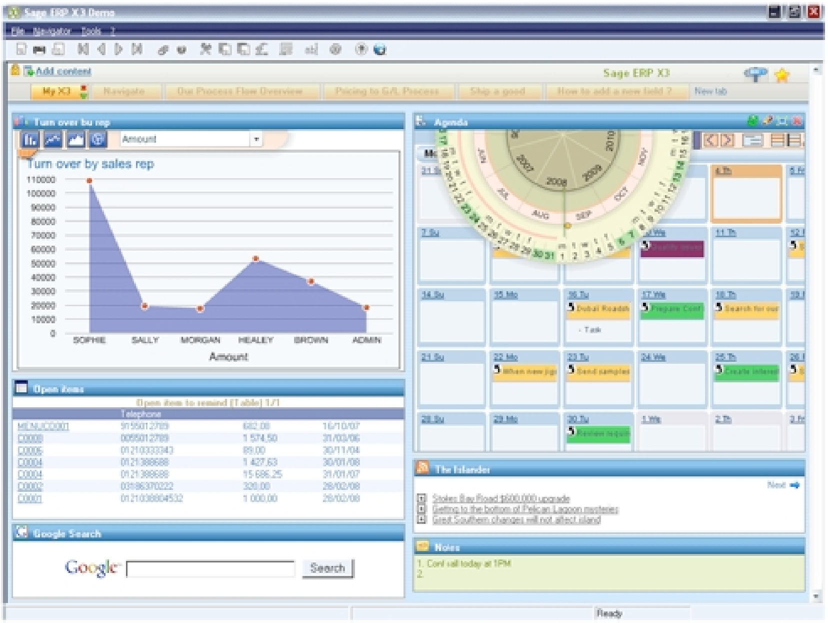
Open item to remind [1 table] 1/1

	Telephone		
MENUCO0001	9155012789	682.00	16/10/07
C0008	0055012789	1 574.50	31/03/06
C0006	01210333343	89.00	30/11/04
C0004	0121388688	1 427.63	30/01/08
C0004	0121388688	15 686.25	31/01/07
C0002	03186370222	320.00	28/02/08
C0001	01210388804532	1 000.00	28/02/08

Stokes Bay Road $600,000 upgrade
Getting to the bottom of Pelican Lagoon mysteries
Great Southern changes will not affect island

1. Conf call today at 1PM
2.

Figure 3.20: SAGE Personal Agenda Dashboard



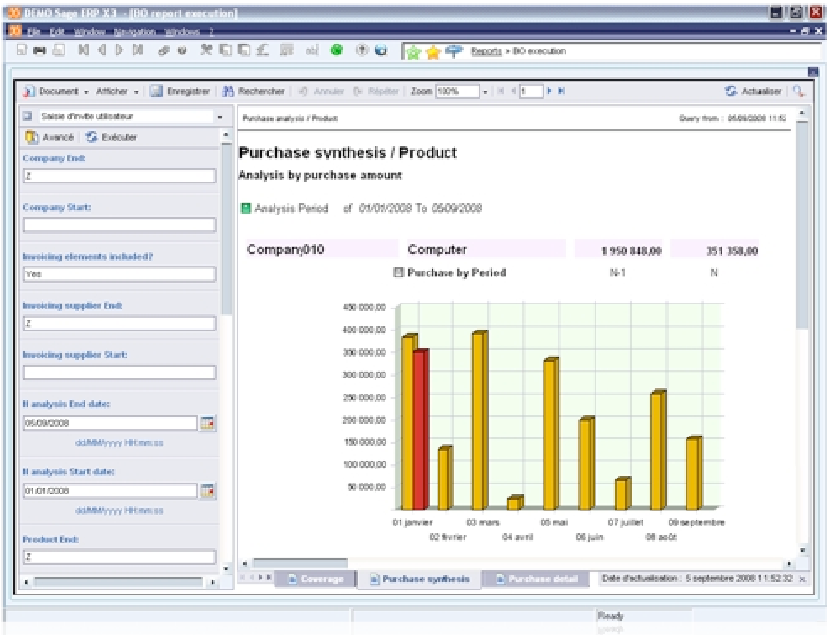
Open item to remind [1 table] 1/1

	Telephone		
MENUCO0001	9155012789	682.00	16/10/07
C0008	0055012789	1 574.50	31/03/06
C0006	01210333343	89.00	30/11/04
C0004	0121388688	1 427.63	30/01/08
C0004	0121388688	15 686.25	31/01/07
C0002	03186370222	320.00	28/02/08
C0001	01210388804532	1 000.00	28/02/08

Stokes Bay Road $600,000 upgrade
Getting to the bottom of Pelican Lagoon mysteries
Great Southern changes will not affect island

1. Conf call today at 1PM
2.

Figure 3.21: SAGE Predefined Views Example 1



Company End: Z

Company Start:

Invoicing elements included? Yes

Invoicing supplier End: Z

Invoicing supplier Start:

II analysis End date: 05/09/2008 ddMMyyyy HH:mm:ss

II analysis Start date: 01/01/2008 ddMMyyyy HH:mm:ss

Product End: Z

Purchase analysis / Product

Query from : 05/09/2008 11:52

Purchase synthesis / Product

Analysis by purchase amount

Analysis Period of 01/01/2008 To 05/09/2008

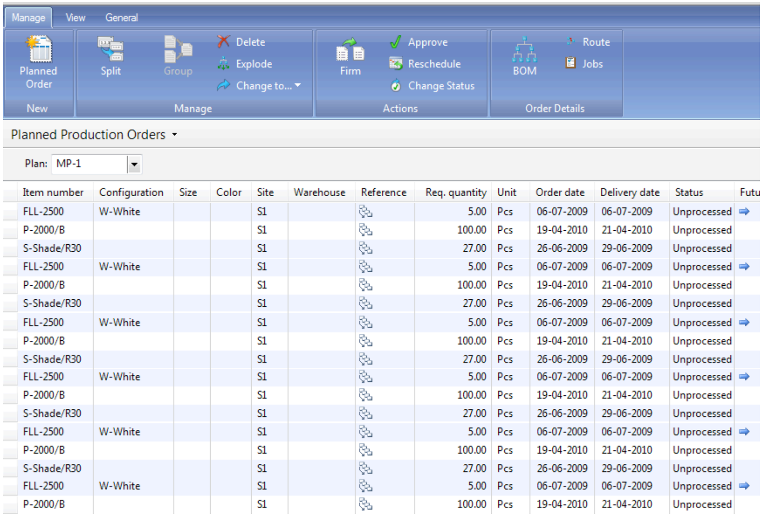
Company010	Computer	1 950 848,00	351 358,00
	Purchase by Period	N-1	N

Date d'actualisation : 5 septembre 2008 11:52:32

Figure 3.22: SAGE Predefined Views Example 2

3.3.1.4 Microsoft Dynamics AX

Microsoft Dynamics AX visualization solutions are limited, and most of the framework is composed of table-like views, as displayed in Figure 3.23. For capacity visualization there is a 3D view, which relates to the concepts previously described (Subsection 2.1.2).



Planned Production Orders

Plan: MP-1

Item number	Configuration	Size	Color	Site	Warehouse	Reference	Req. quantity	Unit	Order date	Delivery date	Status	Futu
FLL-2500	W-White			S1			5.00	Pcs	06-07-2009	06-07-2009	Unprocessed	→
P-2000/B				S1			100.00	Pcs	19-04-2010	21-04-2010	Unprocessed	
S-Shade/R30				S1			27.00	Pcs	26-06-2009	29-06-2009	Unprocessed	
FLL-2500	W-White			S1			5.00	Pcs	06-07-2009	06-07-2009	Unprocessed	→
P-2000/B				S1			100.00	Pcs	19-04-2010	21-04-2010	Unprocessed	
S-Shade/R30				S1			27.00	Pcs	26-06-2009	29-06-2009	Unprocessed	
FLL-2500	W-White			S1			5.00	Pcs	06-07-2009	06-07-2009	Unprocessed	→
P-2000/B				S1			100.00	Pcs	19-04-2010	21-04-2010	Unprocessed	
S-Shade/R30				S1			27.00	Pcs	26-06-2009	29-06-2009	Unprocessed	
FLL-2500	W-White			S1			5.00	Pcs	06-07-2009	06-07-2009	Unprocessed	→
P-2000/B				S1			100.00	Pcs	19-04-2010	21-04-2010	Unprocessed	
S-Shade/R30				S1			27.00	Pcs	26-06-2009	29-06-2009	Unprocessed	
FLL-2500	W-White			S1			5.00	Pcs	06-07-2009	06-07-2009	Unprocessed	→
P-2000/B				S1			100.00	Pcs	19-04-2010	21-04-2010	Unprocessed	
S-Shade/R30				S1			27.00	Pcs	26-06-2009	29-06-2009	Unprocessed	
FLL-2500	W-White			S1			5.00	Pcs	06-07-2009	06-07-2009	Unprocessed	→
P-2000/B				S1			100.00	Pcs	19-04-2010	21-04-2010	Unprocessed	

Figure 3.23: Microsoft Dynamics AX - Planned Production Orders

The dimensions of this capacity view are capacity load, time and resource group, as displayed in Figure 3.24, which provide a clearer idea of what resources are loaded. One of the problems is that it does not relate directly to actual sales orders, production orders or any other customer action.

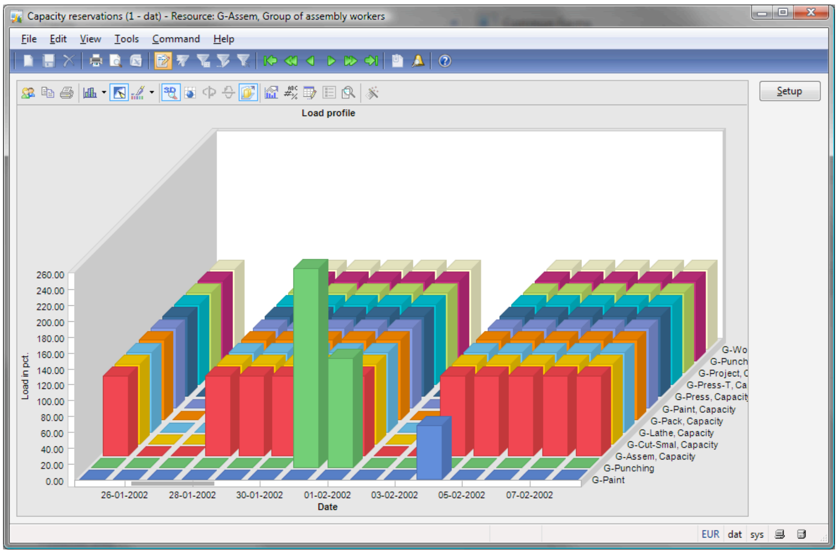
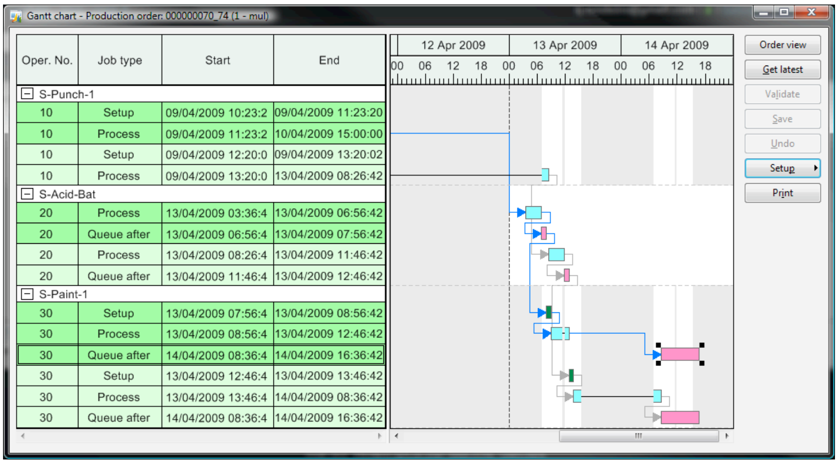


Figure 3.24: Microsoft Dynamics AX - Capacity Overview

The final visualization tool in Dynamics AX is a Gantt chart view for displaying *Job Scheduling*, which is a requirement for every ERP system on the market, as displayed in Figure 3.25.



Oper. No.	Job type	Start	End
⊟ S-Punch-1			
10	Setup	09/04/2009 10:23:2	09/04/2009 11:23:20
10	Process	09/04/2009 11:23:2	10/04/2009 15:00:00
10	Setup	09/04/2009 12:20:0	09/04/2009 13:20:02
10	Process	09/04/2009 13:20:0	13/04/2009 08:26:42
⊟ S-Acid-Bat			
20	Process	13/04/2009 03:36:4	13/04/2009 06:56:42
20	Queue after	13/04/2009 06:56:4	13/04/2009 07:56:42
20	Process	13/04/2009 08:26:4	13/04/2009 11:46:42
20	Queue after	13/04/2009 11:46:4	13/04/2009 12:46:42
⊟ S-Paint-1			
30	Setup	13/04/2009 07:56:4	13/04/2009 08:56:42
30	Process	13/04/2009 08:56:4	13/04/2009 12:46:42
30	Queue after	14/04/2009 08:36:4	14/04/2009 16:36:42
30	Setup	13/04/2009 12:46:4	13/04/2009 13:46:42
30	Process	13/04/2009 13:46:4	14/04/2009 08:36:42
30	Queue after	14/04/2009 08:36:4	14/04/2009 16:36:42

Figure 3.25: Microsoft Dynamics AX - Job Scheduling Gantt Chart View

3.4 Graphical Visualization Development Software

There are several frameworks that meet the requirements for the development of a visualization tool. If the system is web based, it can be developed in PHP, ASP.NET, Ruby, Python, Java, and use a number of plug-ins to achieve 2D and 3D functionalities. If it is developed for desktop usage, then Java, C/C++ or .NET would be able to perform the task. For a fully integrable application, one software framework was required: Microsoft .NET Framework.

3.4.1 .NET Framework

Developed by Microsoft, the .NET Framework is the most commonly used tool to develop software for Microsoft Windows Operating Systems. Its package includes several solutions, for desktop or web development, along with an impressive library full of solutions to improve and facilitate the programming experience [43] [44] [45] [46].

The concepts behind the .NET framework are as follows:

- Provide a stable and consistent object-oriented programming environment, wether the execution is local, through a network or internet, or executed remotely;
- Provide a better integration, version conflict reduction and safe execution of code through a common code-execution environment;
- Increase performance through an optimized common code-execution environment;
- Provide the developer a consistent environment for web and desktop development.

The main concepts and components of .NET are the following:

- **Common Language Runtime**

  Although .NET provides several programming languages, such as *Visual Basic* and *Visual C#*, the code follows a conversion into a common code, which is then compiled and executed in a virtual machine. The benefits from this design are interesting; the developer may use whatever language he/she is accustomed to, or which ever fits best to solve a certain problem, knowing that the outcome common code will execute in the same way, and allowing cross-language development at a new level. Due to the use of a *Garbage Collector*, there is automatic handling of objects and references, and their disposing. All of this results in performance improvements, extensibility and fully integrable components [44] [47] [45].

- **Common Type System**

  The CTS is a set of rules, defining how types are declared, used and managed in the runtime. These rules also define integration levels and interaction between all .NET languages, turning cross-language development into a reality [44] [48] [46].

- **Cross-Language Interoperability**

  Although .NET is cross-language, in order to develop code for different application environments, developers must follow the Common Language Specification (CLS).

By following this set of rules, they assure their software is CLS-Compliant [44] [49] [46].

- **Class Library**

  Along with the *Common Language Runtime* the *Class Library* is the other main component. It provides a set of object types and tools for developing applications. These range from command-line applications, graphical user interface applications, or web based applications, with libraries designed for xml interactions, mathematics, events handling, and so forth. One of the newest features is the Language Integrated Query (LINQ) component, which reduces database query complexity and provides an easy to use interface, between databases and .NET components [44] [45].

- **Security**

  The .NET framework provides a set of rules for safe code execution and controlled access to resources; handles new software installations, preventing interference with older versions [44] [46].

- **Managed Code**

  .NET uses the concept of *managed code*, which ensures that developers do not have to concern with memory allocation, tracking object references or disposal. Any object is created by using a high level abstract construct, having its allocation, tracking and disposal performed by the virtual machine [44] [50] [46].

- **Compilation And Execution**

  Every .NET's programming language code is transformed into Microsoft Intermediate Language (MSIL) code which is executed using the Common Language Runtime (CLR) virtual machine. MSIL is managed code, which means the underlying layer handles object reference and disposal [51] [46]. It is complied using a *just-in-time* (JIT) compiler, whose task is to reduce memory footprint and increase performance. The compiling and execution procedures are as follows:

  1. Compiling: The code undergoes two main steps before it is executed; first a type check is performed to ensure type safety, if everything is accordingly it is transformed into the virtual machine code.
  2. Execution: The CLR converts the MSIL code into assembly machine code on demand, using a JIT compiler; this reduces the memory footprint of the program and reduces startup times. When the program starts only the needed routines are loaded and stores into memory and as routines are executed they will be placed into memory.

3.4.1.1 Programming Languages

The .NET Framework provides several languages for development, all sharing the Common Language Infrastructure (CLI) which converts specific language code into MSIL. This has the advantage of similar execution times, while also allowing the developer to choose the language that mostly suits his programming style or needs.

Visual Basic .NET

Visual Basic (VB) is the oldest Microsoft event-oriented programming language, with some object-oriented basic support. It derived from *Basic*, which was invented my Bill Gates and Paul Garner Allen for deployment in MITS's Altair 8800 system. The name is self explanatory, its key goal is simple and fast development of GUI applications, also providing several libraries for interfacing with different technologies.

With the introduction of the .NET Framework, VB went from an event-oriented to a fully object-oriented programming language, and it was then renamed to: Visual Basic .NET (VB .NET). The main goal resides, provide a programming environment which is easy to program in and has a fast learning process [52] [53].

Visual C#

Fully designed and created by Microsoft, and unlike VB .NET, Visual C# was thought up to be an object-oriented programming language from the start. Its design is based on C++ syntax merged with Java architectural concepts, but built to run on Windows OS. This merge proved to be a success for Microsoft due to the fast learning curve from C++ and Java developers. In Microsoft's Visual C# website there is an easy guide for users migrating from C++ or Java, and a *How-To* on easy migration of applications from Java [54] [55] [56].

Along with VB this is the most used language in the .NET development package.

Visual C++

Visual C++ is a Microsoft implementation of the world's most popular development language, C++. It enables users to build Microsoft Windows-based applications, Microsoft .NET-based applications, Web applications, and solutions for thin-clients and smart-clients mobile devices. It integrates several components: Visual C++ 2008 compiler tools, Visual C++ Libraries and Visual C++ Development Environment. It is based on C++, which enables C++ developers to migrate with relative ease [57].

Visual J#

Visual J# is a Microsoft implementation of Java, following its concepts and syntax. It makes transitioning from Java or J++ (which was the previous version of Microsoft's implementation of Java) to .NET a simple task, enabling yet another programming style into the .NET framework [58].

JScript

JScript is the only Microsoft supported scripting language. It incorporates features from the .NET Framework, and as other modern scripting languages, supports typed and typeless variables, late- and early-binding, packages, cross-language support and objects [59].

ASP.NET

Active Server Pages (ASP) was one of the first steps Microsoft took into the web programming languages world, and ASP.NET is an evolution of ASP for the .NET Framework. ASP was similar to other scripting web programming languages around, such as PHP, Ruby, Python, and so forth. With ASP.NET the experience has changed.

ASP.NET separates the business logic from the visual logic, giving developers a similar programming perspective to that of desktop languages. Another advantage is the fact that it supports .NET technology so developers may use desktop developed components for web usage, and obey one of the first programming principles: code reutilization [60] [61].

3.4.1.2 Windows Forms

The most commonly used graphical software development solutions are based on the *Windows Forms* (WinForms) concept. Design as a set of classes, within the .NET Framework, it allows fast development of rich graphical applications for Windows OS.

When announced in the year 2000, Windows Forms was the center of media attention, but the final release was only available in 2002, when it proved to change the way Windows applications were developed.

Before Windows Forms, Microsoft Foundations Class (MFC) library was the development route for GUI applications. MFC development is still tightly coupled with direct usage of the main Windows API (Win32), which proved to be an issue in programming due to the low level instruction management developers had to endure. Another problem was that .NET's main development language, VB, relied on ActiveX Controls or COM components for some platform services, which meant when a new feature was released

VB developers would have to wait until C++ developers, wrote the designated control. This is because C++ is able to work with unmanaged code and was able to use these releases straight away. This bottleneck in development along with complexity of MFC usage, complicated life for GUI developers.

The .NET Framework introduced a new concept, a single API: Windows Forms. With a new level of abstraction, which interacts with Win32, developers could now focus on creating their softwares without having to directly interact with the underlying operating system. Another asset for VB developers is that .NET introduced the concept of a cross-language system, where developers may use any language with whatever controls, avoiding ActiveX bottleneck.

The two main elements that provided success where MFC had failed: the CLR, enabling all languages to share a single runtime based on the same type system and runtime semantics, and the .NET Framework API abstraction, allowing developers to use the Class Library to interact with the whole system [44].

There are several features within WinForms that make it so extensible, flexible and powerful; the main changes of WinForms to MFC are explained below:

- **Controls**

  When creating a new class the developer knows that it derives from the base class *Object*, which implements certain methods for easy access. This same logic applies for creating a different type of user interface, all deriving from the *Control* class, which most user-defined controls indirectly inherit, as seen in Figure 3.26.

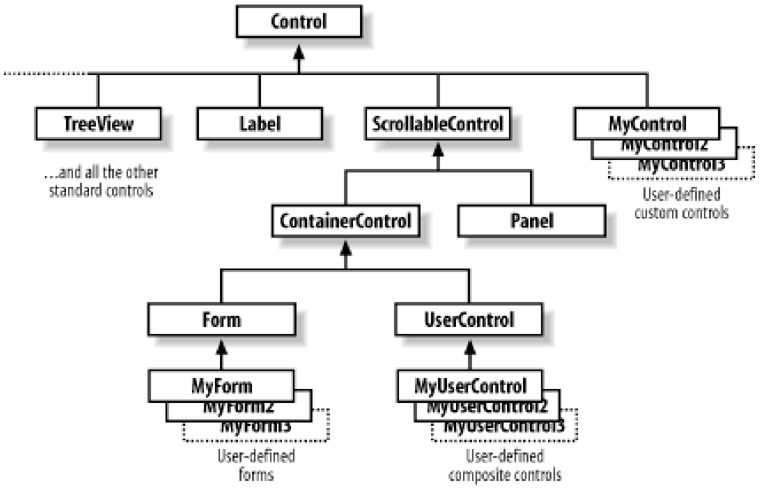


Figure 3.26: Control Class Hierarchy

  All visual elements are extended from this base class, simplifying their representation by providing visual features such as background color, caption, event handling and layout management.

Windows Forms controls' layout is managed through a very flexible system, providing tools for automatic or explicit control placement (location and size) [44].

- **Drawing and GDI+**

  Graphics Device Interface Plus (GDI+) is an evolution of GDI, Microsoft Windows application programming interface. This component handles Windows OS graphical objects representation and transmission to output devices, such as monitors and printers. Microsoft Win32 applications were based on GDI, and unlike its predecessor, GDI+ interacts directly with hardware devices providing graphics acceleration, and an interface to applications.

  While the *Control* class handles the positioning and provides several other features, when designing a custom control the user must also implement the *OnPaint* method, which is responsible for drawing. Every control owns a part of the screen and is drawn into this area [62] [44].

- **Data Binding**

  The goal of WinForms is to allow the programmer to develop rich user interfaces for data manipulation. Usually the visual elements are not responsible for handling data, and their location is usually other objects or database systems. The .NET framework introduced the concept of data binding.

  There are numerous sources for providing data to replenish controls; each data source is associated with a control property, making sure that the control becomes lightweight and does not handle data directly.

  Data Binding with certain controls may also be used to interact with the data. Although the controls' main purpose is for viewing, the developer is able to program an input interface for interactions with the data source, allowing changes [63] [44].

3.4.1.3 Windows Presentation Foundation

With the evolution of Windows Operating Systems and to take advantage of the GDI+ improvements of Windows Vista, a new framework and interface development API were released, .NET Framework 3.0 and Windows Presentation Foundation API.

Windows Presentation Foundation (WPF), represents a breakthrough in programming, due to the separation between visual and business logic. Windows Forms cleared the path with the introduction of GDI+, and WPF went even further by introducing DirectX architecture into graphical user applications development, along with several new technologies and techniques; among the best for UI development are: rendering engine supporting 3D graphics, API for animations and storyboards, XML-based markup language for building simple interfaces [64] [65], as seen in Figure 3.27.

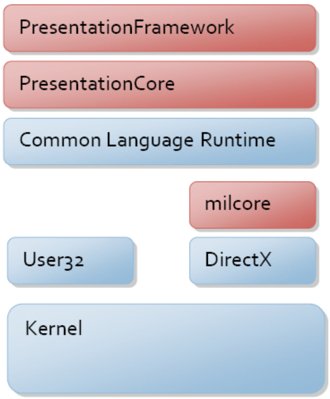


Figure 3.27: WPF Architecture [2]

XAML

Unlike most APIs, WPF introduces a concept to the desktop development only seen in web development, the user interface (UI) and business logic separation. To achieve this a concept similar to HTML was introduced, but based on XML, the Extensible Application Markup Language (XAML) [66].

Similar to WinForms architecture, every WPF visual object is extended from the *Control* base class. Most controls in the framework may be setup from VB/C# code or directly from XAML, and user defined controls may also follow the right guidelines for XAML instantiation. This provides an easy and fast way to create a visual appealing UI without much knowledge.

Through XAML the developer may instantiate controls, setup their layout and style, including positioning, size, color scheme, and so forth.

2D/3D Rendering

All rendering in WPF is done through the DirectX engine, allowing efficient hardware and software rendering, along with powerful 3D graphics [2]. WPF programming introduces the following benefits [67]:

1. **Resolution and device-independent graphics:**
   WPF uses a device independent resolution system; its units are based on dots-per-inch (DPI) instead of direct pixel mapping. This achieves a system which automatically scales each pixel to the measure required for correct displaying.

2. **Improved Precision:**
   WPF's layout system uses doubles instead of floats for every calculation, such as:

positioning, 2D/3D transformations and opacity values. It also introduces wide color gamut RGB (scRGB).

3. **Advanced graphics and animation support:**
   WPF introduces a concept similar to Adobe Flash storyboard system for animations, simplifying building complex animated systems. The developer only needs to concern on what the system must achieve rather then focusing on scene processing, rendering loops, and bilinear interpolation. WPF also provides 2D/3D hit-testing support, an integrated animation system (for simple animations and complex storyboards), and full alpha compositing support.

4. **Hardware acceleration:**
   Making use of tight coupling with DirectX achieves a more efficient and reliable system, also reducing CPU processing through direct hardware acceleration.

The rendering capabilities of a simplified version of DirectX, make it possible to develop visually appealing GUIs unlike never before.

3D Graphics

For 3D rendering WPF provides an environment similar to a simplified version of DirectX, without the same power such as ray-tracing, but for UI development it is able to achieve interesting results.

WPF does not provide a full 3D environment, but rather 3D content in a 2D world, bridging the gap with the *Viewport3D* control. This UI element follows the two-dimensional visual element structure, using the same layout positioning, size and color system. Within this control everything changes, similar to looking out through a 2D window into a 3D world; the coordinate system is fully 3D, providing camera and light support along other features.

- **Coordinate System:**
  The 3D coordinate system in WPF is a bit different then other 3D systems, following the right-handed coordinates system; the positive x-axis values are from right to left, the positive y-axis values are from the screen to the viewing screen and the z-axis are upwards, as seen in Figure 3.28.

- **Cameras And Projections:**
  There are several types of cameras for different purposes: *ProjectionCamera*, *MatrixCamera*, *PerspectiveCamera* and *OrthographicCamera*; the latter two are the most used. The *PerspectiveCamera* is the simpler one because it relates directly with the human 3D viewing concept - the further away objects are the smaller they

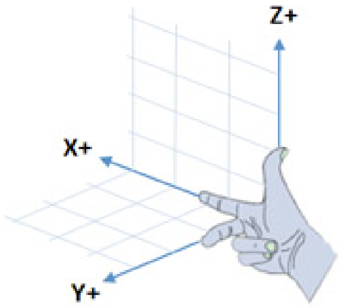


Figure 3.28: WPF 3D Coordinate System [3]

appear. There is the *NearPlaneDistance* and *FarPlaneDistance* which define how close and far away objects have to be to still be visible, and the third important parameter is the *FieldOfView*, defining the viewing angle. The *OrthographicCamera* has a different approach, an object is always rendered in the same size, regardless of how far away it is. This produces unnatural-looking perspectives and is sometimes complicated to setup and use. Figure 3.29 illustrates the difference in perspective between both cameras. Each camera is also defined by a *Position*, *UpDirection* and *LookDirection*. [64] [4] [3]

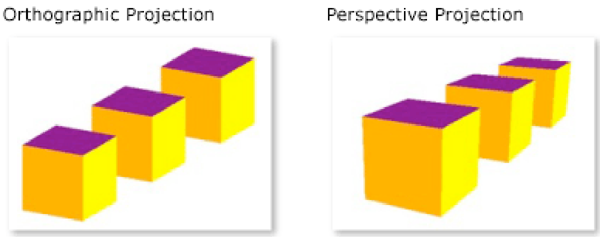


Figure 3.29: WPF Cameras [4]

- **Lights:**
  WPF has a broad light support system, based on four types of lights [4]:
  - *AmbientLight*: illuminates all objects uniformly, similar to sunlight from every direction.
  - *DirectionalLight*: illuminates objects like a distance light source and provides a direction vector but no location.
  - *PointLight*: illuminates objects like a near light source, providing light position and direction vector.
  - *SpotLight*: the illumination is similar to the *PointLight* but it also provides two variables to define the light's *InnerConeAngle* and *OuterConeAngle* for setting up the illumination cone.

- **Models:**
  For modeling WPF supports meshes capable of creating models from a collection of vertices, supporting textures, materials and alpha blend. Most shapes are formed using the most primitive closed figure: a triangle. For positioning, the models follow the 3D coordinate system and for displaying it is necessary to setup the normals. [4]

WPF presents a set of features which turn it into a complex 3D graphics system, unusual in GUI development, changing and enriching the programming experience.

Animations

The WPF animation system follows two different paths: simple animations and complex scenes. It is easy to create simple animations over UI elements due to their animation properties and the different available animation APIs. If the developer wishes to create a more complex scene, following a complex timeline, the WPF animation system provides the storyboard concept similar to video edition software and Adobe Flash.

3.4.2 Microsoft Expression Blend 2

Expression Blend 2 is a Microsoft Integrated Development Environment (IDE) for easy development of WPF applications, with the potential of Visual Studio but targeted towards design instead of the programming experience. It allows for object drag-and-drop, positioning and style setup, along with 3D models import for direct usage. Expression Blend 2 also provides tools for XAML editing, code debugging and development; its main features are [68]:

- Modern Interface
- Design for Desktop and Web
- Over 30 Controls and Containers
- Professional Timeline
- Styling and Customization
- True 3D
- Easy Databinding
- XAML Editing
- Visual Studio Integration

3.4.3 Blender

Blender is an application for 3D development, which contains: a modeling editor, animation development, physics and particles system, realtime 3D/game creation, imaging and compositing, shading and so forth. [69]

Blender is a powerful tool for 3D modeling and when used with Microsoft Blend's import capabilities, simplifies the process of creating complex 3D models and using them directly in WPF. After the model is created in Blender, it is imported into Microsoft Blend 2, which converts it into a XAML file with all the required vectors.

3.5 Summary

The ERP business is complex and requires much investment on developing companies, a goal that must also be applied to systems deployment in their customers. This strainful and arduous integration process must be successful or it may result in the target company's bankruptcy.

The visualization approaches used in scheduling and planning, not only in ERP, are somewhat limited, and there are currently no alternatives to the old ”Gantt Chart” view over scheduling and other limited chart views for planning. Although ERP systems provide rigorous integration of the customer's business model, as of yet, there is no tool that allows interactive planning and scheduling overviews, capable of transversing from the top view, of a long period, down to the smallest detail in one integrated view.

With today's technology it becomes imperative to increase investment on interactive visualization, helping the user of a complex system, such as an ERP, detect possible problems, therefore gaining a competitive edge. Innovative approaches to problem solving, and bottleneck detection are based on advanced artificial intelligence algorithms; these are also limited in capturing systems' issues. When integrated into ERP systems, due to the large amounts of data provided, aggregated views are required and sometimes details are lost. Considering that the customer, in this case the planning manager, has the most knowledge over the business, it is possible to assume that he/she will know where to start looking; therefore, by providing a clearer business overview, the chances of success in problem solving increase.

Dynamics AX is particularly behind in visualization. Due to a late market entrance, Microsoft has been focusing on making sure the business logic goes beyond what other vendors have, disregarding visualization. Most Dynamics tools are table-like views, with only a few exceptions for focusing on particular problems, increasing this application's added value, and making it imperative for gaining a competitive edge.

This research over the top contenders in the ERP business such as: SAP, Oracle, SAGE and Microsoft Dynamics AX, showed that today's technological achievements in visualization remain at the statistical level, allowing the planner to make decisions, but not providing a clear solution. Another part of the research was based on introducing some innovative ideas, showing that thinking *outside-the-box* achieves worthy results.

The final sections of this chapter refer to the study over technologies useful to the project's design and code development. The .NET Framework is a requirement; furthermore a careful research over WinForms and WPF, reveals that is actually one of the top technologies in the market. Microsoft Windows Presentation Foundation API's features and capabilities allow this project to thrive and achieve impressive visual results.

Chapter 4

Solution Specification

> *"In preparing for battle I have always found that plans are useless, but planning is indispensable."*
>
> Dwight D. Eisenhower

This chapter describes the defined solution to address the problem (Chapter 2), integrating innovative concepts gathered from the *State of The Art* research (Section 3), with production and capacity planning business logic. It presents the chosen concepts, their actions and interactions, justifying how they address the visualization problem at hand and integrate with the *Manufacturing* environment. A description on how the solution will aid in capacity problems and bottlenecks detection will also be presented throughout this definition.

4.1 Overview

When a project's focus is to develop an user interface, it is necessary to define all GUI components before its design and implementation stage. This section outlines what controls are going to be developed, along with its integration user interface (*Interactive Visualization GUI*). It contextualizes the research gathered from the *State Of The Art* chapter (Section 3), with the project requirements (Section 2), performing a required concepts and interface goals analysis, providing a solution, and justifying their choice.

4.2 Interactive Visualization GUI

4.2.1 Concepts Analysis

This GUI's goal is to be the main visual control for laying out every other component, integrating their distinct concepts into one tool.

When defining a complex user interface, two area types must be analyzed: static and contextual [31]. The first remains unchanged whatever option is taken, while the latter adjusts to the selected context. Due to this GUI's integration purposes both types will exist; the following areas have been outlined:

1. **Menu:** There must be a menu that provides access for opening production plans and production plan scenarios. Part of this menu will be *static*, and always available to the user, while part will be *contextual*, able to perform actions on the selected GUI.

2. **Filters Area:** The filters is one of the main requirements for both views; this is a *contextual* area, because it adjusts to the relevant filtering in the selected GUI.

3. **Details Area:** This area provides means for obtaining more information on the selected GUI, by clicking different visual controls, and for this reason it represents a *contextual* area.

4. **Content Area:** This a *contextual* area and is where the GUIs themselves will be displayed.

4.2.2 Solution

Following the Dynamics AX user interface development guidelines (Section 2.2.2.2), a prototyped component layout was developed and is displayed in Figure 4.1.

For assuring that all basic features are available in this design, the following explanation relates the visual concepts with the required use cases (Section 2.2.1):

- **Actions Menu:** This menu integrates two components: *Load Production Plan* and *Load Production Plan Scenario*. These are used for loading the production plans from the database into a class model. Relates with use case IV-UC01 and IV-UC07;

- **View Selector Menu:** This menu is standard for Dynamics AX and is a reference pane for selecting different user interfaces. It will allow the user to switch between the *Production Plan GUI* and the *Capacity Utilization GUI*. Relates with use cases IV-UC02, IV-UC05 and IV-UC09;

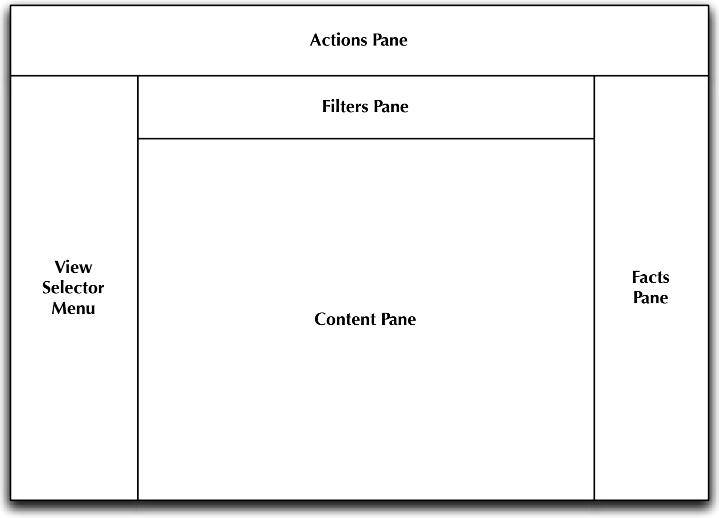


Figure 4.1: Interactive Visualization GUI Layout Prototype

- **Filters Pane:** The filters section adjusts to the GUI environment currently loaded, providing filtering mechanisms related to what is being displayed in the *Content Pane*, also including the search feature from the *Production Plan GUI*. Relates with use cases IV-UC03, IV-UC04, IV-UC06, IV-UC08, IV-UC10 and IV-UC11;
- **Facts Pane:** This section will display the currently loaded production plan, as well as the selected *Production Orders Details*. Relates with IV-UC03;
- **Content Pane:** The central section is where the *Production Plan GUI* and the *Capacity Utilization GUI* will be displayed, along with the *Date Selector GUI*. Relates with IV-UC01, IV-UC07 and IV-UC12.

4.2.3 Solution Justification

In order to be compliant with the Dynamics AX user interface development guidelines (Section 2.2.2.2), the analyzed concepts need some adjusting. The definition, conceived a single menu area, but following the guidelines two menus emerged: the *View Selector Menu*, that provides access to distinct user interfaces, and the *Actions Menu* for performing actions in the selected user interface. From an integration perspective, the *Actions Menu* will maintain its static and contextual purpose; some options will be available in both GUIs, while some will differ.

All other concepts remain unaltered, and obey the guidelines completely: the *Content Pane* displays the currently selected interface, the *Filters Pane* provides filtering mechanisms for the GUI, the *Details Pane* becomes a *Facts Pane* but follows its original goal of providing specific contextual information.

As detailed in the previous section, all defined use cases (Section 2.2.1) are available to the user, and provide a solution for the main GUI.

4.3 Production Plan GUI

4.3.1 Concepts Analysis

The defined solution merges the *Production Plan Orders List* and *Production Plan Orders Visualization Overview* GUIs into the *Production Plan GUI*. Although both views provided different perspectives, the content remained the same. This solution provides a way of relating the listed production orders with their visual representation. As both GUIs concepts are the same, only their distinct goals must be analyzed and considered.

For designing a feasible production plan and possibly optimizing it, the user (henceforth referred to as *planner*), must considerer all concepts. Due to the numerous variables available in a production plan and to avoid cluttering the view, it is imperative to define priorities. The most important concepts are the high priority, which will be present in the standard view; the medium and low priority concepts will be available through features or filters:

1. **Late Orders:** This is the most important factor for the planner, to find out what orders are late, how late and why. If no orders are late, then from his/hers perspective the production plan and supply chain are working perfectly;
   Priority: High;

2. **Orders:** The planner must analyze what are the next orders to go for production, which may be for customers, stock, or other purposes. According to the number of orders and available resources, the planner tries to make sure delivery times are met;
   Priority: High;

3. **Resources:** The planner must take into consideration the available resources (personnel, tools, machines, vendors or material); this concept is available in the Capacity and Utilization view;
   Priority: Concept does not apply to view;

4. **Lead Time:** Besides available resources, the planner must also consider the time it takes to produce an item;
   Priority: High;

5. **Customer Value:** Although the *customer always comes first* and everyone should have the same priority, certain customers will present a greater value for the company, due to a longer lasting relation, costumer turnover, and so forth;
   Priority: Medium;

6. **Market Value:** The company's strategy may be directed into satisfying a specific market's demand, making it important to know which orders are for which markets; Priority: Medium;

7. **Prioritize:** According to customer value the planner may reorganize the orders, resulting in other orders being postponed.
   Priority: Concept does not apply to view;

This analysis, along with the requirements (Section 2.2), helps define the visual concepts for building the *Production Plan GUI*, and also set the main goal: to provide an overview on the production planning status.

4.3.2 Solution

Several charts and innovative approaches were researched (Sections 3.1.1 and 3.1.2, respectively), to determine how the concepts described above should integrate this view.

The chosen solution is an advanced *Bubble Chart*, integrating several features and filters, as seen in Figure 4.2, to fulfill the specified use cases and features. In a bubble chart (Section 3.1.1.8), there are four variables that must be defined:

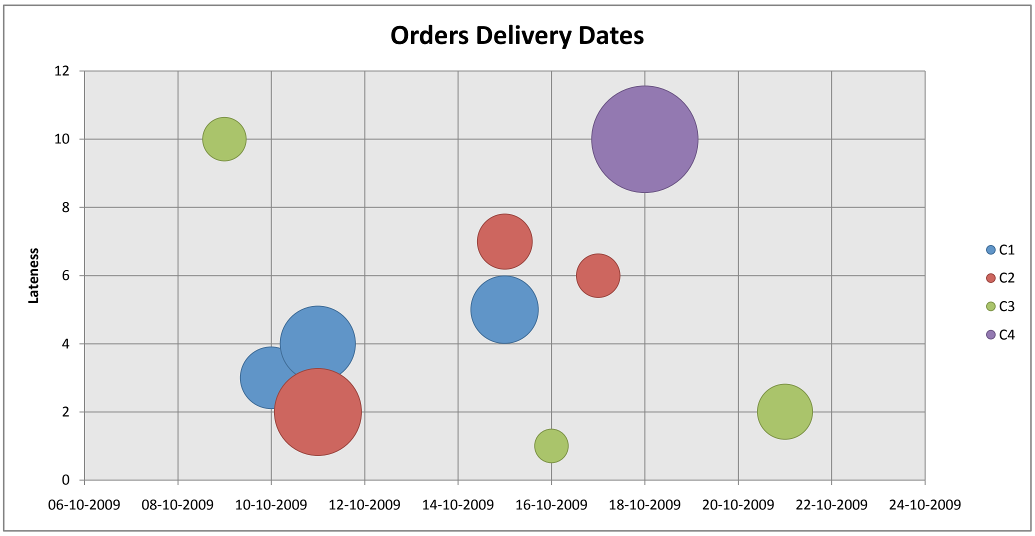


Figure 4.2: Production GUI Chart View Prototype

1. **Bubble:** Each bubble will represent a single production order;

2. **X Axis:** The horizontal axis will represent a time reference, from the left to the right;

3. **Y Axis:** The vertical axis will provide a lateness reference, growing from 0 days (on-time delivery), up to a defined value;

4. **Bubble Size:** The size of the bubble will represent order quantity;

5. **Bubble Color:** The color of the bubble will be used for filtering customer and market types, and comparing the original plan with a loaded scenario.

A full prototype version of the *Production Plan GUI* integrating the *Production Plan Chart GUI* showing how all components fit into the *Interactive Visualization GUI* is displayed in Figure 4.3.

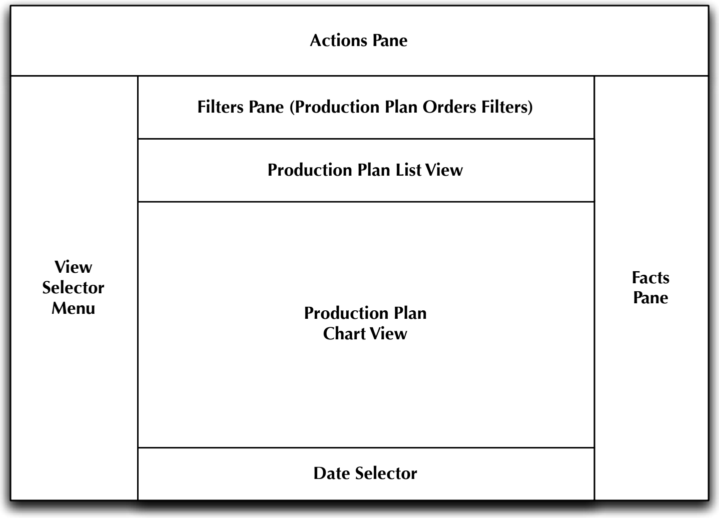

Actions Pane
Filters Pane (Production Plan Orders Filters)
Production Plan List View
View Selector Menu
Production Plan Chart View
Facts Pane
Date Selector

Figure 4.3: Production GUI Prototype integrated in Interactive Visualization GUI

Features

With this GUI the planner is able to change perspective, by filtering out some bubbles, or changing their color (providing a different meaning to the initial concepts).

1. **Drill-Down:** Although the chart represents the planner's highest concern, the production status, it is still possible to obtain more information. By clicking on a bubble, the planner will be presented with more details in the *Facts Pane*: Production Order Number, Route, Desired Delivery Date, Actual Delivery Date, Costumer Type and Market Type;

2. **Lead Time:** This representation would clutter the chart, so the solution is to present a line when the planner clicks on a bubble. This will be from the current delivery date back to when the order starts.

3. **Filters:** A visual concept easy to grasp is color changes; this is the basis for filtering in this chart. By switching filters, the planner will change the chart's meaning.

These filters will be used for differentiating costumer type, market type and for selecting production orders. Another filtering mechanism is to remove bubbles from the view, reducing its complexity and increasing the planner's focus. The available filters for customer and market types are:

(a) Customer Types: Gold, Silver, Bronze;

(b) Market Type: Africa, Asia, Europe, North-America, South-America;

4. **Date Selection:** In case the plan represents a long time frame, e.g. one year, it would be very hard to display all its production orders in one single view. The solution was to develop a filter for selecting the time period the planner wants to see. This concept is presented in section 4.5 of this chapter.

5. **Scenario Comparison:** When a different scenario is loaded the bubbles' color will change; if a production order delivery date has improved it will turn *green*, if it has worsen it will turn *dark red*, and will remain the same if there are no changes.

6. **Customer Level KPI:** For showing the Key Performance Indicators in the customer service level, a simple calculation for orders delivered per orders scheduled must be performed. It will be displayed in a simple 2D chart available in the .NET system.

7. **Production Orders List:** The previously defined GUI was integrated into this one view, and is displayed in Figure 4.3.

4.3.3 Solution Justification

The charts and innovative approaches researched in the *State of The Art* chapter, provided a path on how to define the *Production Plan Chart GUI*. Although certain innovative concepts could be appealing and integrable into a view, they would be very distant from the usual chart analysis, to which production planners are used to and know how to work with.

With the innovative concepts removed from the view's definition, still a number of possible chart alternatives remained. Although there are several charts what could be useful for providing a production plan overview, only the Bubble Chart possesses the variables for displaying all given concepts in a good way, with the following advantages:

1. **Different Concepts Integration:** The *Bubble Chart* provides five variables capable of displaying the main *Manufacturing concepts*. This relates with IV-UC05.;

2. **High Filtering Capabilities:** There are three different filter types available in the *Bubble Chart*:

   (a) *Bubble Removal:* If any bubble is removed from the chart there is no break in reasoning, the view remains the same, but with less content, attending the planner's perspective needs, which relate to IV-UC06;

   (b) *Bubble Color:* If the bubble color changes the perspective instantly changes, allowing the planner to easily focus on a specific concept;

   (c) *Period Selection:* The last filter type corresponds to a time frame selection, showing only the production orders within the designated period. This feature is enabled through a different control as specified in section 4.5 of this chapter;

3. **Scenario Comparison:** Through the bubble color change and position, the planner assimilates differences at a high pace. This relates to IV-UC08;

4. **Visually Different Concepts Integration:** In a *Bubble Chart* it becomes easy to introduce new features such as the *Lead Time*. This is due to the fact that the main visual objects are the bubbles (which are circumferences), and the introduction of lines will not disturb this visual concept.;

These advantages of the Bubble Chart were the main reason other charts were discarded. Their limitations to integrate completely different variables into one view and the cluttering factor, by decreasing visibility, when inserting other visual concepts into the same view, made them less competitive in face of the Bubble Chart

This solution supports the desired manufacturing concepts and meets this GUIs main goal of providing a production plan visual overview. The integration of the *Production Plan List GUI* and the *Production Plan Chart GUI* also provides an interesting feature, allowing the planner to relate the production plan bubbles directly with their list representation. It also provides easy access to the planner for selecting an order in the list, and discovering which bubble it represents, implementing the following use cases: IV-UC02, IV-UC03 and IV-UC04.

4.4 Capacity Utilization GUI

4.4.1 Concepts Analysis

This control's goal is to bridge the production orders' lateness with the capacity utilization, which will allow the planner to relate the concepts delivery lateness with resource utilization. By providing perspectives for comparing finite with infinite capacity, this view becomes an important tool for aiding in bottleneck detection. It is also important to outline that several time periods may be selected for comparison, providing an useful tool for comparing different time frames.

The capacity concept itself defines the number of reserved hours in one resource for a certain period of time. For simplification purposes, the used time unit is *one week*, starting on a Monday and ending on Sunday, and the number of reserved hours will be displayed in percentages. It is also necessary to outline the interesting capacity aggregation levels, that the planner will have available for filtering. These will relate with the concepts defined in the *Capacity Planning and Utilization Concepts* section (Section 2.1.2) in the *Problem Description* chapter. Analyzing such concepts lead to the following aggregation levels:

1. Aggregated Capacity by Capability: This level will aggregate capacity in the desired time period, by a specified capability;
2. Aggregated Capacity by Resource Group: This level will aggregate capacity in the desired time period, by a specified resource group;
3. Aggregated Capacity by Resource Type: This level will aggregate capacity in the desired time period, by a specified resource type;
4. Aggregated Capacity by Capability and Resource Type: This level will aggregate capacity in the desired time period, by a specified capability and resource type;
5. Aggregated Capacity by Resource Group and Resource Type: This level will aggregate capacity in the desired time period, by a specified resource group and resource type.

The aggregation level *Capacity by Capability and Resource Group*, does not make sense because they are two separate concepts, one may exist without the other.

To sum up this interface's required features are:

- Capacity Overview of Several Time Periods;
- Aggregated Filters;
- Finite Capacity Perspective;
- Infinite Capacity Perspective.

4.4.2 Solution

From the concepts' analysis and goals, a clear path was set for how to define this view - the solution was to build a *3D Bar Chart*. This chart is a widely used for displaying capacity utilization and it provides all the dimensions required: loaded capacity utilization, time frames and aggregation levels. The prototype in Figure 4.4 displays how these concepts will be setup in the chart.

When the planner desires to see what is the infinite capacity load he/she may select the appropriate action, which changes the displayed content. Figure 4.5 displays this chart's representation.

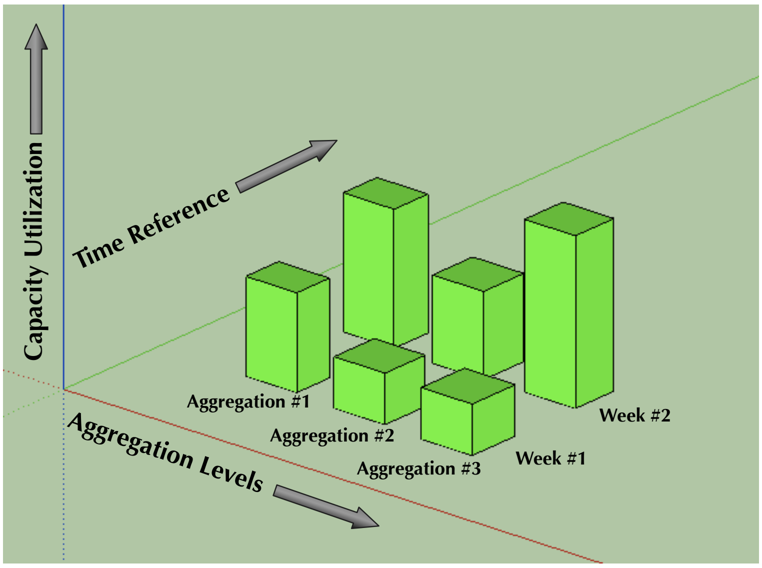


Figure 4.4: Capacity Utilization GUI Prototype - Finite View

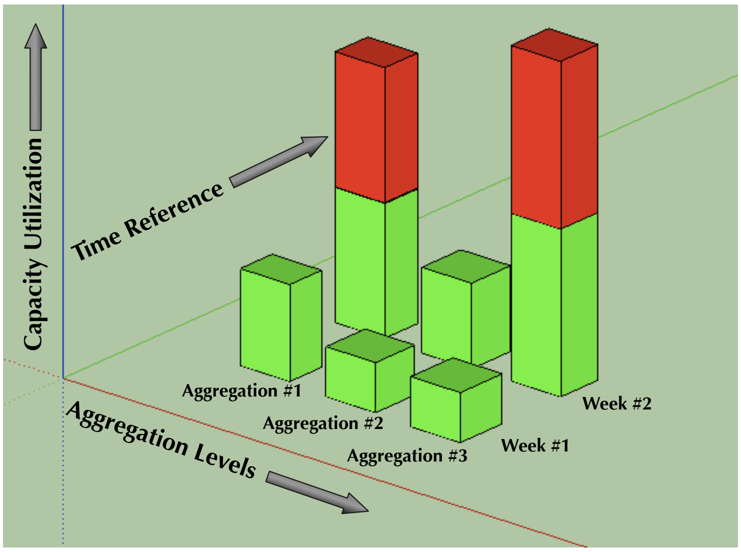


Figure 4.5: Capacity Utilization GUI Prototype - Infinite View

4.4.3 Solution Justification

The capacity comparison by different aggregations lead to one solution, a *Bar Chart*, but the time problem remained. The solution was to develop a 3D version of the bar chart, introducing the time dimension.

The innovative approaches researched in the *State of The Art* chapter represented over-engineered solutions to solve this problem. The other researched charts provided complicated views for displaying the required dimensions. Due to the common usage of bar charts, the planner has a better understanding of how to analyze its output.

The proposed solution fills all the requirements defined in the analysis process and it achieves the infinite view by altering the chart's content, displaying bars that turn red after 100% capacity is reached. This will allow the planner to have an clear view of what the resources should be if he/she wishes to meet deadlines, complying with the requirement for aiding in bottleneck detection and problem solving.

4.5 Date Selector GUI

4.5.1 Concepts Analysis

The goal for this control is to allow the planner to select the desired production plan time period, for analysis, and also relate the *Production Plan GUI* with the *Capacity Utilization GUI*. For long term planning, e.g. one year planning, it becomes impractical to have a clear view of a full year and this is where the *Date Selector GUI* comes into place.

There are several ways for selecting a dynamic time period, with the necessary concepts: a *Start Date* and an *End Date*. For this control to be innovative and still meet the required goals, it must follow some visualization rules:

- **Practical:** Simplicity is a key word for this control as its main function is to be used as a date selector, so over-engineering may be an overkill.
- **Intuitive:** Following its practicality, the planner must find out how it works without aid.

From the perspective of the manufacturing environment and considering the previously defined GUIs, other concepts must also be analyzed:

- **Look Ahead:** Apart from allowing the planner to select a specific time frame, this control should also provide more information on where to start looking for late orders, on the rest of the non selected time frame;
- **Compare Plans:** This feature must be helpful for comparing a production plan with a scenario;

- **GUIs Link:** The date selector must link both GUIs and provide a way of switching between views;
- **Production Plan GUI:** This control must allow the planner to select a time frame for analyzing production orders;
- **Capacity Utilization GUI:** This control must allow the planner to select different points in time for displaying in the capacity user interface;
- **Integrable:** This control is to be inserted into the previously defined GUIs, and as such it must be functional and represent an asset in both.

If the control follows these premises, it will become an asset tool, with its look-ahead feature, for analyzing the *Production Plan GUI*, and for synchronizing with the *Capacity Utilization GUI*.

4.5.2 Solution

As previously stated (Section 4.4.2) the control must be: practical, intuitive, provide look ahead functionalities and be integrable with the other GUIs. From the research in the *State of The Art* chapter one control emerged as key for integrating all these functionalities, the *Google Finance* date selector (Sub-section 3.1.2.1). Its core functionalities are similar but the look ahead functionality need to be altered and the solution found was the selector shown in Figure 4.6.

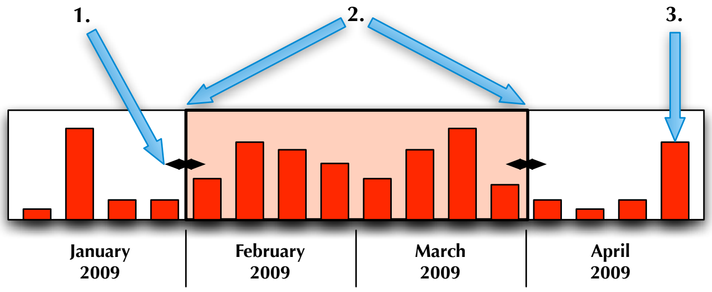


Figure 4.6: Date Selector GUI Prototype

This control works similarly to the supplied example, with a key difference: instead of displaying a similar version of the main chart's output it contains a new concept, achieved by changing from a *Line Chart* to a *Bar Chart*. The date selector is split into weeks, and the vertical bars represent the number of late orders. The concepts are the following:

1. **Date Extensor:** These controls allows the planner to expand the current selected time period and change the view.

2. **Selected Time Frame:** This control represents the currently selected time frame, and the planner may move it around without changing the frame itself (e.g. 4 weeks), but changing the start and end dates.

3. **This Week's Late Production Orders:** The planner may look at a certain week, even outside the selected time frame, and know straight away if it has too many orders.

4. **Plan Comparison:** This control displays a two series bar chart providing an easy comparison between the original production plan and a planned scenario, as displayed in Figure 4.7.

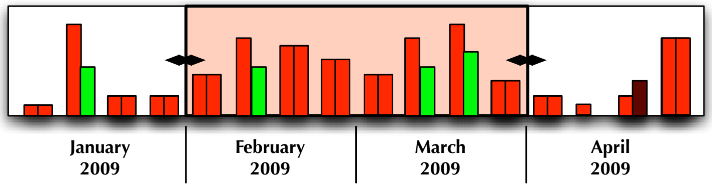


Figure 4.7: Date Selector GUI Prototype - Plan Comparison

GUIs Interaction

This solution integrates both GUIs: the selector will handle the defined time period in the *Production Plan GUI*, and as for the *Capacity Utilization GUI* it will provide a tool for selecting different time periods, in this case weeks. The user may simply use the mouse to click on a vertical bar, which will highlight and select that week.

4.5.3 Solution Justification

The most common solution is to present two *text-boxs*, representing the desired start and end date, providing the key functionality of defining a time period while also being practical and intuitive. Although this concept works, it fails on two key integration concepts: aid production plan viewing and integrable into previously defined controls. This solution is unable to select multiple time periods, and cannot be integrated with the *Capacity Utilization GUI*.

The production planner requirements state that: the key for a successful *Production Plan* is on-time delivery. Following this premise and the ones defined in the *Concept Analysis* (visualization rules and manufacturing environment concepts), a solution was provided that integrates all requirements.

From the research in the *State of The Art* chapter the *Google Finance* date selector (Sub-section 3.1.2.1) is seen as a breakthrough in time frame selections. Following this notion a similar concept was adopted. A *Line Chart* solution would prove complicated to know how many late orders there are in a specific time period, due to the granularity of being a day, or an average of a period. The key is to switch to a *Bar Chart* with a defined granularity of weeks, allowing the planner to quickly look outside the selected time scope, at how many orders are late elsewhere. It also allows for plan comparison by inserting two series into the bar chart, as seen in Figure 4.7. If the color represents lateness, color can change to reflect variations in this aspect, to *green* if it improves, and to *dark red* if it worsens, providing the planner with a quick overview of how many production orders have improved its lateness and where, as seen in Figure 4.7.

One of the key features that was missing was the integration of the *Production Plan GUI* and the *Capacity Utilization GUI*, for which this control proves to be essential, implementing use case IV-UC12. The planner may start with the first interface, select the weeks that have the most problems and switch to the latter.

4.6 Summary

This chapter has defined a specification based on the gathered requirements in the *Problem Description* chapter. A clearer view of how the software should be designed and implemented was achieved.

The specification provided a solution for the problem at hand based on research from the *State of The Art* chapter. The methodology followed was to introduce each problem, analyzing specific business concepts in the SCM and Manufacturing environments, providing solutions and their justification. These steps insure that the provided solution was studied and compared with other alternatives, awarding more integrity and value to the software.

The researched solutions involved describe how data should be presented to the user (referred to as *planner*), in order to provide a production plan overview, aid in bottleneck detection, problem solving and planning optimization, as stated in this project's requirements (Section 2.2). The solutions represent the integration of innovative visual concepts with advanced chart views, attempting to ease the planning process.

The following chapter, *Design and Implementation*, will rely on this specification for building the software solution, as well as verify if the goals were met, if the tool is helpful.

Chapter 5

Design And Implementation

"The greatest value of a picture is when it forces us to notice what we never expected to see."

John Turkey

This section consists of two parts of the development process, the *Design* and the *Implementation* stages. The first will present the architecture by using diagrams and detailing components' interactions, providing a basis for how the latter stage will proceed. The level of detail is adjusted to that of reading an article, providing high-level overviews, focusing on the big picture solution and drilling-down into important concepts, not entering low level details.

The design stage will focus on architecting user interfaces, providing different project perspectives, outlining expected behaviors and interactions. The implementation stage will take these concepts and show what was the actual development, describing how some of the problems were overcome.

Throughout this chapter deviations from the original specification will be pointed out, providing its reasons and justifying the chosen approach.

5.1 Design

After the project requirements were gathered (Chapter 2), visual components were defined (Subsection 2.2.1). The following step is to specify the system architecture and design. This will provide insight and set a path for the implementation stage, by defining components, their actions, flows and expected behaviors. Detailing how the system works will result in architectural effectiveness and an application that is closer to what the user demands.

The 4+1 model proposed by Philippe Kruchten [5] was the used approach, defining a clearer design and architecture, as seen is Figure 5.1. This model describes architectures for software-intensive systems, based on the use of multiple, concurrent views. By defining different perspectives the architecture's stakeholders concerns are specifically addressed, also handling separately functional and non-functional requirements [46].

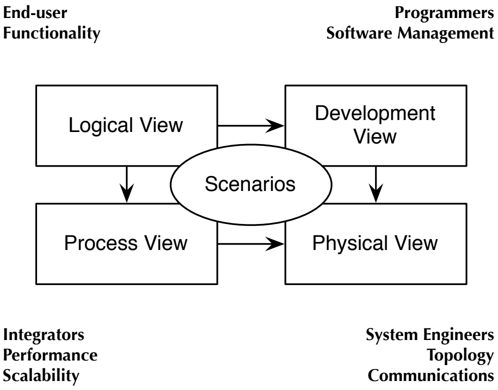


Figure 5.1: The ”4+1” view model [5]

The *Scenarios* view, was already defined in the problem description by the use of use cases (Subsection 2.2.1); the remaining views will be described in the following subsections.

5.1.1 Architecture Overview

From the visual components previously defined (Subsection 2.2.1), an architecture was designed and a high level view is displayed in Figure 5.2. Two layers were delineated, *Interactive Visualization* and *Business Logic*. Their goals are clearly distinct: the first handles the visual components available to the user gathering data from the business layer, while the latter is the business logic behind the GUI, communicating with a database for retrieving the selected production plan, and building an object model.

5.1.2 Logical View

This view's goal is to support the functional requirements, defining what the system should provide in terms of services to its users. The logical view is decomposed into a set of key abstractions, derived from the problem domain in the form of objects or object classes, exploring the principles of abstraction, encapsulation and inheritance. Not

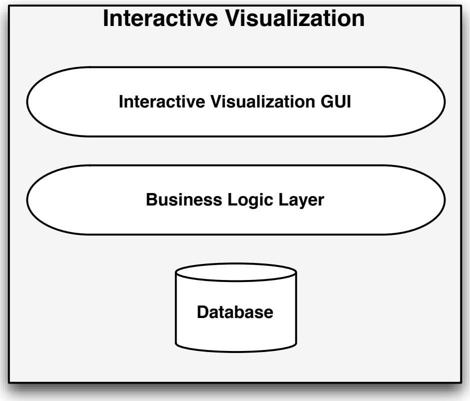


Figure 5.2: Interactive Visualization - High Level Components Architecture

only does it provide functional analysis, but also identifies common mechanisms and design elements across the various parts of the system [5]. A study on which components will exist is needed for developing a stable and overall efficient system.

The high level architecture was therefore split into smaller components, as depicted in Figure 5.3, exposing the *Interactive Visualization GUI*, and the *Business Logic Layer* components' relations.

The architecture in Figure 5.2 outlines the project's components, detailing the previously defined layers. Note that the number of classes in the system is higher than what is detailed in this chapter.

5.1.2.1 Interactive Visualization Layer

The following layer explanation will detail what are the main classes' goals, from the ones in Figure 5.2, which instantiate other classes and explaining the links between them. The interaction between the main classes will be detailed in the *Process View* (Subsection 5.1.3).

Interactive Visualization GUI

This components holds visual controls and other classes for managing the system. It is a graphical interface where all controls will be laid out, providing a basis for user interaction, communicating with the *Business Logic* lower level layer, retrieving the production plan for building its visual components, and following the solution analysis and justification previously presented (Subsection 4.2).

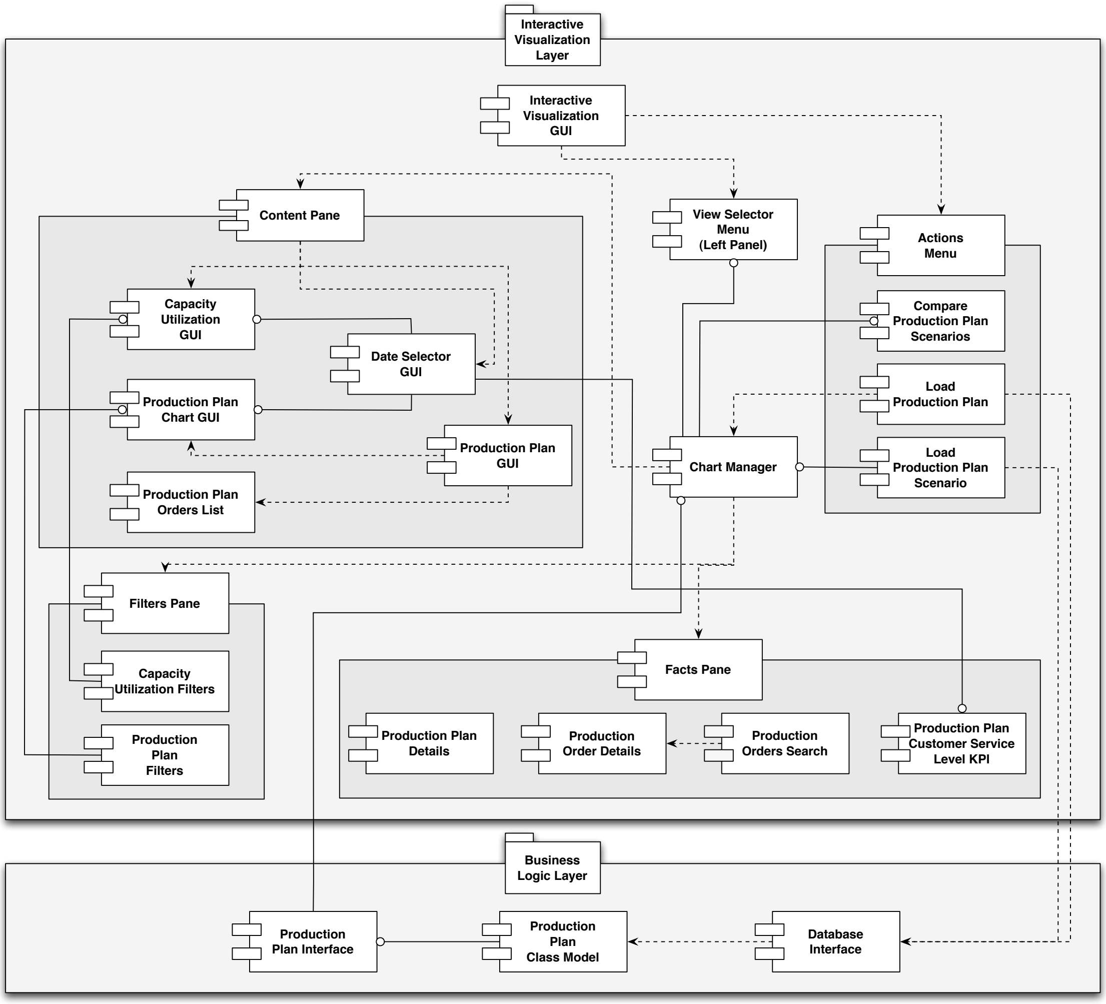

Interactive Visualization Layer
Interactive Visualization GUI
Content Pane
View Selector Menu (Left Panel)
Actions Menu
Capacity Utilization GUI
Date Selector GUI
Compare Production Plan Scenarios
Load Production Plan
Production Plan Chart GUI
Production Plan GUI
Chart Manager
Load Production Plan Scenario
Production Plan Orders List
Filters Pane
Capacity Utilization Filters
Production Plan Filters
Facts Pane
Production Plan Details
Production Order Details
Production Orders Search
Production Plan Customer Service Level KPI
Business Logic Layer
Production Plan Interface
Production Plan Class Model
Database Interface

Figure 5.3: Interactive Visualization - Class View

Chart Manager

When developing any system, some control must be responsible for maintaining data consistency and integrity throughout the application. This is even more of a certainty when there is visual logic and business logic. Separating concepts is important, by creating lightweight viewers that rely on a centralized system for fetching data and updating their visual states.

The *Chart Manager* is the main synchronizing class, providing means for interfacing the *Business Logic Layer* with the visual controls. Following this premise, all controls will report to it for: filtering, switching views, loading data, and so forth, centralizing user interaction. Whenever a control is activated the chart manager will process its inputs providing the outputs, cascading from a logic level up to the visual level. For example, after the production plan load, the user decides to turn on the market type color filter; the chart manager will be notified of this action, change its defined filter values, and activate the *InvalidateVisual()* method on the main graphical user interface. This cascading architecture is further explained in the *Process View* (Subsection 5.1.3).

When the user activates the *Load Production Plan Control* this class will be instantiated and start retrieving the loaded production plan from the *Production Plan Interface* in the *Business Logic Layer*. Following the load, it will instantiate several visual components and provide a basis for data retrieval.

By centralizing the main logic, all controls will have access to a common control, assuring that the displayed data is consistent throughout the entire system.

Menus and Panes

In a graphical user interface some controls show the data, while others alter its perspective. The user interaction for filtering, drilling down to other details, and changing views is achieved by using complementary menus and panes.

The *Actions Menu* is a static and contextual visual control (Subsection 4.2.1) where the user interaction will start. The *Load Production Plan* and *Load Production Plan Scenario* are other visual controls held and instantiated by the actions pane, providing means for loading data, by interacting with the *Database Interface* provided in the *Business Logic Layer*, and accessing the database. The previous controls are always available to the user, while the final control, *Compare Production Plans Scenarios*, is of contextual nature and exists only in the *Production Plan GUI*; it allows the user to switch between the original plan and its scenario.

The *Facts Pane* is a visual control and it instantiates and holds similar controls within its logic, providing a basis for the layout of information that complements the produced view. The *Production Orders Search*, given a specific *Production Order ID*, will interact with the *Chart Manager* for data retrieval and display it using the *Production Order Details Class*. The final component available in this view is the *Production Plan Customer Service Level KPI*, which provides a customer service level within a specific time frame.

The *Filters Pane* is a contextual visual control, that holds all available filters. It interacts with the *Chart Manager* for updating the user interface layout.

The *View Selector Menu* is a static visual control that provides access for switching between the *Production Plan GUI* and the *Capacity Utilization GUI*.

Content Pane and GUI Styling

The *Content Pane* is a visual control that will instantiate and hold the main developed GUIs, the *Capacity Utilization GUI*, the *Production Chart GUI*, and the *Date Selector GUI*. The first two are laid out within this control and interact with the *Chart Manager* for updating its layout, based on the *Date Selector GUI* and their filters.

All available graphical user interfaces instantiate their own *Style* class, following the logic in Figure 5.4, allowing an easy way for altering how the layout is managed, e.g. the *Production Plan Chart GUI* will provide methods for changing colors such as bubble or filters, and also changing size time and lateness axis thickness, and so forth.

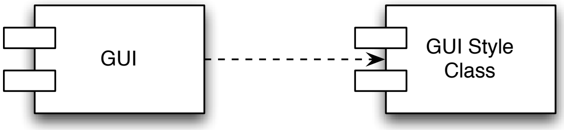


Figure 5.4: Graphical User Interface Example of Style Class Instantiation

Production Plan GUI

As described by the diagram in Figure 5.2, this graphical user interface relies on the instantiation of two controls, the *Production Plan Orders List* and the *Production Plan Chart GUI*. The first provides a listing of the available production orders in the production plan. Having both available in the same interface allows the user to directly relate the list view with its visual representation.

The *Production Plan Chart GUI*, as displayed in Figure 5.5, is a more complex user interface, that holds many different components for displaying its content. While the GUI itself is responsible for contacting the *Chart Manager* for retrieving data and the *Date Scheduler* for retrieving the specified time frame, the *Bubble Chart* class is a base for displaying the bubbles. It provides several useful methods, for positioning a bubble according to its delivery date and lateness, and applies the filtering logic by removing and coloring bubbles.

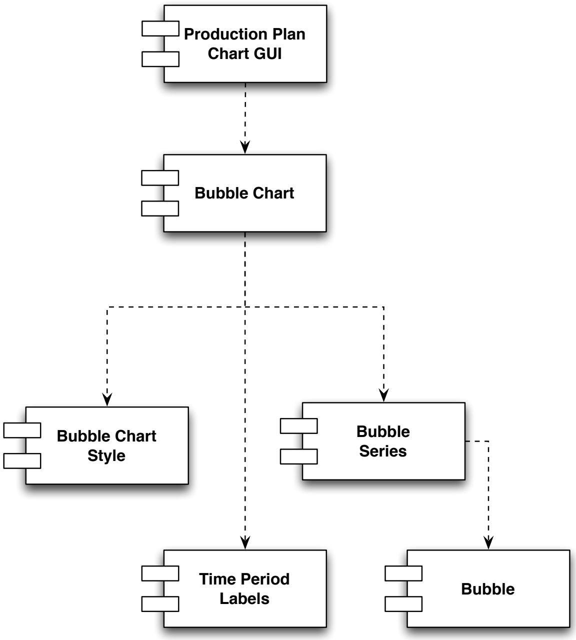


Figure 5.5: Production Plan Chart GUI - Class Diagram

The *Bubble Series* class, represents a group of *Bubble* objects. This provides a simple way of filtering out production orders, by using different series, an important factor for performance purposes. This allows fast time period transversing and filtering, without reloading unnecessary production orders.

The *Time Period Labels* class is the one responsible for analyzing the currently selected time period and dynamically adjusting the chart labeling system, according to the specific time period. It provides key references to the user, such as the time period is currently shown and what is the actual date a production order (represented by a bubble) will be delivered on.

Capacity Utilization GUI

The *Capacity Utilization GUI* is responsible for building a *3D Bar Chart* view capacity loads. It gathers its three dimensions from the selected time periods, the selected filters, and capacity level.

The main class is responsible for contacting the *Chart Manager* for retrieving data and the *Date Scheduler* for retrieving the specified time frame. As shown in Figure 5.6, a *3D Chart* class instantiated for displaying the chart itself from the gathered data, having each 3D bar represented by the *3D Chart Week Aggregation*. Besides representing a bar, this

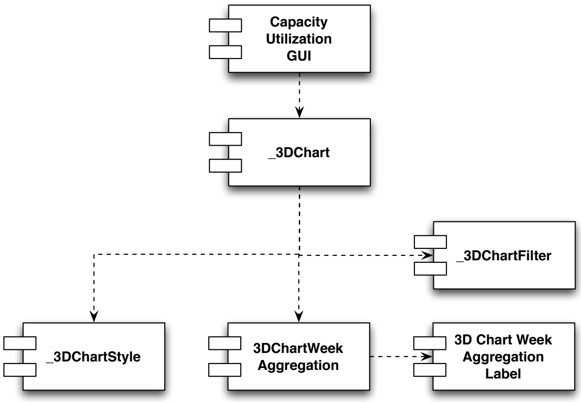


Figure 5.6: Capacity GUI - Class Diagram

class is key for fulfilling the goal of comparing finite and infinite capacity levels, because it's able to display a bar with two colors, *green* and *red*. The first color represents up to 100% capacity reached, while the latter represents everything above this level.

The *3D Chart Week Aggregation* also instantiates the *3D Chart Week Aggregation Label* which will be positioned on the top of the bar, showing to the user what is the actual value it represents.

The *3D Chart* class will be responsible for the user interaction within the 3D world, providing a way for the user to rotate the view, and obtain different perspectives.

Date Selector GUI

This control is available within the *Content Pane* and allows the user to specify a time frame for viewing production orders, relaying these changes to the *Production Plan GUI* and the *Capacity Utilization GUI*.

The *Date Selector*'s class diagram is displayed in Figure 5.7. For representing each 2D bar the *ChartWeek* visual control is used, which grows according to the specified number of late orders. The main control also holds a *Time Period Labels* object, which adjusts to the production plan full time frame, displayed in months.

5.1.2.2 Business Logic Layer

The *Business Logic Layer* is responsible for interfacing between the graphical interface and the database, as well as to provide an object model of the production plan schedule. Accessing the production plans database is achieved through the *Database Interface* class, which after loading instantiates the *Production Plan Class Model*, that builds the business logic, later accessed by the *Chart Manager* through the *Production Plan Interface*.

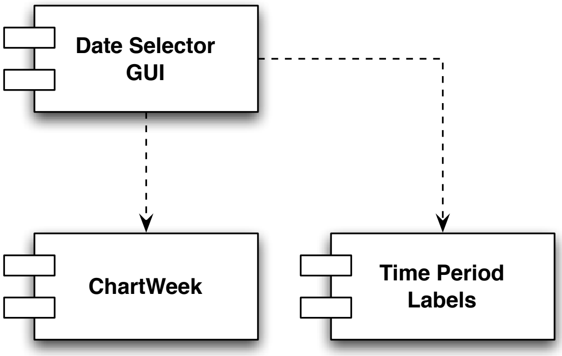


Figure 5.7: Date Selector - Class Diagram

The database logic behind this follows a simplified version of the concepts introduced in Subsection 2.1.2, shown in Figure 5.8.

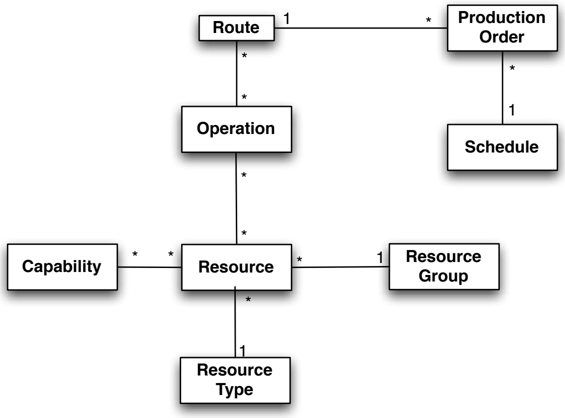


Figure 5.8: Business Logic - Simplified Class Diagram

5.1.3 Process View

The process view takes into account some non-functional requirements, such as performance and availability, addressing different issues, such as: concurrency and distribution, system's integrity, fault-tolerance, outlining how the main abstractions from the logical view fit within the process architecture. There are several levels of abstraction for addressing different concerns. The process view describes interactions between the systems' components and communication flow [5].

For architecting a user interface and aiding the project's design and implementation, it is important to define the components behaviors, communications and user interactions.

The *Interactive Project*'s architecture has one main task that is essential for displaying data: the production plan database loading.

When the user loads a production plan, there is a sequence of events that takes place: the *Interactive Visualization GUI* and the *Business Logic* layer interact to load data from the database, and build the visual components. In Figure 5.9 the communication of all entities is portrayed by using an Unified Modeling Language (UML) sequence diagram [70].

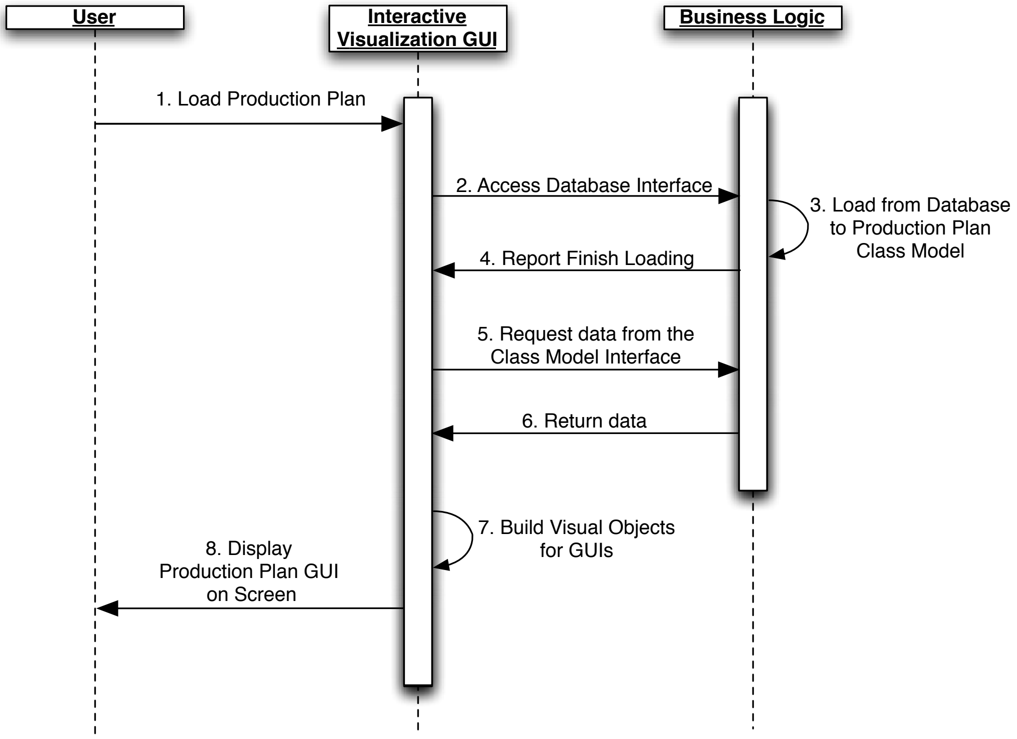

User
Interactive Visualization GUI
Business Logic
1. Load Production Plan
2. Access Database Interface
3. Load from Database to Production Plan Class Model
4. Report Finish Loading
5. Request data from the Class Model Interface
6. Return data
7. Build Visual Objects for GUIs
8. Display Production Plan GUI on Screen

Figure 5.9: Production Plan Load Sequence Diagram

There is also the need for specifying user interactions and behaviors with the interface; a good tool for achieving such goal is using a state diagram. Following the defined requirements (Subsection 2.2), two different action types are identified: changing content or perspective. The first will have to communicate with the database following the previously described procedure and rebuild the visual components, while the latter only causes a component update, changing the user's perspective.

A state diagram based on user interaction is shown in Figure 5.10; the logic is that when an action changes the displayed content, a new state is created, when the user performs an action that does not change the content but only the perspective, therefore the state remains the same.

The main class for synchronizing data and organizing the visual components is the *Chart Manager*. It provides a centralization environment from which user interactions from visual controls will serve as inputs, and directed to their respective output controls. This cascading effect guarantees that visual integrity and consistency is achieved. For

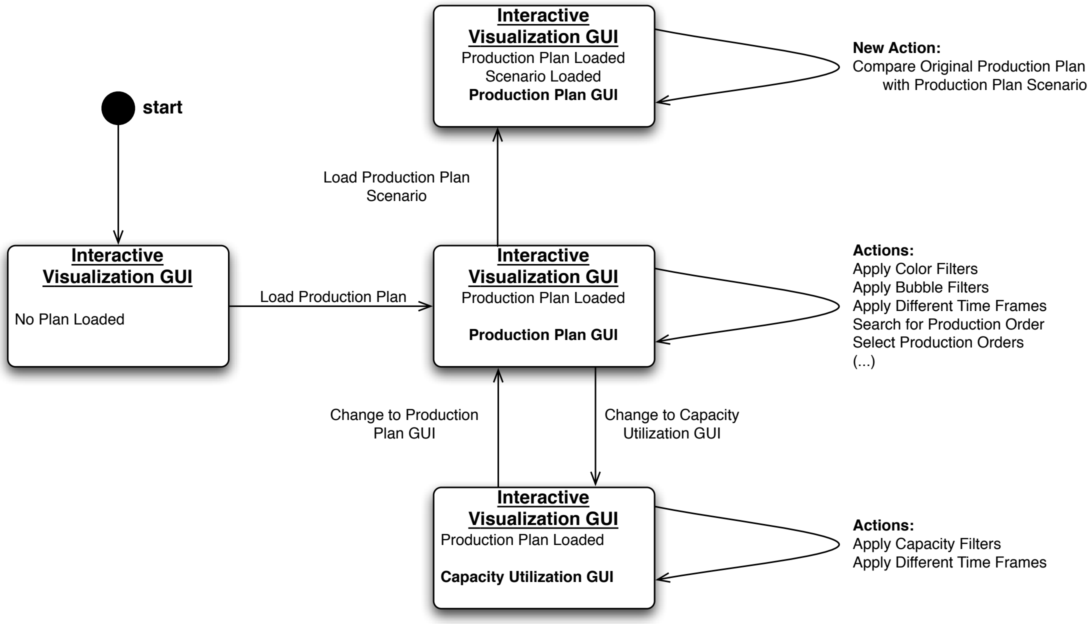

start
Interactive Visualization GUI
No Plan Loaded
Load Production Plan
Interactive Visualization GUI
Production Plan Loaded
Production Plan GUI
Actions:
Apply Color Filters
Apply Bubble Filters
Apply Different Time Frames
Search for Production Order
Select Production Orders
(...)
Load Production Plan Scenario
Interactive Visualization GUI
Production Plan Loaded
Scenario Loaded
Production Plan GUI
New Action:
Compare Original Production Plan with Production Plan Scenario
Change to Production Plan GUI
Change to Capacity Utilization GUI
Interactive Visualization GUI
Production Plan Loaded
Capacity Utilization GUI
Actions:
Apply Capacity Filters
Apply Different Time Frames

Figure 5.10: Interactive Visualization GUI - User Interaction

demonstrating this behavior, an example is displayed in Figure 5.11, describing what happens when the user changes the selected time frame.

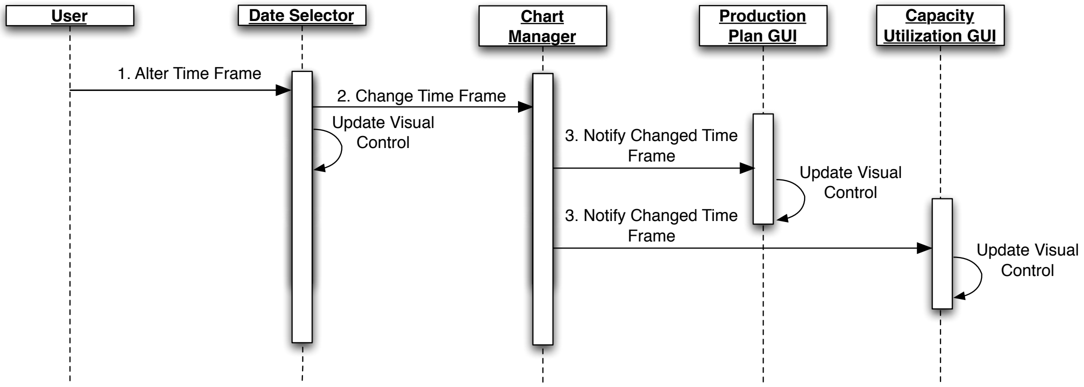


Figure 5.11: Chart Manager Component Notification Example

We are able to see a set of interactions taking place in order to maintain visual integrity. The *Chart Manager* is responsible for emitting visual update calls on the required components, or a system-wide call to the main window.

5.1.4 Development View

The development view presents a high level architecture, focusing on the system module organization, outlining the software's main sections, and displaying them in an hierarchical view [5]. Figure 5.12 outlines the main components. Each is able to work without the others, handling the task, and fulfilling their specified requirements. Although they are separated, for the project to work as whole they must interact as described in the previous subsections. This is meant as a high level view of the designed software; it does not interact with the aid of other libraries, all components are designed to be specific for performing their task.

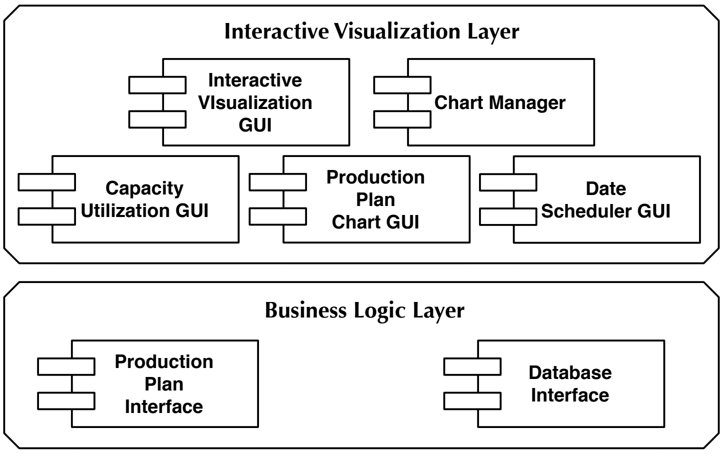

Interactive Visualization Layer
Interactive VIsualization GUI
Chart Manager
Capacity Utilization GUI
Production Plan Chart GUI
Date Scheduler GUI
Business Logic Layer
Production Plan Interface
Database Interface

Figure 5.12: Development View - Main Components

5.1.5 Physical View

The physical view considers, primarily the non-functional requirements of the system such as availability, reliability, performance and scalability. It presents the software communication through a network of computers, the topology and layers with whom the software interacts and its different levels [5]. The produced software does not change the Dynamics AX physical architecture, or add any new feature or value, therefore it was disregarded.

5.1.6 Visual Performance Considerations

The used .NET API, WPF, provides a set of visual animations capable of producing visually appealing controls. Although this represents a break-through in software development, because the programmer does not need to consider animation timing, the performance issue arises.

For a faster performance, when a production plan is loaded the *Production Plan GUI* and *Capacity Utilization GUI* will access the loaded data through the *Production Plan Interface* in the *Business Logic* package, build their visual components, and save them in memory, providing a faster data access which is crucial for a jitter free application.

This means that the only access to the database will be during the loading periods, all components will be instantiated at once, providing an overall better user experience.

5.1.7 Technologies

This section provides information regarding the selected technologies for this project's development, describing what is the used framework and programming language, as well as other tools that aided in the development process.

5.1.7.1 .NET Framework and WPF

Microsoft Dynamics AX is built using the .NET Framework. For integration purposes, one of its technologies must be chosen for desktop development: *Windows Forms* or *Windows Presentation Foundation* (Subsection 3.4.1). WPF was chosen for two main reasons: its the most recent Microsoft API and represents the latest in cutting edge GUI development, with its rendering engine built on top of DirectX (Section 3.4.1.3), providing a powerful framework for developing an interactive visualization desktop system.

5.1.7.2 Visual Studio .NET 2008

VS.NET 2008 is the main development platform for the .NET Framework, and will be the IDE for this project. It integrates all of the .NET programming languages and provides the usual IDE features along with some specific .NET solutions: code debugging, code highlighting, code debugging, LINQ, Expression Blend Support, Multi-Targeting Support, Nested Master Page Support, and so forth [71].

5.1.7.3 WPF Prototyping Framework 1.0 for Microsoft Dynamics

This toolkit, developed by the Microsoft Dynamics User Experience team, provides Dynamics AX WinForms controls for Windows Presentation Foundation. This allows to develop a system that interacts with the user similarly to the product release version, maintaining its look and feel, within WPF.

5.1.7.4 Microsoft Expression Blend 2

One of WPF's main design feature is the fact it separates the business logic from the visual logic, allowing both concepts to be developed in separate (Section 3.4.1.3). The visual logic is defined in XAML (Extensible Application Markup Language), an XML (Extensible Markup Language) specification for WPF, and the business logic uses any of the .NET programming languages. Microsoft Expression Blend 2 is a software that provides an easy way of developing Graphical User Interfaces. Every feature is accessible by drag-and-drop, providing a simpler way of developing a visually appealing GUI (Subsection 3.4.2).

5.2 Implementation

This section details how from the specification and design we achieve the implementation itself, describing the main classes, their goals, features and main visual components they derive from, outlining some of the development issues and describing the solution. A description of the non-visual classes that provide interfacing to the visual components will be defined, followed by the visual implementations.

5.2.1 Interactive Visualization Layer

The *Interactive Visualization Layer* is responsible for controlling all visual components and laying them out, using the *Interactive Visualization GUI*, separating the business logic from the visual logic. The main GUI represents a lightweight visual control, deriving from the *Page* class in the WPF API.

5.2.1.1 OnRender

In WPF there are two different ways of displaying visual controls within a visual container, adding *Children* to the container, and overriding the *OnRender* method. By simply adding children to the container the developer must rely on the components rendering system for updates. If the *OnRender* method is overridden then the development capabilities increase, children can be added as well as perform direct calls to the rendering system. This latter programming style was the one adopted in the *Production Plan Chart GUI*, *Capacity Utilization GUI* and the *Date Selector GUI*, providing a set of tools that enables the programmer to control the entire behavior of the visual component. The development difficulty is increased along with the power of the API.

5.2.1.2 Chart Manager

The *Chart Manager* is a non-visual class deriving from the main class *Object*. It acts as the ”brain” of the entire system, maintaining data integrity throughout all components and providing an abstraction layer through an interface for all visual controls to communicate with each other. This programming methodology allows every component to keep their own logic and abstract from the rest of the system, allowing them to concentrate on their main tasks, without other concerns.

Methods for interfacing are available for all components, changing the user's perspective; such examples are:

public void Add3DFilter(_3DChartFilter filter)
public void Remove3DFilter(_3DChartFilter filter)
These methods will be activated from the *Filters Pane*, passing this object down to the *Capacity Utilization GUI* which will update its visual.

public void SelectWeek(int weekNumber)
public void DeselectWeek(int weekNumber)
These methods are available for the *Date Selector GUI*, and will select or deselect different time periods, represented in weeks, to the *Capacity and Utilization GUI*.

public void ShiftStartEndWeeks(int _startWeek, int _endWeek)
This method provides a way for the *Date Selector GUI* to update the system time period, from switching the selected time frame, also resulting in a visual update to the *Production Plan Chart GUI*.

public void SelectProductionOrder(int ID)
After selecting a production order from the *Production Plan List*, this will be called, resulting in the selection of this orders' visual representation in the *Production Plan Chart GUI*.

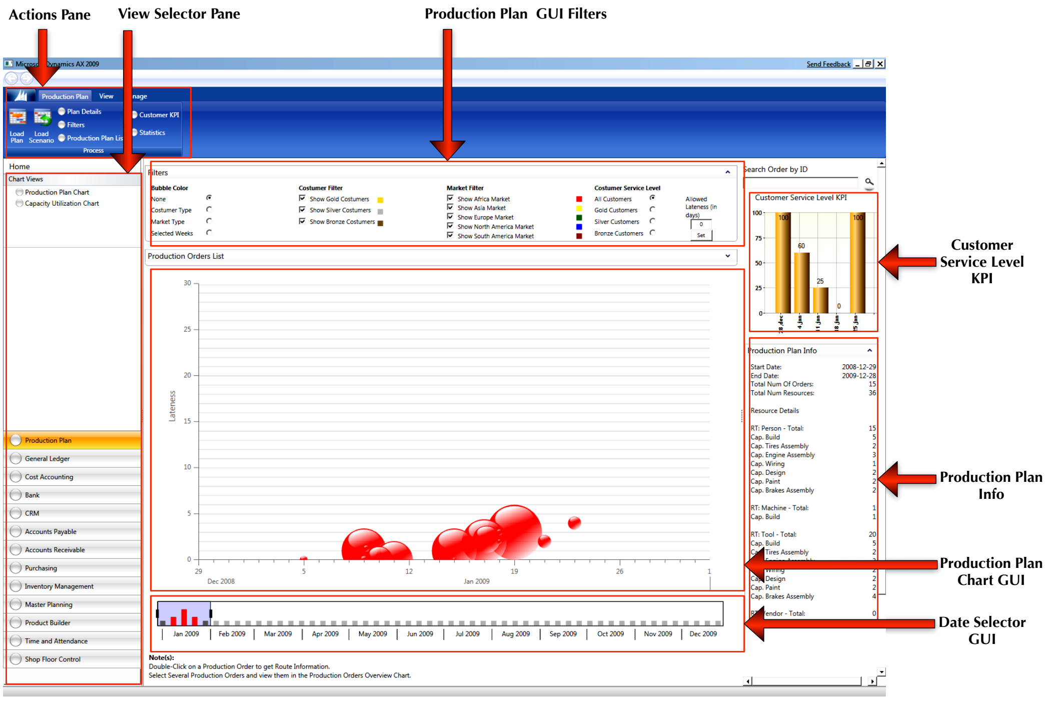
Such interfaces will make sure the system is up-to-date in terms of business logic, and the visual components are set accordingly to the selected filters and time periods.

5.2.1.3 Interactive Visualization GUI

This is the main graphical user interface where all other controls are layout. It follows the Dynamics AX UX guidelines (Section 2.2.2.2) and the final user interface is displayed in Figure 5.13.

5.2.1.4 Production Plan GUI

The *Production Plan GUI*, displayed in Figure 5.13, integrates two specific GUIs, the *Production Plan List GUI* and the *Production Plan Chart GUI*, also integrating the *Date Selector GUI* for time period selection. This user interface presents a number of visual development challenges; for the chart view, issues arise: representing and positioning the production order bubbles, providing a ”click on the bubble” feature, dynamically time labels positioning, filtering along other details and features.



Bubble Color: ● None ○ Costumer Type ○ Market Type ○ Selected Weeks

Costumer Filter: ☑ Show Gold Costumers ☑ Show Silver Costumers ☑ Show Bronze Costumers

Market Filter: ☑ Show Africa Market ☑ Show Asia Market ☑ Show Europe Market ☑ Show North America Market ☑ Show South America Market

Costumer Service Level: ● All Customers ○ Gold Customers ○ Silver Customers ○ Bronze Customers

Allowed Lateness (in days): 0 Set

Customer Service Level KPI	28 dec	4 jan	11 jan	18 jan	25 jan
	100	60	25	0	100

Production Plan Info

Start Date:	2008-12-29
End Date:	2009-12-28
Total Num Of Orders:	15
Total Num Resources:	36
Resource Details	
RT: Person - Total:	15
Cap. Build	5
Cap. Tires Assembly	2
Cap. Engine Assembly	3
Cap. Wiring	1
Cap. Design	2
Cap. Paint	2
Cap. Brakes Assembly	2
RT: Machine - Total:	1
Cap. Build	1
RT: Tool - Total:	20
Cap. Build	5
Cap. Tires Assembly	2
Cap. Engine Assembly	3
Cap. Wiring	2
Cap. Design	2
Cap. Paint	2
Cap. Brakes Assembly	4
RT: Vendor - Total:	0

Note(s):
Double-Click on a Production Order to get Route Information.
Select Several Production Orders and view them in the Production Orders Overview Chart.

Figure 5.13: Interactive Visualization Final GUI - Production Plan GUI

Bubble and Positioning

The production order representation is achieved by a bubble, which is derives from the *Canvas* class. The *OnRender* method is overridden and a complex gradient bubble is computed using several WPF APIs.

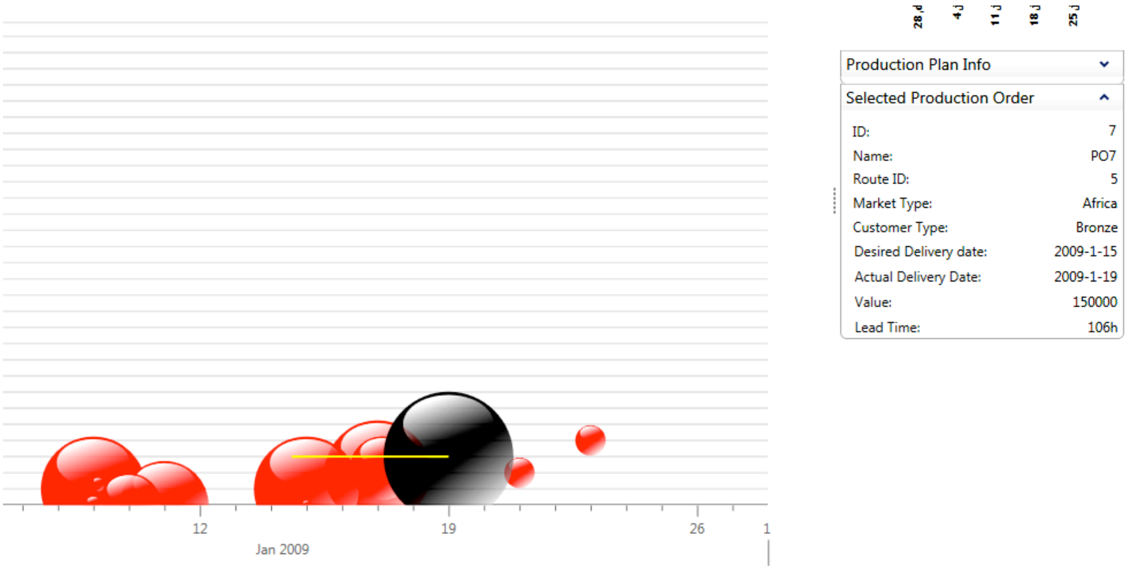
The chart bubble positioning must be simple, relying only on the necessary variables: the desired delivery date and actual delivery date; another important input is to know the selected time period, defined by a start and end date. With these four variables we are able to position the bubble accordingly.

The lateness, represented in the vertical axis (Subsection 4.3.2), is computed from the difference between the desired delivery date and the actual delivery date, while the horizontal axis position is calculated from the shift from the current start date to the actual delivery date.

The bubbles size itself also follows a set of rules, to avoid cluttering. According to the selected time frame's span the bubble's size will vary, increasing its value if the time period is small, and decreasing if the user defines a longer span.

Click On Bubble

For providing the user with an easy way of knowing what production order corresponds to which bubble, a needed feature was to allow the user to click on the visual component itself and get detailed information, as seen in Figure 5.14.



Selected Production Order

ID:	7
Name:	PO7
Route ID:	5
Market Type:	Africa
Customer Type:	Bronze
Desired Delivery date:	2009-1-15
Actual Delivery Date:	2009-1-19
Value:	150000
Lead Time:	106h

Figure 5.14: Production Plan Chart GUI - Bubble Click

There are a number of details available to the user after selecting a bubble: the *Production Plan List GUI* is highlighted with the correspondent production order; detailed information appears in a box on the *Facts Pane*; a line is drawn representing this order's lead time, allowing the user to immediately visually perceive how long this order takes.

This feature is achieved using WPF *HitTesting* API, which provides a way of associating a mouse event to a visual component, finding out which one was clicked. Because WPF allows components overlaying this it also provides a system for transversing through every control in the clicked area, allowing the developer to decide the necessary one.

Dynamic Labeling

A feature available in any chart is a dynamic labeling system, adjusting to the variables in the given axis. In the *Production Plan Chart GUI* there are two axis, lateness and delivery time axis. The first is static, based on user input(who provides a minimum and maximum lateness levels), while the latter is dynamic and automatically adjusts to fit the time reference.

There are three main types of labels; month, monday (representing a week change), and day. The time axis displays the final two types, according to the selected time period. If the necessary space is available, all days are displayed; if the time period is to broaden then only the mondays are displayed, as seen in Figure 5.15.

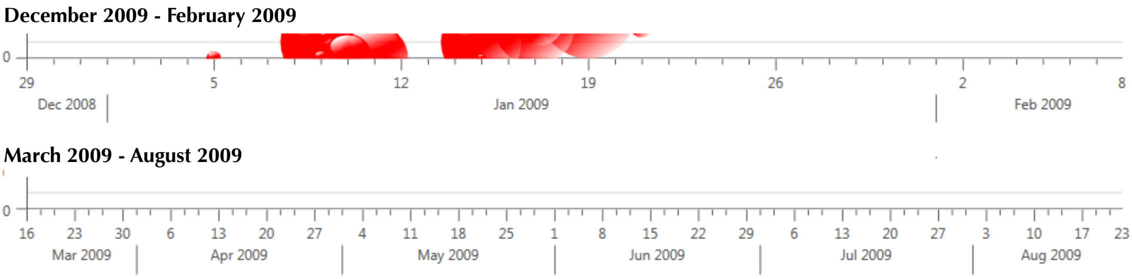


Figure 5.15: Production Plan Chart GUI - Dynamic Labeling

Labels Animations

For increasing the user experience when the user selects a different time frame, an animation occurs, allowing the user to see the chart, sliding to the given start and end dates, instead of updating to its final state. This is achieved using the WPF Animation API, which abstracts all timing calculation and the only required implementation was to define *Dependency Properties* in the *Bubble Chart* class. The *StartDateProperty* and *EndDateProperty* allow the *Animation* object to interact with the *Bubble Chart* by sending them different values and updating, e.g. the properties are set with 10-10-2009/10-10-2009, respectively, while the current selection is 01-01-2009/10-01-2009, the animation object provided a defined animation time, e.g. 2 seconds, will change to the desired values within this period computing the intermediate values and timings, updating the chart every step of the way.

Filters

For allowing the user to change the production plan's perspective by switching the bubble colors, a filtering system was put in place. After the user selects coloring option (Selected Weeks, Market Type or Customer Type), the *Production Plan Chart GUI* will receive an *InvalidateVisual()* call, which will force the bubbles to be redrawn. Their color is retrieved from the *Chart Manager*, based on their order content and active color filters, allowing for a simple and fast color change. Figure 5.16 shows the color difference between using customer type and market type coloring. There are other available filtering mechanisms, that differ from the previously presented. These filters are suppose to remove orders from the chart, increasing its visibility, e.g. if the user selects market type filtering and wants to see a specific market type, he/she may select this option, and the other bubbles will be hidden from view; this works like the color filters.

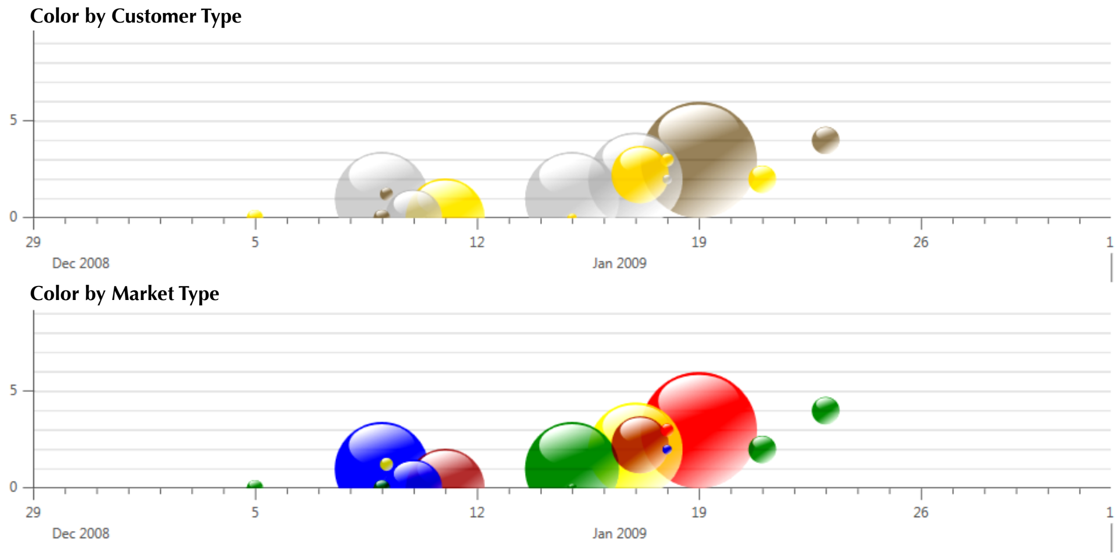


Figure 5.16: Production Plan Chart GUI - Coloring Comparison

Plan Comparison

For comparing a plan the specification demanded that bubbles change color to *green* when improved, or *dark red* when they get worst. In Figure 5.17 we are able to see this feature's visual effect. This feature is activated through the actions pane, and it compares the original plan with its scenarios displaying the changes.

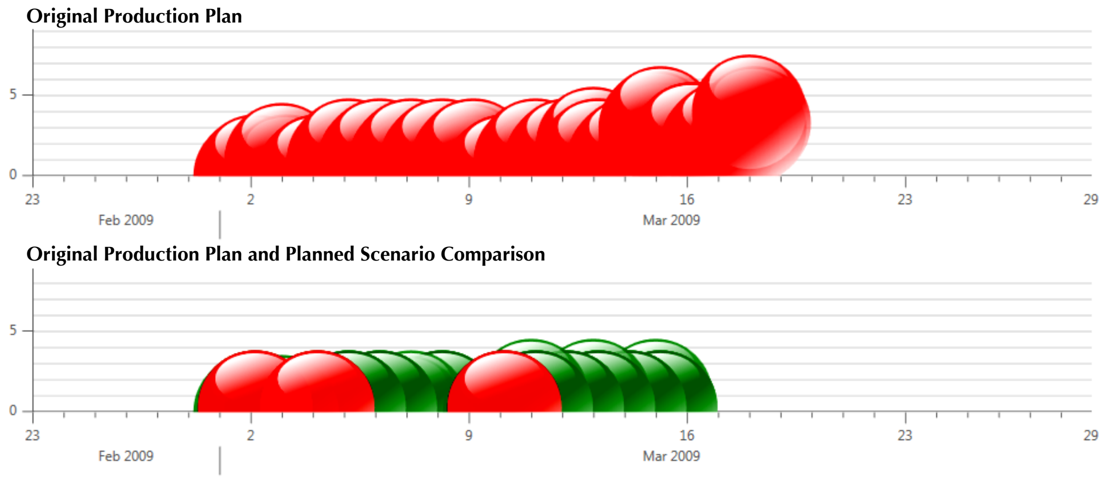
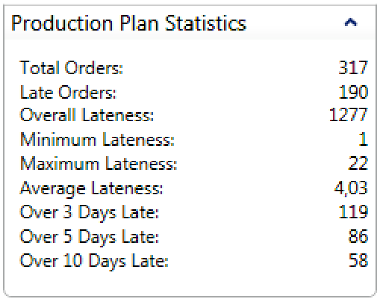


Figure 5.17: Capacity Utilization GUI - Plan Comparison

We are also able to use the *Customer Service Level KPI* feature shown in Figure 5.13, which shows the ratio between the number of delivered and requested orders, in a 2D bar chart. The user may also change the KPI's perspective by choosing a different customer type or setting a higher allowed lateness, which is represented in days and defaults to zero. This feature also shows whichever plan is selected, the main production plan or any of its scenarios.

Another feature for comparing plans is the *Production Plan Statistics*, also available in the *Facts Pane*, allowing the user to know: Total Orders, Late Orders, Minimum Lateness, Maximum Lateness, Total Lateness, Average Lateness, Orders Over 3 Days Late, Orders Over 5 Days Late and Orders Over 10 Days Late, as displayed in Figure 5.18. With this data, the user has another tool for comparing the differences in from the original plan to the planning scenario, providing more information for determining what is the best solution.



Production Plan Statistics

Total Orders:	317
Late Orders:	190
Overall Lateness:	1277
Minimum Lateness:	1
Maximum Lateness:	22
Average Lateness:	4,03
Over 3 Days Late:	119
Over 5 Days Late:	86
Over 10 Days Late:	58

Figure 5.18: Production Plan GUI - Statistics

5.2.1.5 Capacity and Utilization GUI

Following the specification and defined designed the view in Figure 5.19 was achieved. For an enhanced interaction, some features and specific details were developed, facilitating the final perspectives, and making sure the view is as clear as possible from every angle, this 3D view uses the *Viewport3D* class.

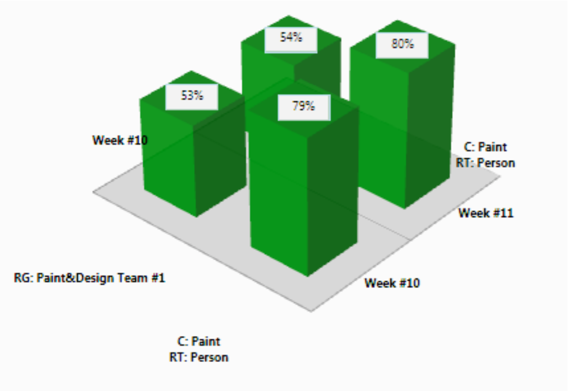


Figure 5.19: Capacity Utilization GUI - Finite and Infinite Comparison

Selecting Different Time Frames

The *Date Selector GUI* provides a key functionality for selecting different weeks and obtaining their capacity levels, as displayed in Figure 5.20, the user has selected one week, represented in *black*, and shown on the chart.

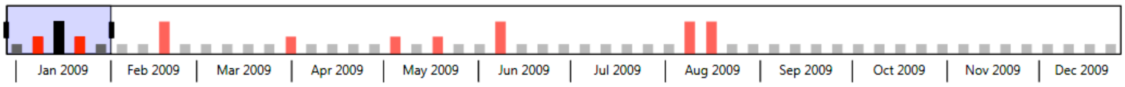


Figure 5.20: Date Selector GUI - Select Capacity Time Frames

Finite and Infinite Comparison

The capacity loaded in each filter by the specified time period is displayed in the standard view, another feature is to show what the capacity levels would be if there were infinite resources.

Figure 5.21 shows the required capacity level in a percentage level of what is currently available, e.g. if resource type *Person* with capability *Build* is at 90% in the finite view and 150% in the infinite view, then the gathered information is that for producing the orders in the selected time period more than 50% of the available resources (of type *Person* with capability *Build*) are required.

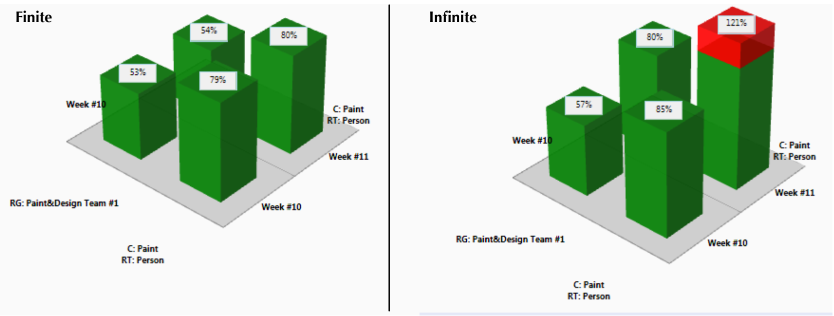


Figure 5.21: Capacity Utilization GUI - Finite and Infinite Comparison

3D Interaction

In WPF a 3D model has an associated *Camera*, defining the user's viewpoint. To provide a complete three dimensional user interaction, camera movement was added, allowing the user to change the camera's position and obtain different perspectives. To achieve this feature, the system in Figure 5.22 was created.

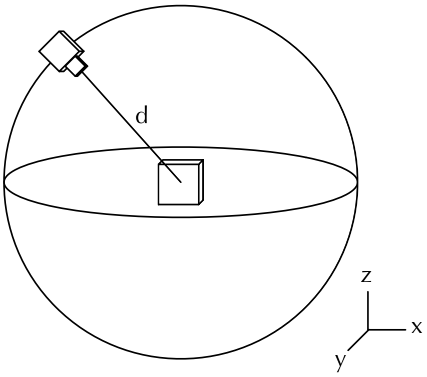


Figure 5.22: Capacity Utilization GUI - 3D Spherical Camera Movement

The 3D world is viewed as a sphere, and the camera rotates around its center. Distance *d* is calculated to automatically fit the contents in the view, so the user does not need to worry about zooming in/out. This provides a way of changing perspective to align the camera with what the user wants to see, increasing the number of possible views.

2D Labeling in 3D World

In order to know what is the capacity level of a 3D object it is necessary to place text in the chart. The solution was to introduce 2D labels into the 3D view, and this feature required the development of two different calculations: find out the coordinates of the 3D object in 2D, and hide the labels if another 3D object is in front.

Placing the labels required calculations using the viewport's 2D width and height, along with the defined camera field of view and the 3D object's coordinates. These calculations output an x and y coordinates for positioning the label. For hiding the labels when another object is in its path the implemented procedure was to use the *HitTesting* API from WPF. In Figure 5.23 there are two objects in front of each other the hit test will return the object that is in front first and determine if from the current view the object in the back is visible. If the object is visible a label is placed on top of the 3D bar.

5.2.1.6 Date Selector GUI

This graphical user interface provides a way to dynamically select a time period for viewing. The devised date selector follows the specification (Section 4.5), and is displayed in Figure 5.24. This tool provides a dynamic labeling system (from a provided start and end date) for the plan, and allows to compare a production plan with its scenario.

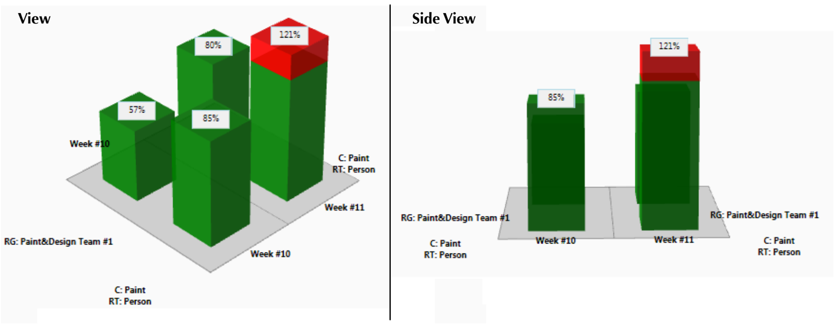


Figure 5.23: Capacity Utilization GUI - 3D Labels Hiding

Lateness Bars

The 2D bars displayed in Figure 5.24 show the lateness in the current week, where the height of the bar represents the number of late orders. For selecting different time periods in the *Capacity Utilization GUI* the user clicks the desired weeks and the GUI is updated.

Dynamic Time Selection

The user is able to choose any time frame within the specified production plan time period. There are two ways to achieve this; the user may drag the time frame's rectangle to another period, maintaining the time frame span, or drag the time frame start and end increasing the span. Figure 5.24 two distinct time frame selections are displayed.

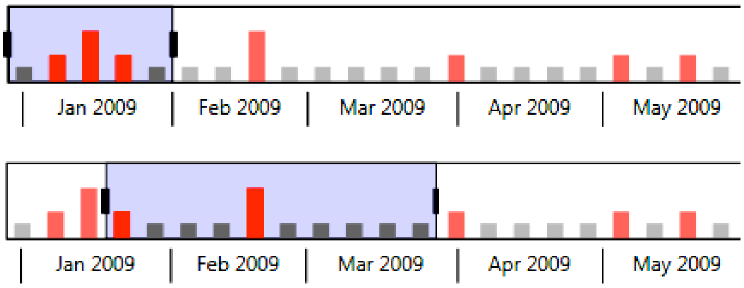


Figure 5.24: Date Selector GUI - Dynamic Time Frame

Labels

The labeling system works similarly to the one developed for the *Bubble Chart* (Subsection 5.2.1.4), adjusting to the start and end date of the production plan, as seen in Figure 5.24.

Plan Comparison

If the user selects a plan scenario for comparison with the original plan, another series of bars is introduced in the user interface. In Figure 5.25, it is possible to see how it works; bars allow the user to quickly compare both plans, by observing their height difference.

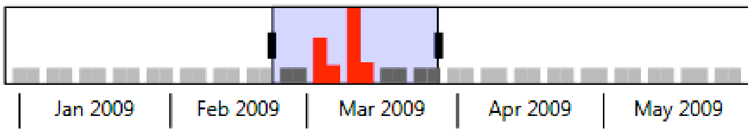


Figure 5.25: Date Selector GUI - Plan Comparison

5.2.2 Business Logic Layer

The visual aspect of the application was separated from the business logic, a lower level layer, by implementing a database model based on the specification (Subsection 5.1.2.2) using Microsoft SQL Server 2008, a Database Management System (DBMS) which provides all necessary performance requirements. This layer also provides interfacing between the database itself and the visual layer, through two interfaces. For loading production plans the top layer simply calls the *Load(int ID)* method with the correct plan identification number. For interfacing with the production plan class model the top layer has access to its interfaces, to know how to query the system.

5.2.3 Other Tools

The proposed architecture relies on using a custom database and not integrating directly with Dynamics AX. This solution will need customized tools for interacting with the database and setting up a production plan, and its scenarios. Although the actual database model will only be defined in the *Design and Implementation* chapter, the required tools must be outlined from the start.

5.2.3.1 Route Visualizer

This application will provide an interface with the database, allowing the user to introduce new routes, and schedule production orders. This will be helpful for visually interacting with the data, instead of a direct input approach. The complexity of a database that supports the requirements for setting up a production plan proved to be a challenge.

5.2.3.2 Production Orders Generator

This tool will simulate a working environment, generating production orders for later scheduling. It will rely on the database model for scheduling and introducing orders, and provide an accessible way of generating a long production plan (e.g. one year planning). For scheduling, it relies on the *Job Scheduling Engine* developed by the Microsoft Dynamics AX Manufacturing team. Its architecture and communication is depicted in Figure 5.26.

This software will also generate production scenarios, by executing a pre-defined production plan, introducing new resources, and running it in the scheduling engine.

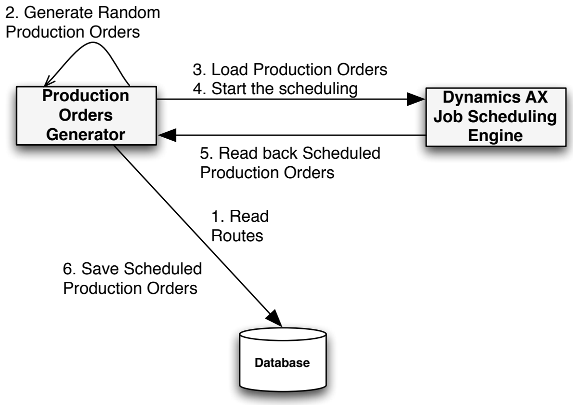

2. Generate Random Production Orders
3. Load Production Orders
4. Start the scheduling
Production Orders Generator
Dynamics AX Job Scheduling Engine
5. Read back Scheduled Production Orders
1. Read Routes
6. Save Scheduled Production Orders
Database

Figure 5.26: Production Orders Generator Architecture and Communication

5.3 Summary

This chapter presented the proceedings for design and implementations, regarding the developed graphical user interfaces. It describes details as to how they are integrated into one main GUI, and also how they are separated into two layers (a visual and a business layer) while introducing the technologies used for its development.

This chapter relates the specified requirements from the *Problem Description* chapter (Subsection 2.2), with the chosen software architecture and design. The implementation also presents how the software was developed, describing why and how several specific solutions were introduced for achieving the desired requirements.

This software solution allows the user to load a production plan, along with planning scenarios, analyzing the production orders' lateness and relating them with their capacity levels, relating both with a date selector interface. The plans' comparison is also a key aspect of the application, allowing the user to understand if a set of changes, achieve a higher customer service level.

The final section details other tools that were developed for inserting and handling data into the implemented database model.

Chapter 6

Evaluation of the Solution

“ The only man who behaves sensibly is my tailor; he takes my measurements anew every time he sees me, while all the rest go on with their old measurements and expect me to fit them”

George Bernard Shaw

After a software’s development there is a series of tests and evaluations that have to be performed, in order to assure it meets the defined requirements. The testing itself is useful to guarantee that every class inputs, delivery the expected output. As for the evaluation there are other concerns; every class may output the desired value, without following the software’s requirements [72], which may be related to bad design or implementation. Therefore this chapter addresses the evaluation of the solution, relating the developed features (Chapter 5) with the defined requirements (Chapter 2).

6.1 Overview

The *Interactive Visualization* project specifies two requirement types that require evaluation: functional and non-functional. The functional requirements are the specific development requirements, and are analyzed to check if the intended features were implemented. The non-functional requirements are split into two categories, user experience and production planning aid. The first will verify that Microsoft’s guidelines were followed; as for the latter, several examples will be presented to describe if the aiding requirement was successfully met, providing statistics on delivery times.

6.2 Functional Requirements

The functional requirements were defined as necessary features for achieving the overall project goals (Section 2.2). Following the performed analysis (Subsection 2.2.1), this subsection relates the project's needs to what was developed.

The overall use cases and functionalities were covered, with some improvements and added features. The *Production Plan Orders List*, *Production Plan Orders Visualization Overview*, *Production Plan Orders Visualization Capacity Overview* GUIs were developed, all following the requirements specification. The *Date Selector Control* was a tool particularly well developed, with several new concepts being fit into one control, capable of integrating all graphical user interfaces while also being practical, innovative and visually appealing.

Following up on the use cases, a summary is displayed in Table 6.1, stating if these are complete, limited or not available.

Use Case	State
IV-UC01: Load Production Plan	Complete
IV-UC02: Visualize Production Plan Orders List	Complete
IV-UC03: Search Production Plan Orders List	Complete
IV-UC04: Apply Filters to Production Plan List	Complete
IV-UC05: Visualize a Production Plan	Complete
IV-UC06: Apply Filters to Production Plan	Complete
IV-UC07: Load Production Plan Scenario	Complete
IV-UC08: Compare Production Plan Scenarios	Complete
IV-UC09: Visualize Production Plan Capacity Levels	Limited
IV-UC10: Apply Filters to Capacity View	Complete
IV-UC11: Compare Finite and Infinite Capacity	Complete
IV-UC12: Select Time Frame	Complete

Table 6.1: Use Cases - Evaluation

All the defined use cases were developed, with only the *Visualize Production Plan Capacity Levels* described as *limited*. This use case is achieved by the *Capacity Utilization GUI* and is a successful implementation, however the period selection is limited because it only allows the user to view weeks. Although weeks is one of the most used time units, the period selection should be flexible, using a time unit down to days, but able to group up to at least a year.

All features (Appendix A) were successfully implemented, with a few others added to the list, such as: Customer Service Level KPI Filtering by Customer Type, Customer Service Level Lateness Filtering and Production Plan and Scenario Statistics, these are among the most relevant for achieving the project's goals.

6.3 Non-Functional Requirements

The non-functional requirements are harder to evaluate due to their ambiguity, by acquiring different meanings according to the perspective. To assess this project's success two components must be analyzed, the user experience, and the planning aid provided by the software.

6.3.1 User Experience

In the beginning of software development, the main goal was to meet the functional requirements, and user friendliness was a secondary or non-existing objective. With software stabilization, companies now have to compete for this area, which in most cases, will make the difference between its widespread usage or its demise. With this in mind, several studies have been performed, resulting in standards and guidelines, ranging from how to layout the visual controls, down to a label's name or color. This latter concept is relevant to ensure the user finds his way throughout the menus and options. Best practices dictate that when designing a graphical user interface, cluttering must be avoided, big letters and buttons are preferred, with common options directly available and less used choices accessible through menus [73] [31].

The Microsoft Dynamics User Experience department, was created to address this area, and has produced *User Experience Guidelines for Dynamics AX 2009* [20], for aiding development. This package includes examples of how to use Dynamics AX controls, and where content should be laid out on the screen.

The project was developed using *WPF Prototyping Framework 1.0 for Microsoft Dynamics*, which provides a similar look and feel to the WinForms released version. With this toolkit and following the UX guidelines, the solution behaves as the latest Dynamics AX version; content is placed in the specified areas, divided into the navigation menu, facts pane, filters pane, actions pane and the main content area.

Some new visual concepts were introduced into the charts themselves, providing a new experience while ensuring the standards are being followed; therefore the user experience requirement was completely fulfilled, and is conform with the Dynamics AX specification.

6.3.2 Planning Aid

A software that provides different perspectives, needs to be evaluated in terms of easing and improving the user's work. In addition to providing alternative views over planning, another main goal of this project is to allow to user to gather knowledge on how the planning is working, and find out where to improve it.

Through this solution's planning and capacity views, the user should be able to determine which resources are overloaded and may represent a source of problems (also known as bottleneck). This subsection introduces some examples of how the software is able to aid the user's planning, providing statistics on actual improvements. A simple production plan will be presented to illustrate the steps the user takes to find a solution, followed by a more complex example in which the solution will not be as clear.

For providing the necessary data for drawing plans and scenarios, a model company needed to exist. The modeled business was a Bike constructor, whose possible resource types, capabilities, and routes have been defined in Appendix B.

6.3.2.1 Simple Production Plan

For a better understanding on how to use this solution a simple example is provided, by defining a single route and verifying that the results are according to what is expected.

Situation Overview

The route was picked and production orders were randomly generated (using the software defined in subsection 5.2.3.2), every two days, during a two weeks span. The plan overview in Figure 6.1 was achieved, also displaying lateness statistics (note that the following figures have been clipped to focus on the important aspects).

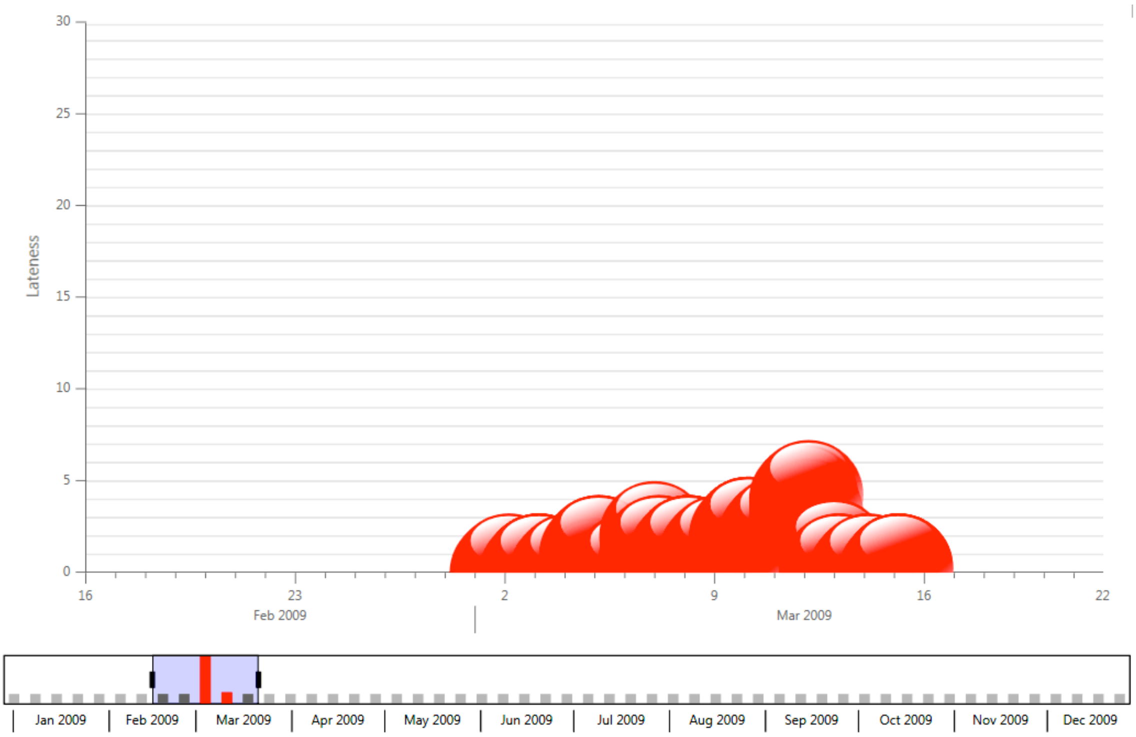
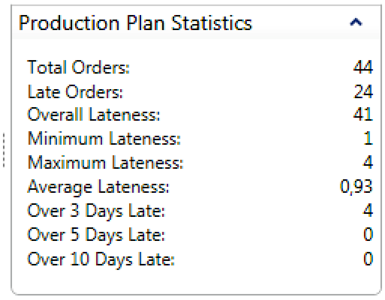


Figure 6.1: Simple Plan - Original Production Plan

To further analyze the production plan, other perspectives need to be considered, such as lateness statistics provided by the solution 6.2, and lateness comparisons, varying the allowed minimum value.

By analyzing the lateness statistics (Figure 6.2) the planner knows 24 orders are late. Further analyzing the *Customer Service Level KPI* (Figure 6.3), the main problem resides with gold customers, which are the ones the company wants to keep the most, representing more value or a longer lasting relation. To know if the service level increases by increasing, the delivery time is increased by one day and two days (Figures 6.4 and 6.5, respectively), the gold customers are still bellow 90%, as seen on the plan shown in Figure 6.6, with customer filtering selected by color.



Production Plan Statistics

Total Orders:	44
Late Orders:	24
Overall Lateness:	41
Minimum Lateness:	1
Maximum Lateness:	4
Average Lateness:	0,93
Over 3 Days Late:	4
Over 5 Days Late:	0
Over 10 Days Late:	0

Figure 6.2: Simple Plan - Lateness Statistics

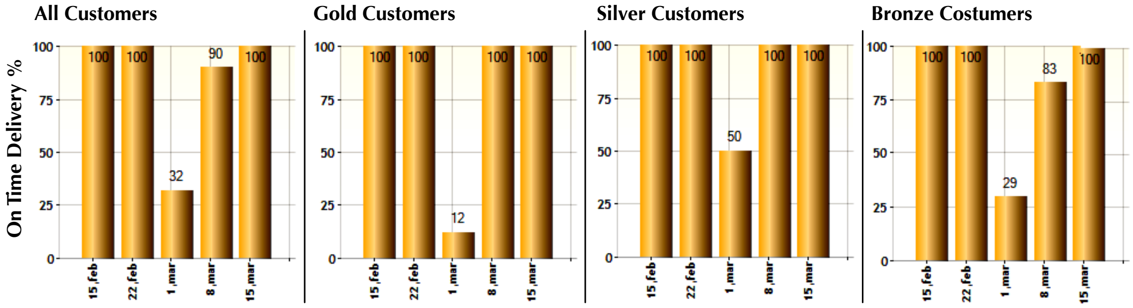


Figure 6.3: Simple Plan - Original Production Plan Customer KPI Comparison

Starting by analyzing the production plan's main orders, the given route is *Route 4 - FixeBikePaint* (Appendix B.2.2), which requires 1 person and 1 tool with capability *Paint*; the available resources (Appendix C.1) that fill these requirements are 2 persons and 2 tools. The production manager in charge of planning is aware of these details and knows what usually are the main bottlenecks in the system; by following this premise it becomes simple to setup the correct capacity filters, which in this case are *RT:Person - Cap:Paint* and *RT:Tool - Cap:Paint*. Figure 6.7 displays the finite and infinite views.

The capacity is set to the best perspective (first week on the right, second week on the left), showing that when switching from finite to infinite, the first week increases from

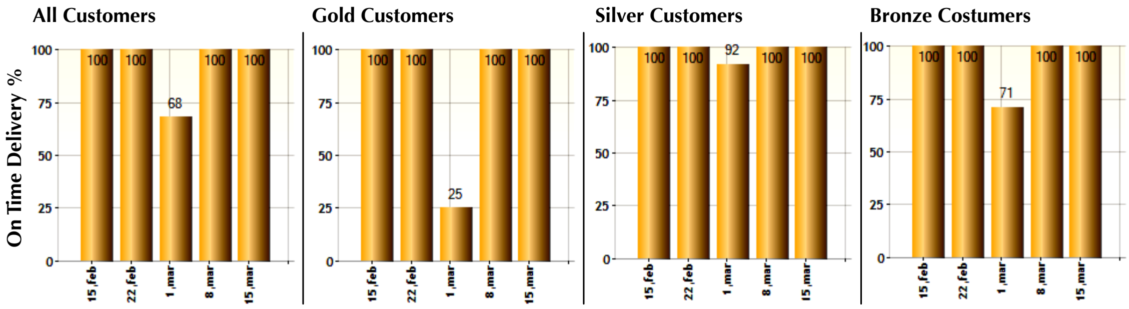


Figure 6.4: Simple Plan - Original Production Plan Customer KPI Comparison - 1 Day Allowed Lateness

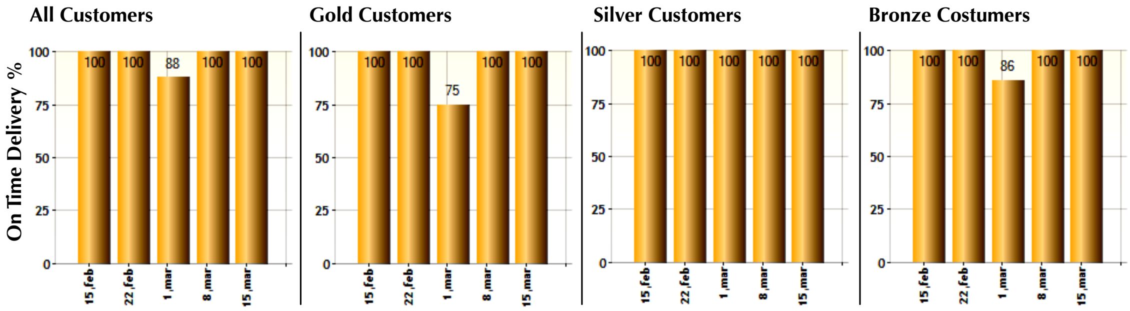


Figure 6.5: Simple Plan - Original Production Plan Customer KPI Comparison - 2 Day2 Allowed Lateness

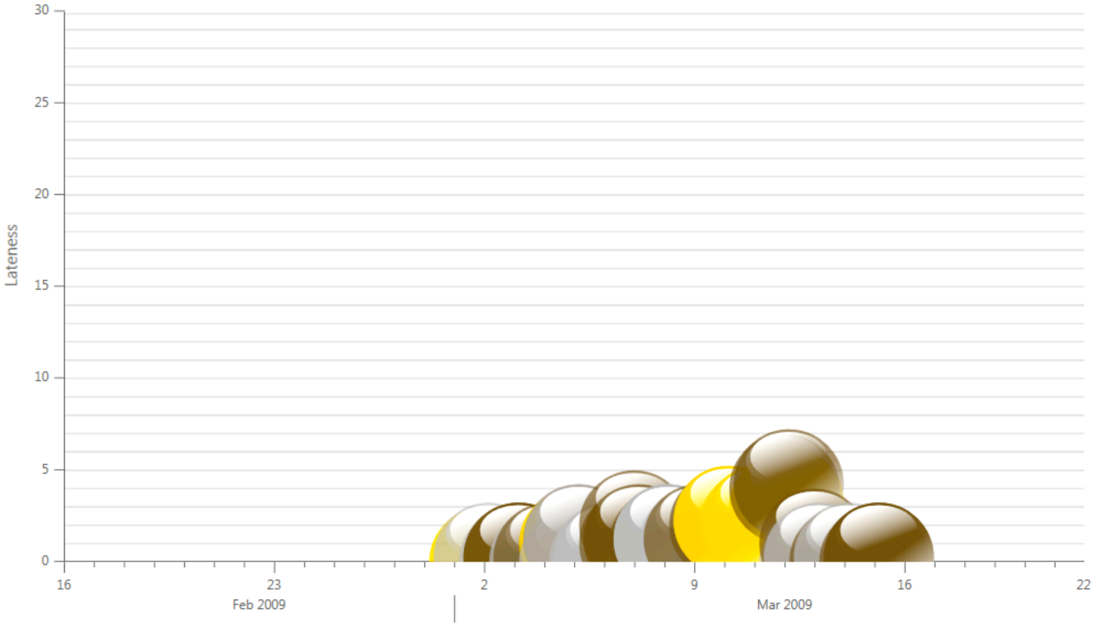


Figure 6.6: Simple Plan - Original Production Plan Customer Filtering

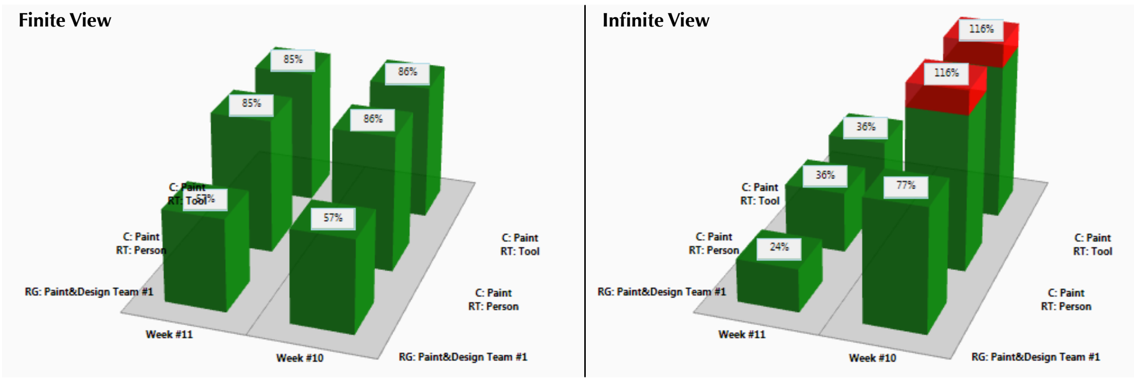


Figure 6.7: Simple Plan - Original Production Plan Finite/Infinite Capacity View

86% to 116% in *RT:Person/Tool - Cap:Paint*, and concluding what resources *Person* or *Tool*, with capability *Paint*, are necessary and may represent the system's bottleneck. To test this solution, more resources will be added to try to improve the customer service level, and the lateness statistics.

First Scenario

The same plan is executed, increasing the number of resources *RT:Person/Tool - Cap:Paint*. One of each resources will be added to the plan; the totals are displayed in Table 6.2.

Resource Type	Capabilities	Available Resource Number
Person	Pt, Dsgn	3
Tool	Pt	3
Tool	Dsgn	3
(...)	(...)	(...)

Table 6.2: Simple Plan - First Scenario Resources Summary

After executing the production plan, the planner will follow the same proceeding, as in the original plan, to find out if any improvements occurred. Loading the first scenario and comparing it to the original plan shows several improvements, as displayed in Figure 6.8. Only 2 orders remain late, both for bronze customers, which is displayed by the *Customer Service Level KPI* in Figure 6.9.

Increasing the resource number by one unit of each of the required resource types with capability *Paint*, will drastically reduce the number of late orders.

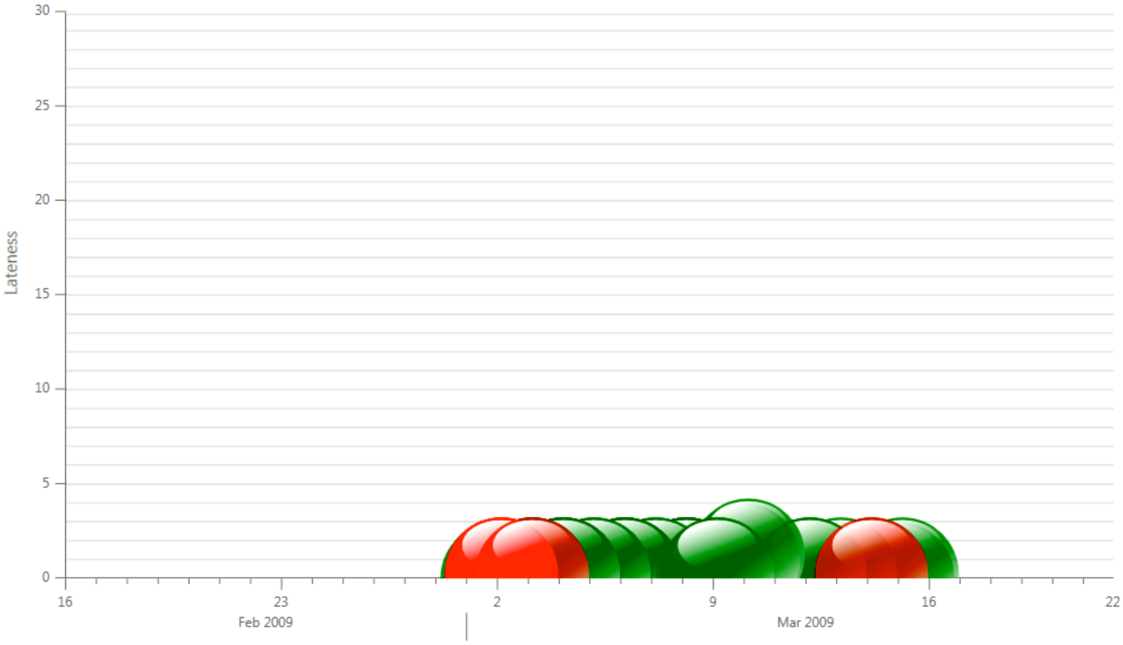


Figure 6.8: Simple Plan - Original Production Plan and First Scenario Comparison

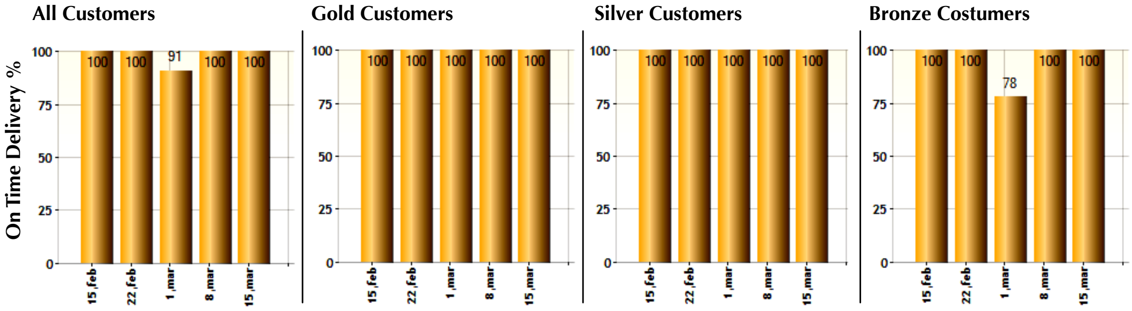


Figure 6.9: Simple Plan - First Scenario KPI Comparison

Second Scenario

For attempting to reach an overall customer service level of 100%, the resources will be increased again by two units of the required type and capability. The totals are displayed in Table 6.3.

Resource Type	Capabilities	Available Resource Number
Person	Pt, Dsgn	4
Tool	Pt	4
Tool	Dsgn	4
(...)	(...)	(...)

Table 6.3: Simple Plan - Second Scenario Resources Summary

After executing the production plan, the planner will follow the same proceeding to find out if any improvements occurred. By loading the second scenario and comparing it to the original plan, no late orders appear, as displayed in Figure 6.10, achieving 100% service level.

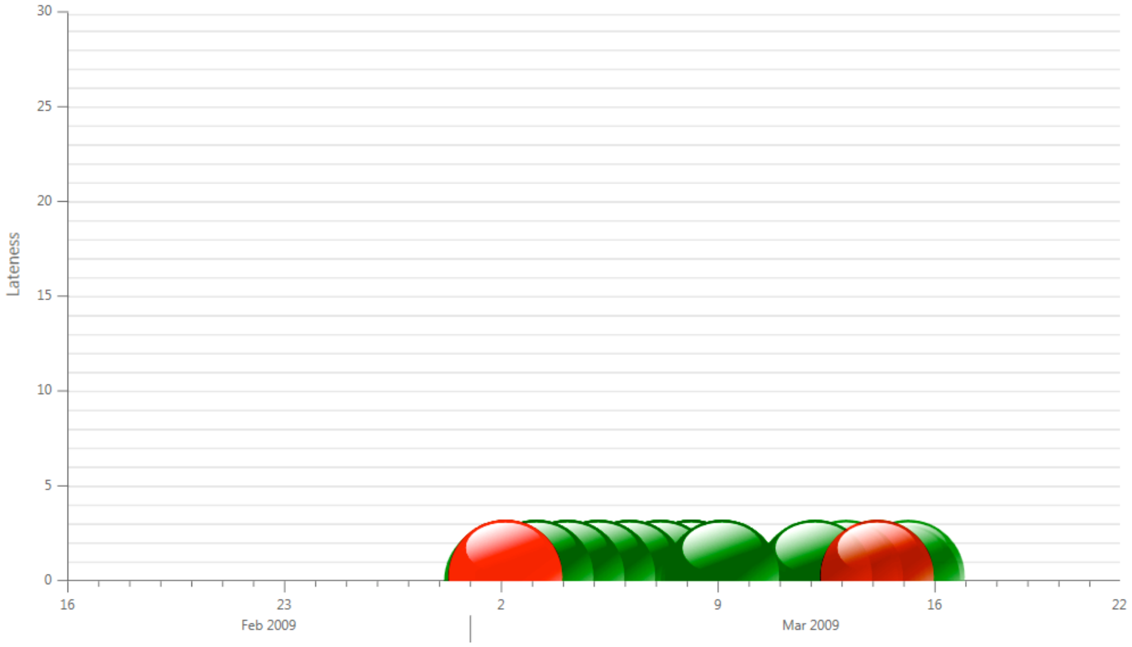


Figure 6.10: Simple Plan - Original Production Plan and Second Scenario Comparison

This solution is the best for meeting the desired delivery times, but several other business factors must be considered. In a company, it is not ideal to have excessive workforce, and this scenario adds two new workers, resulting in an extra cost. The benefits of improving the service level would have to be balanced with the costs of hiring more personnel, or acquiring new resources.

Simple Plan - Conclusions

Following these steps, it was possible to analyze several aspects of the production plan, and the planner was able to determine where the problem sufficed. Several solutions are available:

- Re-prioritize orders - By setting new priorities the planner is able to fulfill the gold customers demands first, then the silver and finally the bronze customers.
- Follow First Scenario - The first scenario is a viable solution, decreasing from 24 late orders to two, belonging to bronze costumers, each with only one day in lateness.
- Follow Second Scenario - The second scenario is the solution for delivering all orders on time, may not be an alternative due to increase of costs.

This example demonstrates how the *Interactive Visualization* solution aids in planning, by providing different views and comparing scenarios. The decision is always up to the planner, he/she can follow any of these solutions and achieve a more efficient plan, or create yet new scenarios and compare them with the original plan.

6.3.2.2 Complex Plan

This plan has been setup to provide a real life situation on how to find problems in the production plan, and run different planning scenarios to improve the delivery times; the real life situation is provided by a bike building company based on the routes and resources from Appendix C.

Situation Overview

In 2008, the company was able to deliver on time, but a raise in the number of sales orders has lead to possible complications for assuring on-time delivery. This plan starts in January 2009, when the production orders have been introduced into the ERP system and scheduled; and an overview of their delivery times and lateness is displayed in Figure 6.11.

An analysis of the production plan overview, shows that the lateness levels until August 2009 do not go above five days, which is an acceptable value for the company's standards. Between August 2009 and October 2009, the number of orders increases dramatically, leading to a vicious cycle of late orders, only recovering in November. Further details about lateness levels and customer service levels are presented in Figures 6.12 and 6.13, respectively. If the lateness was raised to 15 days, the outcome would be a desired one, as displayed in Figure 6.14, but this does not represent a good alternative.

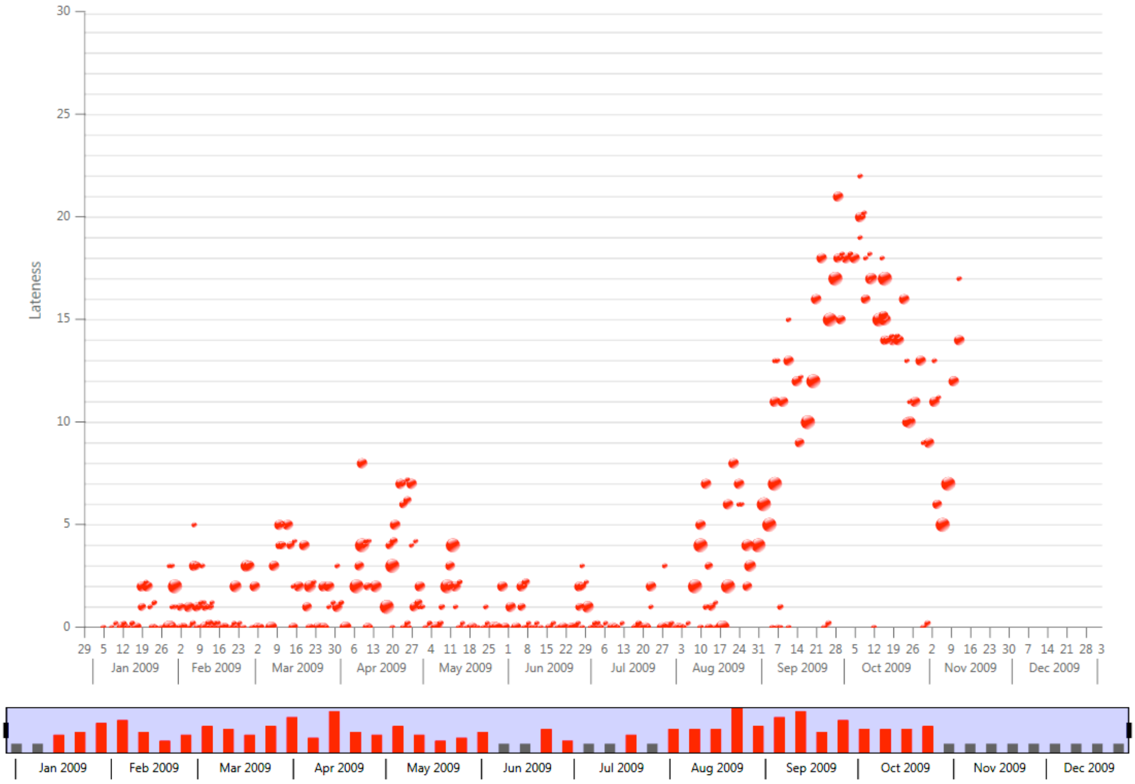
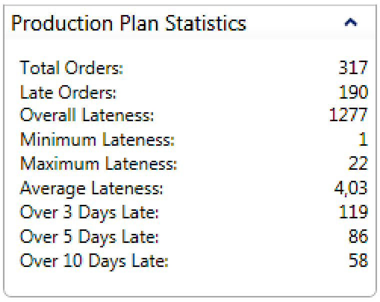


Figure 6.11: Complex Plan - Original Production Plan



Production Plan Statistics	
Total Orders:	317
Late Orders:	190
Overall Lateness:	1277
Minimum Lateness:	1
Maximum Lateness:	22
Average Lateness:	4,03
Over 3 Days Late:	119
Over 5 Days Late:	86
Over 10 Days Late:	58

Figure 6.12: Complex Plan - Original Production Plan Lateness Statistics

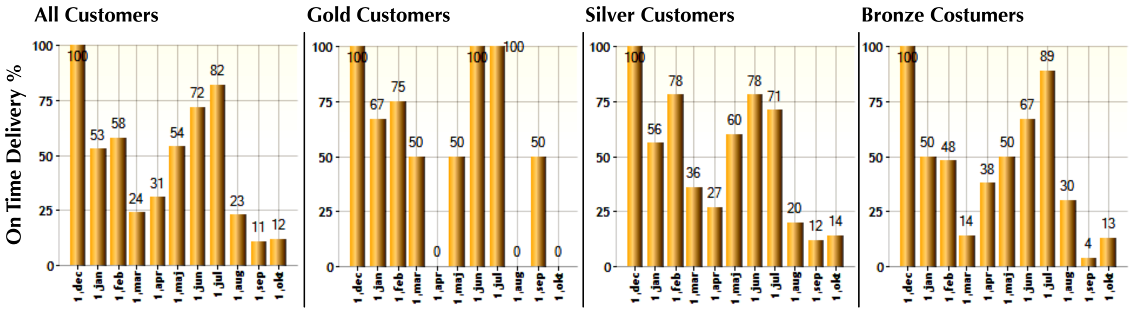


Figure 6.13: Complex Plan - Original Production Plan Customer KPI Comparison

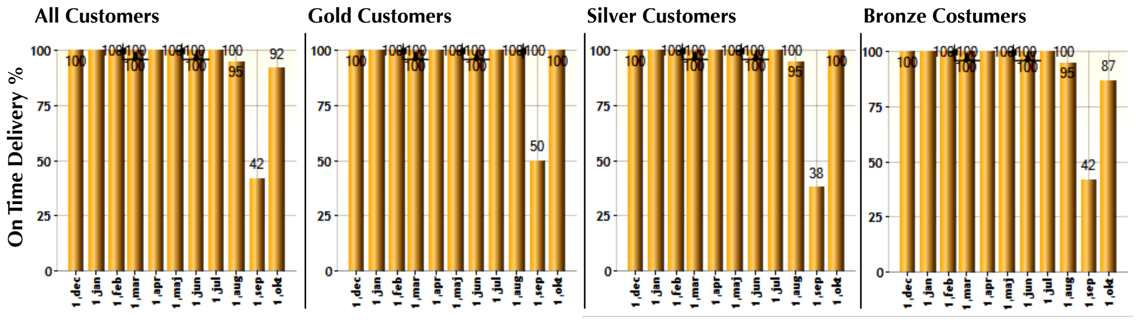


Figure 6.14: Complex Plan - Original Production Plan Customer KPI Comparison - 15 Days Allowed Lateness

The overall picture shows that, for the current production plan, there are not enough available resources, therefore an analysis of their capacity levels is required, for obtaining more information. Due to the high number of orders, the problem area of the plan was selected, as shown in Figure 6.15.

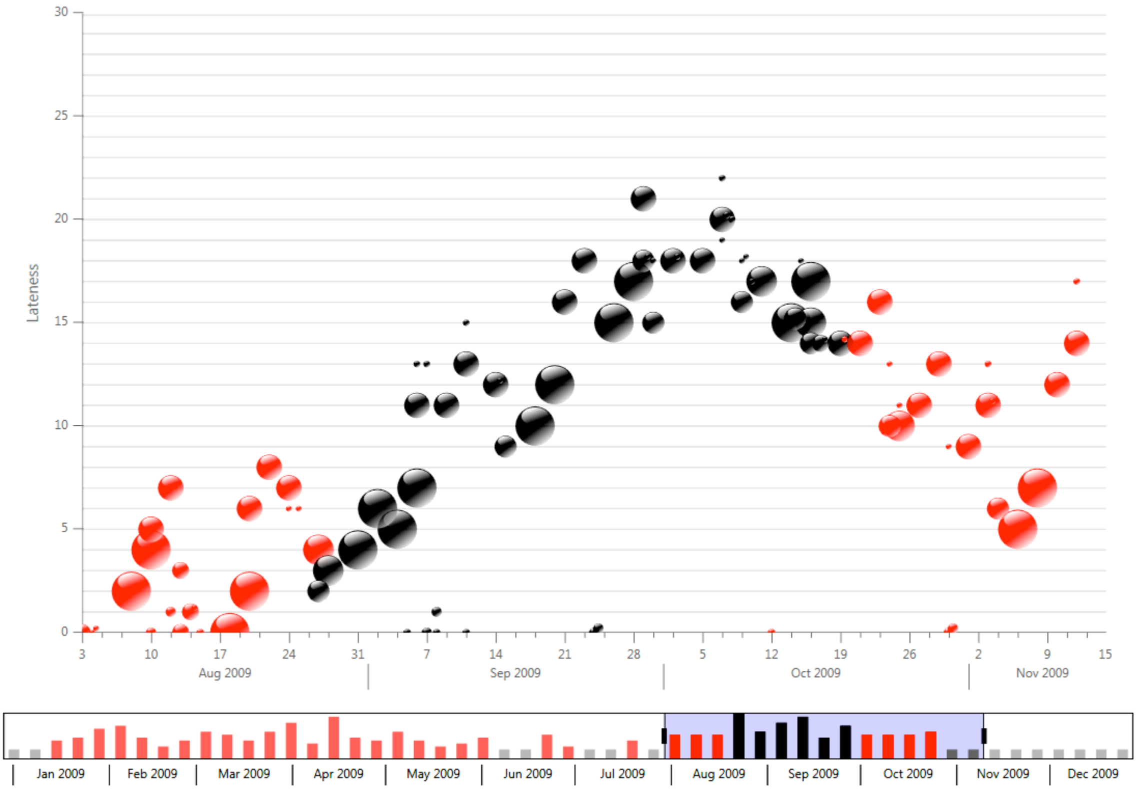


Figure 6.15: Complex Plan - Original Production Plan Problem Area - From August to October

For the capacity view, the user must determine which filters to select, as the high number of orders makes it impractical to go through them one by one, verifying what route they correspond to. Hence, the strategy therefore is to look at capacity levels from a resource group perspective, as displayed in Figure 6.16. Every resource is associated with a resource group, aggregated to perform similar tasks, such as resource group *"Assembly Team #1"* (as show in the previously described concepts in the problem description Subsection 2.1.2, and in the bike company details in Appendix C).

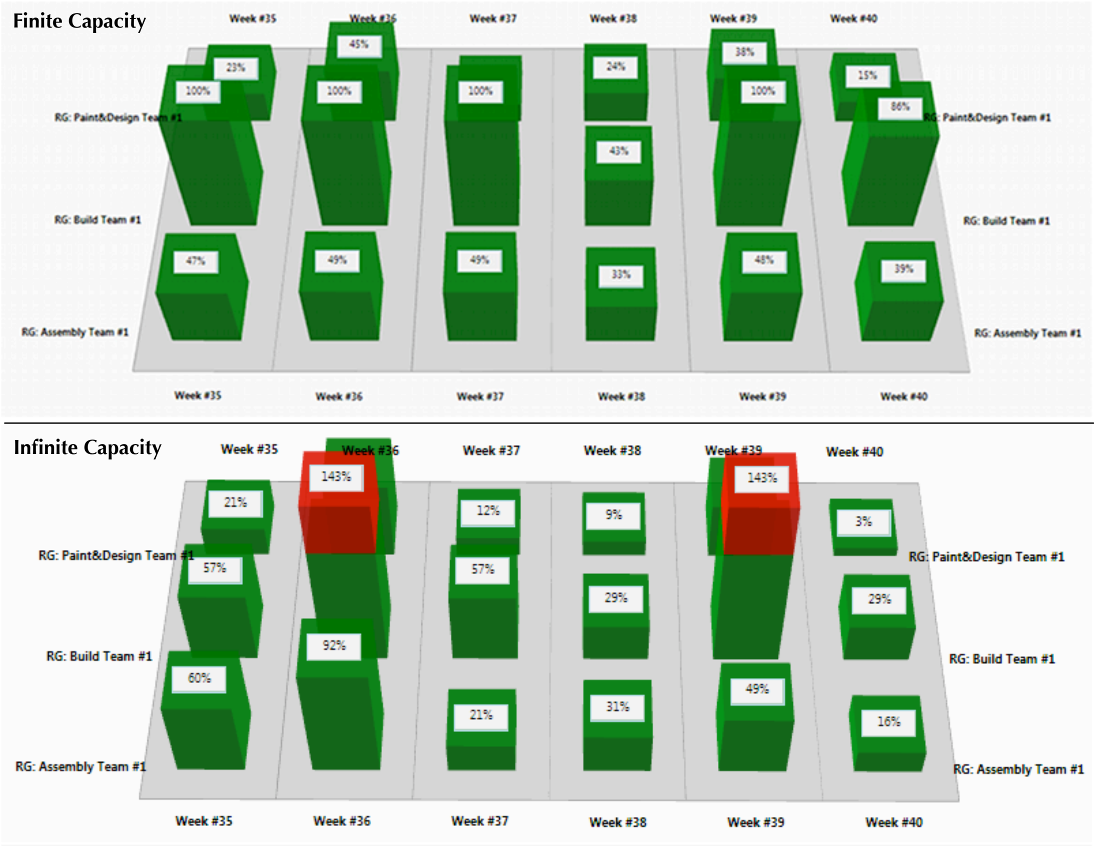


Figure 6.16: Complex Plan - Original Production Plan Finite/Infinite Capacity View by Resource Type

The capacity view shows the planner that the main overload is in resource group *Build Team #1*. Due to the planner's knowledge of the business, he knows this resource group has three types of *resources*, *person*, *tool* and *machine*, with capability *Build*. Therefore new filters are setup for viewing the capacity levels; Figure 6.17 displays the outcome.

The main problem appears to be in the resource type *Machine* with capability *Build*, but the other resource types, person and tool, also seem cluttered. The planner is adding one machine, but also knows that more people are required to handle the new machine, so three other resources of type *Person* and three others of type *Tool* are added, and a scenario is scheduled.

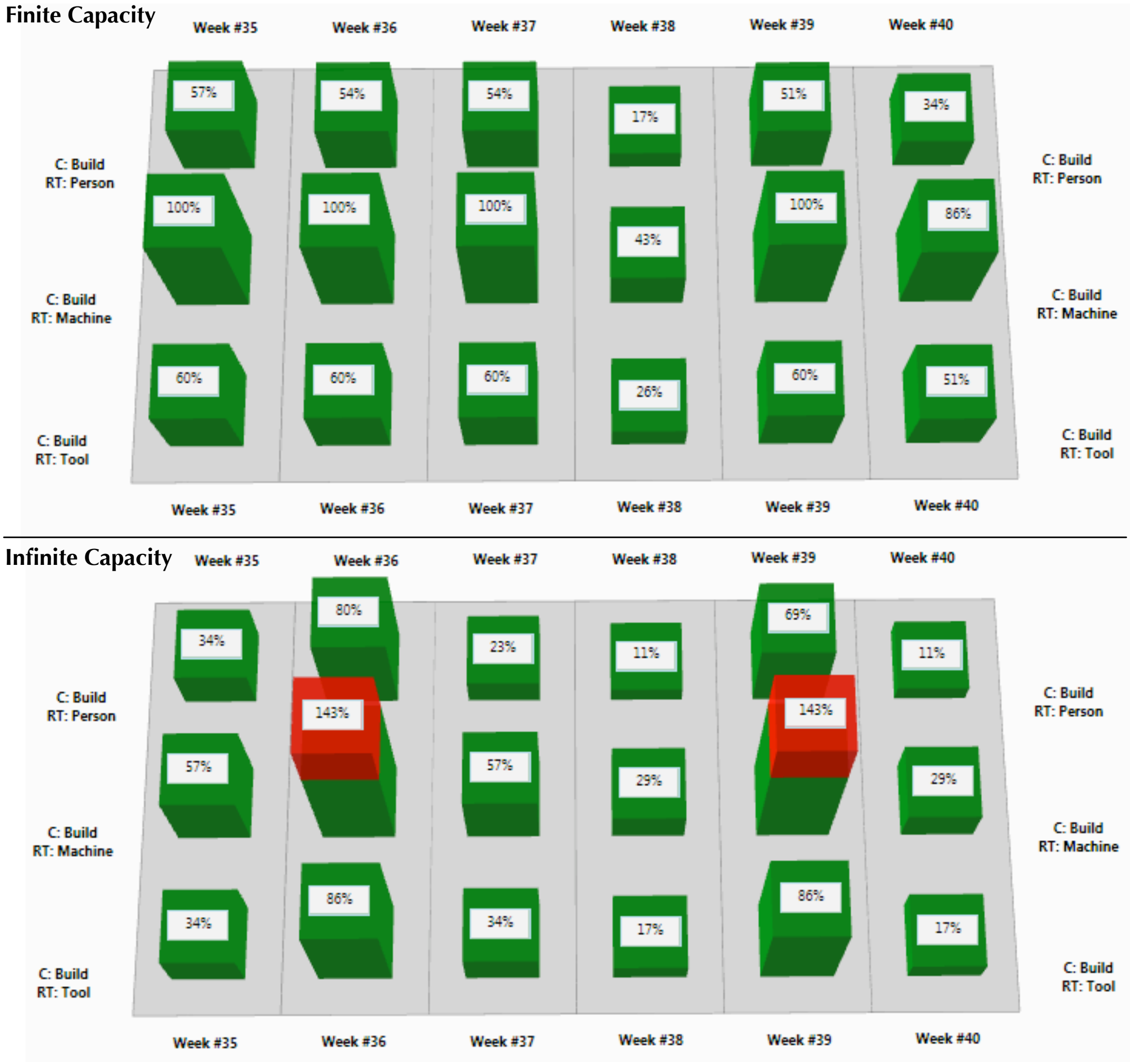


Figure 6.17: Complex Plan - Original Production Plan Finite/Infinite Capacity View by Capability

Scenario

This scenario contains the resources from the original production plan, with a few add-ons, 1 *Machine*, 3 *Person* and 3 *Tool*, all with capability *Build*. The planner executes the scheduling, and loads the new scenario into the planning environment. The results of the comparison between both schedules, statistics, and KPI with 0 days compared to 5 days of lateness, are displayed in Figures 6.18, 6.19 and 6.20, respectively.

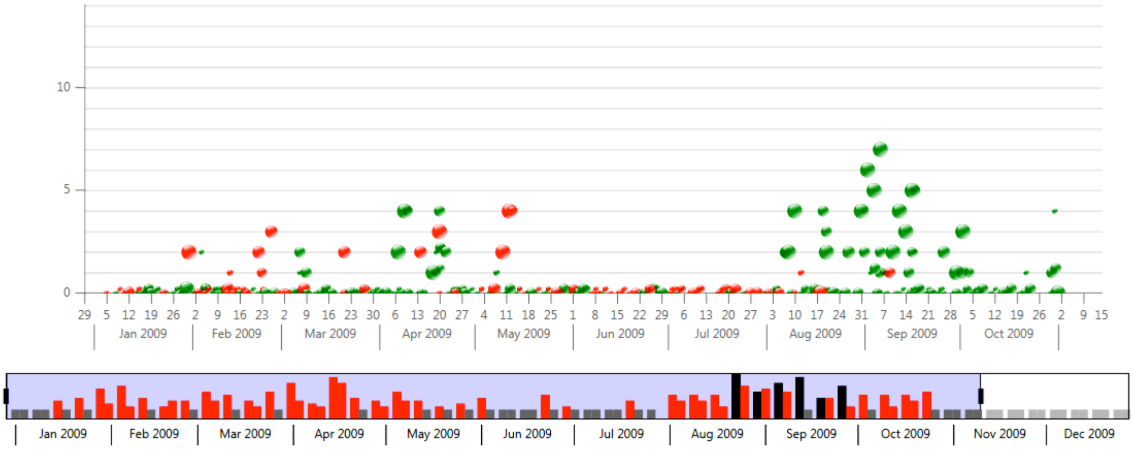
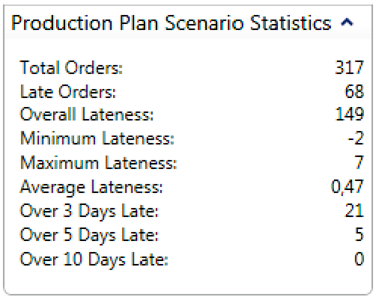


Figure 6.18: Complex Plan - Original Production Plan and Scenario Comparison



Production Plan Scenario Statistics ⌃

Total Orders:	317
Late Orders:	68
Overall Lateness:	149
Minimum Lateness:	-2
Maximum Lateness:	7
Average Lateness:	0,47
Over 3 Days Late:	21
Over 5 Days Late:	5
Over 10 Days Late:	0

Figure 6.19: Complex Plan - Scenario Statistics

Complex Plan - Conclusions

By analyzing the results, the planner is able to see that most of the orders are within the 5 day lateness level, as clearly stated by the KPI in Figure 6.20. Furthermore, the overall lateness has been reduced drastically, along with its average. The outcome is a plan where most of the delivery times are within the company's customer service level. The decision of executing this scenario is again a responsibility and choice of the planner, either by turning it into the main schedule or to run over more scenarios.

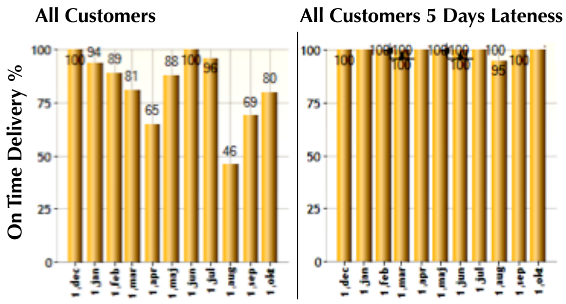


Figure 6.20: Complex Plan - Scenario KPI Comparison

6.4 User Input

To assess if the solution is successful and represents a step forward in visualization, a presentation at Microsoft was performed before an experienced audience. Most of the present were Program Managers, with some exceptions for very experienced Developers, representing a closer view of what a customer would think of this product, and if it fits into Dynamics AX.

After this presentation, the audience was required to complete a survey, addressing some of the concerns about the project's success and future applicability. Questions and results are available in Appendix D.

The audience represented more than a cumulative 150 years of overall work in this field, many experienced in working with customers. The results (Appendix D.2), are very impressive, every respondent chose *Strongly Agree* or *Agree* for every of the statements presented, reflecting the positive reception of the tool developed within this project.

The audience was mainly composed of people with experience in working with costumers, and with an above average level of understanding of the problems within a *Manufacturing* environment.

The results show that the system could be used in Dynamics AX, aiding in production and capacity planning, while also proving useful for other types of planning. From the perspective of the audience, the integration of such a project into Dynamics AX would highly appeal to the market.

For further approval this survey would have to be conducted directly with customers, which represent the target for the Dynamics AX product, and the *Interactive Visualization* project.

6.5 Summary and Conclusions

When a project reaches its conclusion there is the need to determine its success. This chapter reviewed how the initial project requirements fit the developed solution, starting with an overview of the functional requirements. This brief explanation concluded that from a functional perspective the goals of the project were fully achieved.

A project based on developing user interfaces, has other components that need evaluation, such as user experience, and other non-functional requirements. An analysis of the project's user experience followed, comparing it with the guidelines introduced by Microsoft for developing software for the Dynamics AX ERP system.

The non-functional requirements also evaluate if the project is able to perform the tasks for which it was designed; to achieve this, two examples were presented on how to use to software. A simple production plan was outlined and its problem resolution was fully detailed, followed by a complex production plan, which relates to real-life planning experience. Both usages were successful and it was concluded that the user would achieve better planning results by setting up different scenarios and subsequently comparing it with the original plan.

The final section focused on user input. A presentation was performed to an experienced audience and a survey was delivered, to determine if the software would appeal to actual customers and assess its applicability. The results demonstrate that from the Program Manager's perspective, which within a development environment is as close as possible to customers, this project would be highly accepted in the market, and may be introduced into later versions of Dynamics AX.

Chapter 7

Conclusions and Future Work

"Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away."

Antoine de Saint-Exupery

The final chapter of this thesis will resume the project as a whole, outlining the several stages and assessing its overall success, pointing out if the requirements were met, and highlight the project's innovations, limitations and suggestions for future work.

7.1 Innovations

This project represented a challenge in that, its key objectives were to provide an innovative way for visualizing production planning, aiding the user to improve the plan, while still developing an appealing application.

The most relevant innovations were based on research on visualization approaches, presented in the *State Of The Art* chapter and are detailed below:

1. Data Selection with Look Ahead: The *Date Selector GUI* was essential for providing integration between the *Production Plan GUI* and the *Capacity Utilization GUI*, and one of its main achievements was the *look ahead* feature. If the user has selected a specific time frame, it is still possible to know if other time periods have late orders, using a 2D bar chart available within this control.

2. Finite and Infinite Capacity: The ability to compare finite and infinite scheduling allows the user to know what would happen if there were unlimited resources, aiding in problem and bottleneck detection, and representing a key innovation.

3. Orders and Capacity: The integration of the two GUIs allows the user to relate specific time periods with the reserved capacity levels, providing a direct way to relate both concepts. Currently, in Dynamics AX, these features are separated, which complicates the problem detection.

4. Simple view: The *Production Plan GUI* is an intuitive and practical user interface, that allows users to understand what its functionalities are. This is a key aspect for developing a graphical user interface, designing uncluttered views which users are able to understand straight away.

5. Drill-down: For obtaining a clearer view of the production plan, key concepts were selected for providing a global view of the plan's status while still allowing the user to drill down to other concepts.

6. Scenario comparison: Within Microsoft Dynamics AX, this scenario comparison represents an innovation, due to the lacking within the software of what-if scenarios capabilities. This feature allows the user to locate possible bottlenecks, alter the resources that may be the root of the problem, re-run the schedule and compare the new planning scenario with the original. This is an efficient tool for determining where the problem lays.

7.2 Limitations

As with every software, this tool also has its limitations. Causes for these were mainly time constrains, and the need to develop a working prototype within a short time frame. Although the project's goals have been achieved, they still only address half of the production planning problems.

From previous studies (Section 1.2.1), an assessment of the *Production Planning* concepts was introduced. There are two key different types of planning for finding the root problems within the plan, resource capacity and available material. The first was addressed completely by the software, but the latter still has room for improvement. Some companies run all their planning on an infinite material basis; this software addresses this target market, but for finite material planning, a *Materials Utilization View* would have to be introduced, allowing the user to relate planned production orders, with capacity and material, simultaneously.

The other limitation is within the *Capacity and Utilization GUI* and *Date Selector GUI* integration. This capacity interface allows the user to select one or more weeks and display their capacity, and although the logic for viewing the capacity in days is built into the software, the date selector interface does not allow the user to select different time periods.

7.3 Success Evaluation

This section will describe how the original requirements (Subsection 2.2) are integrated into the developed solution, presenting a global perspective. The success details of the project were studied in chapter 6.

1. **Visual overview of Production Orders:** This requirement was fully met, as the developed software is able to display to the user, in a simple way, a full production plan, allowing the user to view the plan's *big picture*, concentrating on key concepts of *Production Planning*.

2. **Transverse through different timelines:** The *Date Selector GUI* is the tool that allows the user to go through the different time periods, in an easy way. It also provides a way to look beyond the currently selected time frame into other possible time periods, through its *Look Ahead* feature (Subsection 4.5.2).

3. **Filtering:** Every ERP system contains large amounts of data, integrated into a single framework. The filtering capabilities introduced into the software provide a way to isolate groups of orders, focusing on a specific problem, customer type or target market.

4. **Visual Capacity Utilization overview:** The *Capacity Utilization GUI* achieves the desired outcome of this requirement, providing a reserved capacity view for the specified time frame and filters.

5. **Aid in capacity problems and bottleneck detection:** The integration of the production plan and capacity views, along with the finite and infinite comparison, turn this requirement into a reality. As seen in the examples given (Subsection 6.3.2), the developed software eases problems and facilitates bottleneck detection.

6. **Provide a tool for comparing different planning scenarios:** It was essential to compare the new scenario with the original plan. This feature was fully completed, with the introduction of different colors into the *Bubble Chart*, allowing for a simple and fast comparison.

7. **Provide Key Performance Indicators over the plans:** Although a visual understanding provides an overview of the improvements in the production plan, other features that measure the actual improvements were required. The KPI feature allows the user to compare the customer service level in the original plan and in its possible scenarios, along with statistics for complementing and accurately assessing its improvements.

The project also went through a presentation at Microsoft, followed by a survey presented to an experienced audience in working with customers. The results were a success, as presented in subsection 6.4, with everyone agreeing that the project would prove to be an asset to Microsoft's ERP system, Dynamics AX.

7.4 Future Work

A future work assessment will provide insight into some features that would bring the solution to its full potential. Most of these features were considered, but due to time constrains a specific development path was chosen.

7.4.1 Date Selector GUI

The *Date Selector* is a powerful tool, which links the *Production Plan GUI* to the *Capacity Utilization GUI*. For this control to be fully customizable, the date selection for the latter user interface should be dynamic, i.e. instead of only allowing weeks to be selected, the selector should have a day as granularity, allowing the selection of more or less than 7 days.

This user interface could also introduce, along with the existing 2D bar chart, a line chart for displaying several concepts filtered by the user, such as average capacity level and average lateness level. The capacity level, would reflect the overall usage of resources; as for the lateness average, due to the bar chart system, the use of absolute numbers may prove limiting, every order may be one day late, adding up to 10 orders, or be 10 days late, adding up to 5. This feature would allow the user to focus on a specific group of orders or look at the overall order lateness.

7.4.2 Material and Capacity Integration

The *Capacity Utilization Planning* represents half of the problem, in production planning, but there is also the need for *Material Requirements Planning*. The *Capacity Utilization GUI* should integrate both concepts into one user interface, possibly with side by side 3D Charts, allowing the user to compare a specific time frame with its capacity and materials problems.

The *Production Plan Chart GUI* could also be used for comparing materials and capacity problems, e.g. if the main problem resides in materials a bubble color could turn blue, while capacity problems would turn green. This would allow the user to focus on one of the two areas and reduce the problem's scope.

7.4.3 Sales Orders

The concept of a *Planned Production Order* is a *Manufacturing* concept, but from a sales perspective, when a customer performs an order it may be separated into several production orders. The *Production Plan List GUI* could be adjusted to display sales orders, and when the user selects a specific order, it would highlight all the production orders that derive from this. The user could then focus on the main production order that has the highest lateness for this particular sales order.

7.4.4 Full integration

This project was developed using *Windows Presentation Foundation*, and although it was created as a *Control* class which makes it fully integrable with Dynamics AX, it uses a simplified database model. The business logic could be adjusted to fully integrate Microsoft's ERP system, providing the same feature set.

7.4.4.1 Filters

The filtering capabilities of a *Bubble Chart* are immense. A bubble color provides a good way for finding patterns, or focusing on a specific problem area. The filtering feature could be adjusted to handle almost every possible grouping concept in the *Manufacturing* environment, but only the main business concepts were chosen to fit with the scope of the project.

7.4.4.2 Key Performance Indicators

For assessing the scenarios improvements when comparing to the original plan, it was necessary to provide performance indicators, KPIs, but due to this project's time span, only one was chosen. The number of possible indicators, if introduced with other types of planning, such as *Materials Requirements Planning*, would increase to provide better understanding of where the problems may be.

7.4.5 Other Plans

The presentation at Microsoft raised numerous comments from the audience for integrating the same visual approaches with other types of plans. The same *Bubble Chart* concepts could be used at a higher level, such as sales orders, with the ability to drill down into specific problems within the system, creating a powerful tool. The main suggestion resided in the development of a *Material Requirements* view, along with the *Sales Orders* view. It was also suggested that the tool could be integrated with of Planning.

7.5 Conclusions

The main goal of this thesis was to develop a tool that would allow planners to have an overview of their production plans; another objective was for the tool to aid in planning. The *Persona Adaptable Visualization Scheduling in Supply Chain Management* project (referred as *Interactive Visualization* project), provided an efficient solution which fits the objectives mentioned above.

Considering the limited time span of the project, only one target environment was chosen, *Manufacturing*, and more specifically *Production Planning* and *Capacity and Utilization Planning*. An analysis of the requirements was performed, followed by a research on the state of the art technology in visualization in ERP systems and other innovative approaches. The requirements and the visualization investigation lead to a specification which followed the best practices in software and user interfaces development, as provided by Microsoft Dynamics UX guidelines [20].

The project was successfully developed and its concepts could be integrated into the Microsoft Dynamics AX ERP system. The the user (also known as planner) is able to obtain a simple view of the production plan, in a more useful way than currently allowed by table-like views, which make it almost impossible to analyze the data and detect problems. Unlike the current tools, the tool developed in this project turns the analysis of the data within ERP systems a reality. Another side-effect was the fact that by visualizing the bubble chart the planner is able to detect customer's buying patterns and its production plan consequences, by the way the orders are disposed along the timeline.

This project was also extremely successful in its presentation to an experienced audience within the field - their aspirations for the usage and implementation of this tool within Dynamics AX were astonishing. Furthermore although the project started with a specific target environment, the future work possibilities of integration with several other types of planning consolidate its extreme potential.

References

[1] Google. Eurusd - euro in united states dollar, June 2009. [Online; accessed 11-June-2009] http://www.google.com/finance?q=EURUSD.

[2] Microsoft MSDN. Wpf architecture, June 2009. [Online; accessed 1-June-2009] http://msdn.microsoft.com/pt-br/library/ms750441.aspx.

[3] WPF Tutorial. Introduction to wpf 3d, June 2009. [Online; accessed 1-June-2009] http://www.wpftutorial.net/IntroductionTo3D.html.

[4] Microsoft MSDN. 3-d graphics overview, June 2009. [Online; accessed 1-June-2009] http://msdn.microsoft.com/en-us/library/ms747437.aspx.

[5] Philippe Kruchten. Architectural blueprints - the ”4+1” view model of software architecture. November 2005.

[6] Microsoft. Microsoft history, 2009. [Online; accessed 15-May-2009] http://channel9.msdn.com/shows/History/.

[7] Wikipedia. Microsoft, May 2009. [Online; accessed 28-May-2009] http://en.wikipedia.org/wiki/Microsoft.

[8] Microsoft. Microsoft development center copenhagen, 2009. [Online; accessed 15-May-2009] http://www.microsoft.com/danmark/mdcc/default.mspx.

[9] InternetNews. Microsoft to buy navision for $1.3 billion, May 2002. [Online; accessed 23-May-2009] http://www.internetnews.com/bus-news/article.php/1038841.

[10] D. Clay Whybark Thomas E. Vollman, William L. Berry and F. Robert Jacobs. *Manufacturing Planning Control for Supply Chain Management*. McGraw Hill, Fifth edition, 2005.

[11] Stephen N. Chapman J.R. Tony Arnold and Lloyd M. Clive. *Introduction to Materials Management*. Pearson Prentice Hall, Sixth edition, 2008.

[12] Stuart K. Card Jeffrey Heer and James A. Landay. prefuse: a toolkit for interactive information visualization. 2005.

[13] Edward R. Tuffle. *The Visual Display of Quantitative Information*. Graphics Press USA, Second edition, 31 January, 2001.

[14] Michael Friendly. A brief history of data visualization. In C. Chen, W. Härdle, and A Unwin, editors, *Handbook of Computational Statistics: Data Visualization*, volume III. Springer-Verlag, Heidelberg, 2006. (In press).

[15] Michael Friendly. Milestones in the history of thematic cartography, statistical graphics, and data visualization. October 16, 2008.

[16] Wikipedia. Data visualization, May 2009. [Online; accessed 28-May-2009] http://en.wikipedia.org/wiki/Data_visualization.

[17] Wikipedia. Information visualization, May 2009. [Online; accessed 28-May-2009] http://en.wikipedia.org/wiki/Information_visualization.

[18] Michael Fruergaard Pontoppidan Arthur Greef and Lars Dragheim Olsen. *Inside Microsoft Dynamics AX 4.0*. Microsoft Press, First edition, 2003.

[19] Microsoft. Microsoft dynamics is familiar to your people, June 2006. [Online; accessed 1-June-2009] http://www.microsoft.com/dynamics/product/familiartoyourpeople.mspx.

[20] Microsoft. User experience guidelines for dynamics ax 2009, June 2009. [Online; accessed 9-June-2009] http://www.microsoft.com/downloads/details.aspx?FamilyID=ca546f4d-8628-4546-a9c6-a1df855d1600&displaylang=en.

[21] Ben Fry. *Visualizing Data*. O'Reilly Media, Inc., First edition, December 18, 2007.

[22] GanttChart.com. Gantt history, May 2009. [Online; accessed 27-May-2009] http://www.ganttchart.com/history.html.

[23] Smashing Magazine. Data visualization: Modern approaches, August 2007. [Online; accessed 28-May-2009] http://www.smashingmagazine.com/2007/08/02/data-visualization-modern-approaches/.

[24] Smashing Magazine. Data visualization and infographics, January 2008. [Online; accessed 28-May-2009] http://www.smashingmagazine.com/2008/01/14/monday-inspiration-data-visualization-and-infographics/.

[25] Information Architects. Web trend map 4, 2009. [Online; accessed 28-May-2009] http://informationarchitects.jp/web-trend-map-4-final-beta/.

[26] Marcus Weskamp. Newsmap, 2009. [Online; accessed 28-May-2009] http://newsmap.jp/.

[27] Andy Biggs. Voyage - rss feed reader, 2009. [Online; accessed 28-May-2009] http://rssvoyage.com/.

[28] Time Magazine. America by the numbers, October 2006. [Online; accessed 28-May-2009] http://www.time.com/time/covers/20061030/where_we_live/.

[29] Fidg't. Fidg't visualizer, May 2009. [Online; accessed 28-May-2009] http://www.fidgt.com/visualize.

[30] CrazyEgg. Visualize your visitors, May 2009. [Online; accessed 28-May-2009] http://crazyegg.com/.

[31] Robert Reimann Alan Cooper and Dave Cronin. *About Face 3: The Essentials of Interaction Design*. John Wiley Sons, First edition, May 07, 2007.

[32] SAP. Sap history: From start-up software vendor to global market leader, May 2009. [Online; accessed 30-May-2009] http://www.sap.com/about/company/history/index.epx.

[33] SAP. Sap facts and figures, May 2009. [Online; accessed 30-May-2009] http://www.sap.com/about/investor/pdf/SAP_FactSheet.pdf.

[34] SAP. Sap business suite, May 2009. [Online; accessed 30-May-2009] http://www.sap.com/solutions/business-suite/index.epx.

[35] SAP. Sap - xcelsius 2008, May 2009. [Online; accessed 30-May-2009] http://www.sap.com/solutions/sapbusinessobjects/sme/xcelsius/index.epx.

[36] Loren Abdulezer. *Xcelcius 2008 Dashboard Best Practices*. Sams, First edition, December 22, 2008.

[37] Oracle Corporation. History, May 2009. [Online; accessed 30-May-2009] http://www.oracle.com/corporate/story.html.

[38] Oracle Corporation. Strategic acquisitions, May 2009. [Online; accessed 30-May-2009] http://www.oracle.com/us/corporate/acquisitions/index.htm.

[39] Oracle Corporation. E-business suite, May 2009. [Online; accessed 30-May-2009] http://www.oracle.com/applications/e-business-suite.html.

[40] Oracle Corporation. E-business suite mid-sized companies, May 2009. [Online; accessed 30-May-2009] http://www.oracle.com/solutions/mid/oracle-ebusiness-suite.html.

[41] Oracle Corporation. Applications, May 2009. [Online; accessed 30-May-2009] http://www.oracle.com/applications/home.html.

[42] Oracle Corporation. Dashboards, May 2009. [Online; accessed 30-May-2009] http://www.oracle.com/technology/pub/articles/rittman_dash.html.

[43] Microsoft MSDN. Overview of the .net framework, May 2009. [Online; accessed 31-May-2009] http://msdn.microsoft.com/en-us/library/a4t23ktk.aspx.

[44] Ian Griffiths and Matthew Adams. *.NET Windows Forms in a Nutshell*. O'Reilly Media, Inc., First edition, March 26, 2003.

[45] Jesse Liberty and Alex Horovitz. *Programming .NET 3.5*. O'Reilly Media, Inc., First edition, July 25, 2008.

[46] Dino Esposito and Andrea Satarello. *Microsoft .NET - Architecting Applications for the Enterprise*. O'Reilly Media, Inc., First edition, October 15, 2008.

[47] Microsoft MSDN. Common language runtime overview, June 2009. [Online; accessed 1-June-2009] http://msdn.microsoft.com/en-us/library/ddk909ch.aspx.

[48] Microsoft MSDN. Common type system, June 2009. [Online; accessed 1-June-2009] http://msdn.microsoft.com/en-us/library/zcx1eb1e.aspx.

[49] Microsoft MSDN. Common-language interoperability, June 2009. [Online; accessed 1-June-2009] http://msdn.microsoft.com/en-us/library/730f1wy3.aspx.

[50] Microsoft MSDN. Managed execution, June 2009. [Online; accessed 1-June-2009] http://msdn.microsoft.com/en-us/library/k5532s8a.aspx.

[51] Microsoft MSDN. Compiling msil to native code, May 2009. [Online; accessed 25-May-2009] http://msdn.microsoft.com/en-us/library/ht8ecch6(VS.71).aspx.

[52] Microsoft MSDN. Visual basic, May 2009. [Online; accessed 31-May-2009] http://msdn.microsoft.com/en-us/library/2x7h1hfk.aspx.

[53] Jesse Liberty. *Programming Visual Basic .NET*. O'Reilly Media, Inc., Second edition, April 11, 2003.

[54] Microsoft MSDN. Visual c#, May 2009. [Online; accessed 31-May-2009] http://msdn.microsoft.com/en-us/library/kx37x362.aspx.

[55] Microsoft MSDN. Migrating to visual c#, May 2009. [Online; accessed 31-May-2009] http://msdn.microsoft.com/en-us/library/ms228394.aspx.

[56] Donis Marshall. *Programming Microsoft Visual C# 2008: The Language*. Microsoft Press, First edition, June 09, 2008.

[57] Microsoft MSDN. Visual c++, May 2009. [Online; accessed 31-May-2009] http://msdn.microsoft.com/en-us/library/60k1461a.aspx.

[58] Microsoft MSDN. Visual j#, May 2009. [Online; accessed 31-May-2009] http://msdn.microsoft.com/en-us/library/7xsxf8e2.aspx.

[59] Microsoft MSDN. What is jscript 8.0?, May 2009. [Online; accessed 31-May-2009] http://msdn.microsoft.com/en-us/library/xkx7dfw1.aspx.

[60] Dan Maharry Jesse Liberty and Dan Hurwitz. *Programming ASP.NET 3.5*. O'Reilly Media, Inc., Fourth edition, October 27, 2008.

[61] Microsoft MSDN. Asp.net overview, May 2009. [Online; accessed 31-May-2009] http://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx.

[62] Microsoft MSDN. Gdi+, May 2009. [Online; accessed 31-May-2009] http://msdn.microsoft.com/en-us/library/ms533798(VS.85).aspx.

[63] Akadia. Data binding in .net / c windows forms, May 2009. [Online; accessed 1-June-2009] http://www.akadia.com/services/dotnet_databinding.html.

[64] Chris Sells and Ian Griffiths. *Programming WPF*. O'Reilly Media, Inc., Second edition, August 28, 2007.

[65] Microsoft MSDN. Wpf fundamentals, June 2009. [Online; accessed 1-June-2009] http://msdn.microsoft.com/en-us/library/ms746927.aspx.

[66] Microsoft MSDN. Xaml, June 2009. [Online; accessed 1-June-2009] http://msdn.microsoft.com/en-us/library/ms747122.aspx.

[67] Microsoft MSDN. Wpf graphics, animation and media overview, June 2009. [Online; accessed 1-June-2009] http://msdn.microsoft.com/en-us/library/ms742562.aspx#milfeatures.

[68] Microsoft. Microsoft expression, June 2009. [Online; accessed 1-June-2009] http://www.microsoft.com/expression/features/Default.aspx?key=blend.

[69] Blender. Features, June 2009. [Online; accessed 1-June-2009] http://www.blender.org/features-gallery/features/.

[70] Russell Miles and Kim Hamilton. *Learning UML 2.0*. O'Reilly Media, Inc., First edition, April 25, 2006.

[71] Microsoft MSDN. Quick tour of the integrated development environment, June 2009. [Online; accessed 1-June-2009] http://msdn.microsoft.com/en-us/library/ms165088.aspx.

[72] Karl E. Wiegers. *Software Requirements*. Microsoft Press, Second edition, February 26, 2003.

[73] Russ Unger and Caralyn Chandler. *A Project Guide to UX Design: For User Experience Designers in the Field or in the Making*. New Riders, First edition, March 13, 2009.

Appendix A

Complete Feature List

A.1 Interactive Visualization System

IV-F001: Present a list of the scheduled production plans in the database

This list will provide a way for selecting the scheduled production plan the user wants to analyze.
Priority: High.
Relates to use case IV-UC01.

IV-F002: Load a scheduled production plan

With the list provided by feature IV-F001 the user will load into the system the selected production plan.
Priority: High
Relates to use case IV-UC01 and feature IV-F001.

IV-F003: List the loaded production plan's orders

Priority: High
Relates to use case IV-UC02.

IV-F004: Search for a production order

Provide an area for the user to search for a specific production order.
Priority: Medium
Relates to use case IV-UC03.

IV-F005: Filter by Customer Type

Provide a filtering feature to view only the production orders with the selected customer type.
Priority: High
Relates to use case IV-UC04.

IV-F006: Filter by Market Type

Provide a filtering feature to view only the production orders with the selected market type.
Priority: High
Relates to use case IV-UC04.

IV-F007: Switch to Production Order Visual Overview

This feature will allow the user to switch from the production order list view to the visual overview.
Priority: High
Relates to use case IV-UC05.

IV-F008: Switch to Production Order Visual Overview with selected production orders

Allow the user to select several production orders and then switch to the visual overview, displaying the selected production orders.
Priority: Low
Relates to use case IV-F007 and feature IV-F010.

IV-F009: Tooltips

Introduce tooltips into the production plan list view for aiding the user, this is an usability requirement.
Priority: Low

IV-F009: Help

Provide a help system over the controls in the view, aiding the user's navigation.
Priority: Low

A.2 Production Plan Orders Visualization Overview

IV-F101: Visually display the production orders

Provide a system for visually displaying the production orders in the form of a bubble chart.
Priority: High
Relates with use case IV-UC05.

IV-F102: Click bubble get info

Provide a way of clicking on the bubble in the chart and getting more information, in a fact box, on the correspondent production order.

Priority: High
Relates with use case IV-UC05.

IV-F103: Display lead time

Provide a way of clicking on the bubble in the chart and getting visual details on the correspondent production order's lead time.
Priority: Low
Relates with use case IV-UC05.

IV-F104: Filter by date

Provide a filtering system for selecting the time period the user wants to visualize.
Priority: High
Relates with use case IV-UC12.

IV-F105: Select time period

Provide a feature for the user to select which period to visualize when switching to the capacity view.
Priority: High
Relates with feature IV-F104.

IV-F106: Auto adjust chart labels

If the user changes the selected time period the labels in the chart must adjust to display the current time frame.
Priority: High
Relates with feature IV-F103.

IV-F107: Filter by customer type

Provide a filtering system for allowing the user to view only the selected customer types, which will be Gold Customers, Silver Customers and Bronze Customers.
Priority: High
Relates with use case IV-UC06.

IV-F108: Filter by market type

Provide a filtering system for allowing the user to view only the selected market types.
Priority: High
Relates with use case IV-UC06.

IV-F109: Filter by selected time period

Provide a filtering system for allowing the user to view only the selected production orders within a specific time period.

Priority: High
Relates with feature IV-F105.

IV-F110: Filter by maximum and minimum lateness

Provide a filtering system for allowing the user to filter out production orders that aren't within the given max and min lateness.
Priority: Medium
Relates with use case IV-UC06.

IV-F111: Filter by production order value

Provide a filtering system for allowing the user to filter out production orders that aren't at least of the specified value.
Priority: Medium
Relates with use case IV-UC06.

IV-F112: Filter by maximum and minimum lateness

Provide a filtering system for allowing the user to filter out production orders that aren't within the given max and min lateness.
Priority: Medium
Relates with use case IV-UC06.

IV-F113: Production Orders Change color

Provide a system that allows the user to view the production orders' bubbles chart in different colors, according to customer type or market type.
Priority: High
Relates with features IV-F107 and IV-F108.

IV-F114: Original Plan Key Performance Indicators

Provide KIPs over the original plan for evaluating it's state.
Priority: High

IV-F115: Load Planned Scenario

Allow the user to load a different planned scenario for comparing with the original plan.
Priority: High
Relates with use case IV-UC07.

IV-F116: Planned Scenario Key Performance Indicators

Provide KIPs over the original plan for evaluating it's state and comparing with the original plan.
Priority: High

Relates with use case IV-UC08.

IV-F117: Visually display the planned scenario production orders with different colors

When loading a planned scenario the user must be able to see differences between the production orders. The bubble chart will display all orders in the new position and their color will stay the same if the schedule hasn't been improved, if the oder is actually better it switches to green.
Priority: Medium
Relates with use case IV-UC08.

IV-F118: Planned Scenario click bubble get info

After loading the planned order the user and visualizing the production orders, the user will be able to click on a production order and retrieve information about it's previous delivery date and it's current delivery date.
Priority: Medium
Relates with use case IV-UC08.

IV-F119: Switch to Production Order Visual 3D Overview

This feature will allow the user to switch from the production order view to the resource capacity load 3D overview. The selected time periods must be a reference to what is going to be displayed.
Priority: High
Relates with use case IV-UC08.

A.3 Production Plan Orders Resource Capacity 3D Overview

IV-F201: Visually display the resource capacity load in a 3D overview

This view will display resource capacity load, in the form of bars, relating selected time periods, in weeks, with the selected filters, which may be aggregated by Capability, Resource Type, Resource Group, Capability and Resource Type or Resource Group and Resource Type.
Priority: High
Relates with use case IV-UC09.

IV-F202: Time period filtering

This view must provide a feature which allows the user to select different time periods without going back to the chart view.
Priority: High
Relates with use case IV-UC09.

IV-F203: Capacity Filters

This view must have a filter system allowing the user to aggregate capacity by: Capability, Resource Type, Resource Group, Capability and Resource Type or Resource Group and Resource Type.
Priority: High
Relates with use case IV-UC09.

IV-F204: 3D Chart labels

The 3D capacity view must be integrated with labels in the axis, so the user knows what are the selected weeks, and the filters.
Priority: High
Relates with use case IV-UC09, features IV-F201, IV-F202 and IV-F203.

IV-F205: 2D labels over every 3D Chart bar

The bars representing capacity in 3D chart view must have labels displaying the load percentage. These labels will be presented in a 2D view over the 3D world.
Priority: High
Relates with feature IV-F201.

IV-F206: 2D labels hiding

The bar's labels in the 3D view must disappear when the bar is hidden behind another 3D object.
Priority: Medium
Relates with feature IV-F205.

IV-F207: 3D World View Change

The user must be able to rotate the 3D view to fit the perspective he desires.
Priority: Low
Relates with feature IV-F201.

IV-F208: Finite and Infinite Capacity

The user must be able to switch between a finite and infinite capacity view. This will provide a problem solving tool.
Priority: High
Relates with use case IV-UC11.

IV-F209: Finite and Infinite Capacity in Planned Scenario

For a greater understanding of the differences between scenarios the user should be able to switch between finite and infinite schedules, from the original plan and planned scenario.

Priority: High
Relates with use case IV-UC11.

Appendix B

Bike Construction Company

The routes presented in this model company, have been developed for testing purposes, the applied concepts are described in chapter 2.1.2 and modeled in Figure 5.8.

The chosen company model is a Bike shop-floor, building bikes, parts and fixing them on an order basis, it contains several routes, some more complex then others. Each may require one or more resources with one or more capabilities.

B.1 Resource Types And Capabilities

Resource Types:

1. Person
2. Machine
3. Tool
4. Vendor

Available Capabilities:

1. Build
2. Tires Assembly
3. Engine Assembly
4. Brakes Assembly
5. Wiring
6. Design
7. Paint

B.2 Routes

These are the available routes in the manufacturing system. A diagram presenting their interaction will be presented follow by each operations required resource types, capabilities and the required resource number.

B.2.1 Route 1 - Build Bike M1

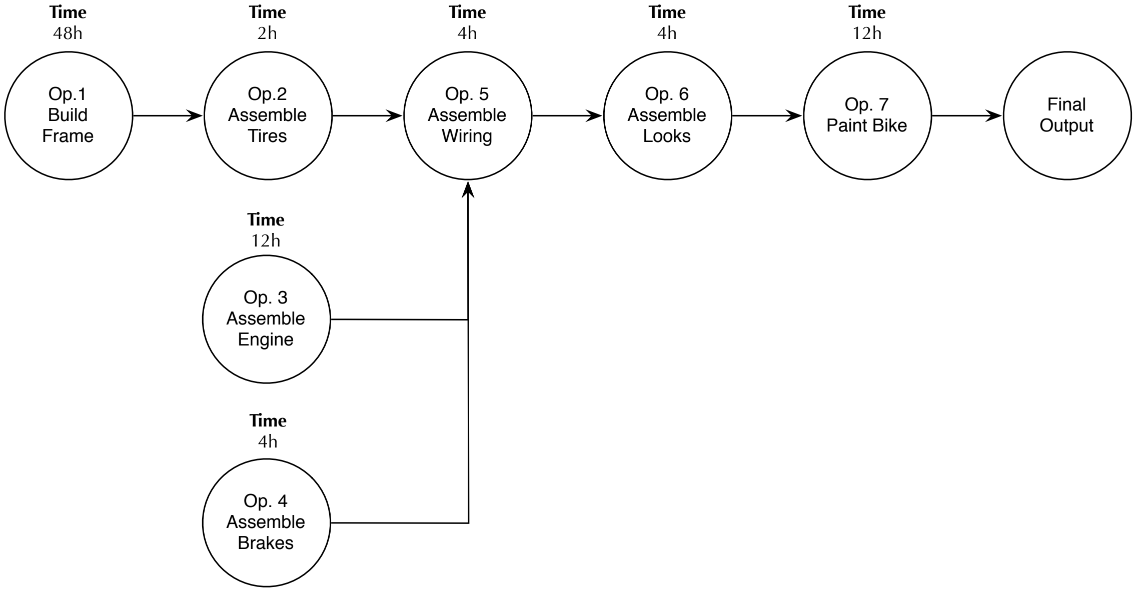


Figure B.1: Route 1 - Build Bike M1

Lead Time: 80h

- **Operation 1: Build Frame**
  Capabilities: Build
  Resources: 2 Person, 3 Tool, 1 Machine

- **Operation 2: Assemble Tires**
  Capabilities: Tires Assembly
  Resources: 1 Person, 1 Tool

- **Operation 3: Assemble Engine**
  Capabilities: Engine Assembly
  Resources: 2 Person, 2 Tool

- **Operation 4: Assemble Brakes**
  Capabilities: Brakes Assembly
  Resources: 1 Person, 2 Tool

- **Operation 5: Assemble Wiring**
  Capabilities: Wiring
  Resources: 1 Person, 2 Tool

- **Operation 6: Assemble Looks**
  Capabilities: Design

Resources: 1 Person, 1 Tool

- **Operation 7: Paint Bike**
  Capabilities: Paint
  Resources: 1 Person, 1 Tool

B.2.2 Route 2 - Fix Bike

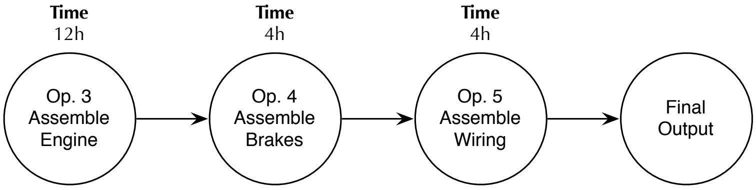


Figure B.2: Route 2 - Fix Bike

Lead Time: 20h

- **Operation 3: Assemble Engine**
  Capabilities: Engine Assembly
  Resources: 2 Person, 2 Tool

- **Operation 4: Assemble Brakes**
  Capabilities: Brakes Assembly
  Resources: 1 Person, 2 Tool

- **Operation 5: Assemble Wiring**
  Capabilities: Wiring
  Resources: 1 Person, 2 Tool

B.2.3 Route 3 - Bike Checkup

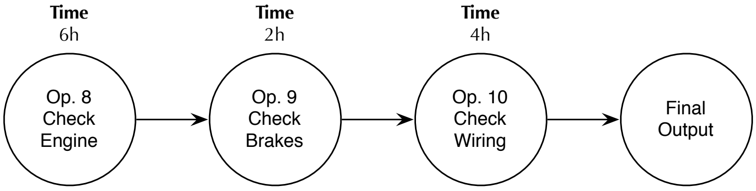


Figure B.3: Route 3 - Bike Checkup

Lead Time: 12h

- **Operation 8: Check Engine**
  Capabilities: Engine Assembly
  Resources: 2 Person, 2 Tool

- **Operation 9: Check Brakes**
  Capabilities: Brakes Assembly
  Resources: 1 Person, 2 Tool

- **Operation 10: Check Wiring**
  Capabilities: Wiring
  Resources: 1 Person, 2 Tool

B.2.4 Route 4 - Fixe Bike Paint

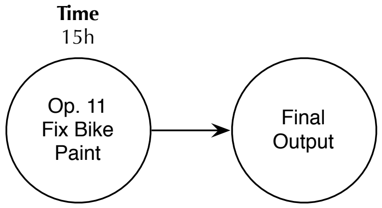


Figure B.4: Route 4 - Fix Bike Paint

Lead Time: 15h

- **Operation 11: Fix Bike Paint**
  Capabilities: Paint
  Resources: 1 Person, 1 Tool

B.2.5 Route 5 - Build Bike M2

- **Operation 1: Build Frame**
  Capabilities: Build
  Resources: 2 Person, 3 Tool, 1 Machine

- **Operation 2: Assemble Tires**
  Capabilities: Tires Assembly
  Resources: 1 Person, 1 Tool

- **Operation 12: Assemble Engine M2**
  Capabilities: Engine Assembly
  Resources: 2 Person, 3 Tool

- **Operation 4: Assemble Brakes**
  Capabilities: Brakes Assembly
  Resources: 1 Person, 2 Tool

- **Operation 5: Assemble Wiring**
  Capabilities: Wiring
  Resources: 1 Person, 2 Tool

- **Operation 13: Assemble Looks M2**
  Capabilities: Design
  Resources: 1 Person, 1 Tool

- **Operation 14: Paint Bike M2**
  Capabilities: Paint
  Resources: 1 Person, 1 Tool

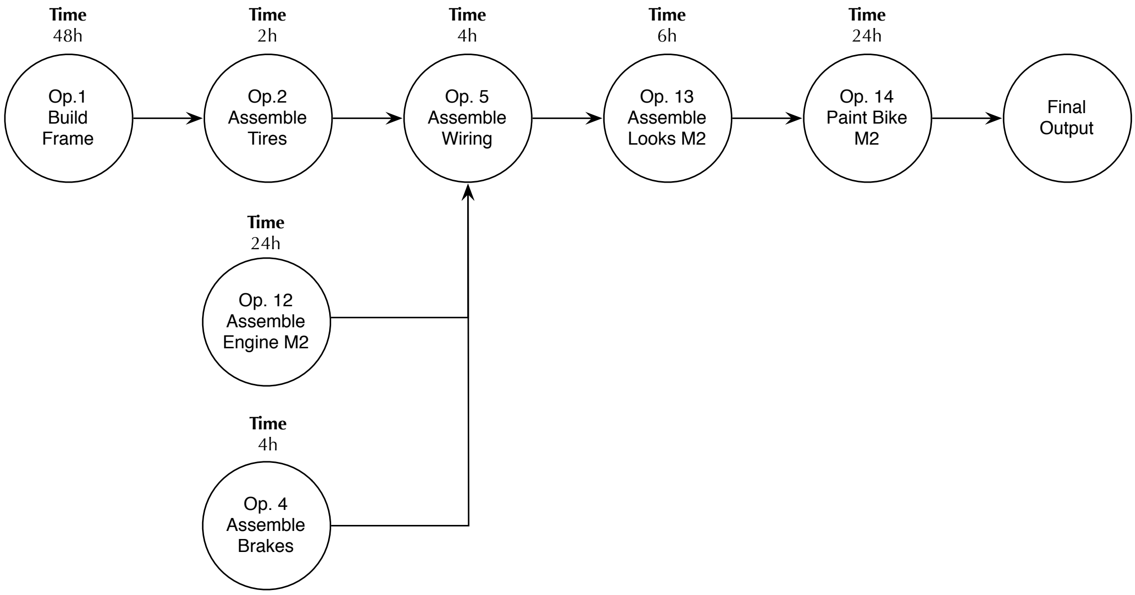


Figure B.5: Route 5 - Build Bike M2

Lead Time: 106h

B.2.6 Route 6 - Build Trike

Lead Time: 148h

- **Operation 15: Build Trike Frame**
  Capabilities: Build
  Resources: 3 Person, 3 Tool, 1 Machine

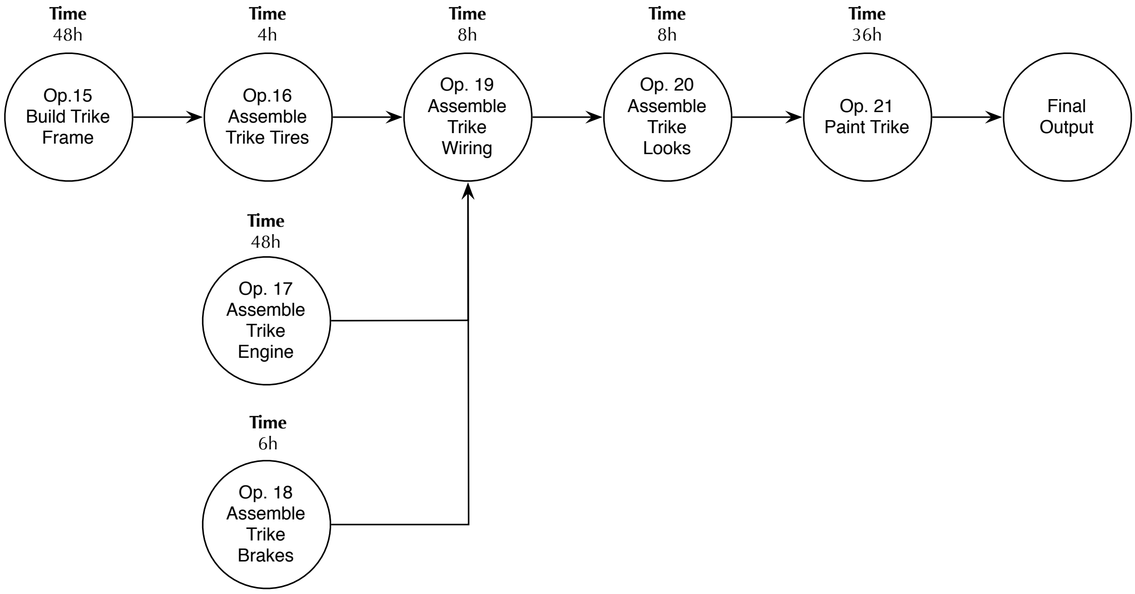


Figure B.6: Route 6 - Build Trike

- **Operation 16: Assemble Trike Tires**
  Capabilities: Tires Assembly
  Resources: 1 Person, 1 Tool

- **Operation 17: Assemble Trike Engine**
  Capabilities: Engine Assembly
  Resources: 3 Person, 3 Tool

- **Operation 18: Assemble Trike Brakes**
  Capabilities: Brakes Assembly
  Resources: 1 Person, 2 Tool

- **Operation 19: Assemble Trike Wiring**
  Capabilities: Wiring
  Resources: 1 Person, 2 Tool

- **Operation 20: Assemble Trike Looks**
  Capabilities: Design
  Resources: 2 Person, 1 Tool

- **Operation 21: Paint Trike**
  Capabilities: Paint
  Resources: 1 Person, 1 Tool

B.2.7 Route 7 - Fix Trike

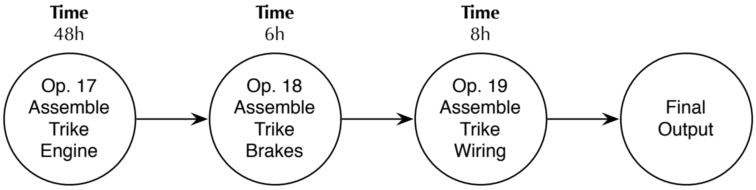


Figure B.7: Route 7 - Fix Trike

Lead Time: 62h

- **Operation 17: Assemble Trike Engine**
  Capabilities: Engine Assembly
  Resources: 3 Person, 3 Tool

- **Operation 18: Assemble Trike Brakes**
  Capabilities: Brakes Assembly
  Resources: 1 Person, 2 Tool

- **Operation 19: Assemble Trike Wiring**
  Capabilities: Wiring
  Resources: 1 Person, 2 Tool

B.2.8 Route 8 - Trike Checkup

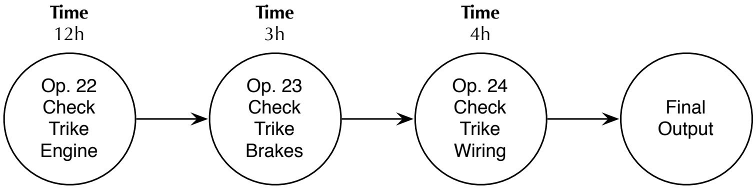


Figure B.8: Route 8 - Trike Checkup

Lead Time: 19h

- **Operation 22: Check Trike Engine**
  Capabilities: Engine Assembly
  Resources: 2 Person, 2 Tool

- **Operation 23: Check Trike Brakes**
  Capabilities: Brakes Assembly
  Resources: 1 Person, 2 Tool

- **Operation 24: Check Trike Wiring**
  Capabilities: Wiring
  Resources: 1 Person, 2 Tool

B.2.9 Route 9 - Fix Trike Paint

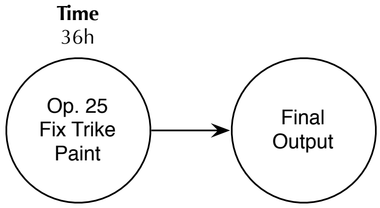


Figure B.9: Route 9 - Fix Trike Paint

Lead Time: 36h

- **Operation 25: Fix Trike Paint**
  Capabilities: Paint
  Resources: 1 Person, 1 Tool

Appendix C

Bike Company

C.1 Available Resources

The following resources are the requirements for the operations defined in Appendix B. For simplification the resource type and capabilities have been shortened. Table C.4 summarizes the resources defined in table C.3, grouping them by resource type and capabilities.

Capability	Short Name
Build	Bld
Tires Assembly	TirA
Engine Assembly	EngA
Brakes Assembly	BrkA
Wiring	Wrng
Design	Dsgn
Paint	Pt

Table C.1: Capability Shorten Names

Resource Type	Short Name
Person	P
Machine	M
Tool	T
Vendor	V

Table C.2: Resource Type Shorten Names

Resource Name	Resource Type	Capabilities	Resource Group
PersonDesignerPaintor1	P	Pt, Dsgn	Paint&Design Team #1
PersonDesignerPaintor2	P	Pt, Dsgn	Paint&Design Team #1
DesignTool1	T	Dsgn	Paint&Design Team #1
DesignTool2	T	Dsgn	Paint&Design Team #1
PaintTool1	T	Pt	Paint&Design Team #1
PaintTool2	T	Pt	Paint&Design Team #1
Machine1	M	Bld	Build Team #1
PersonBuild1	P	Bld	Build Team #1
PersonBuild2	P	Bld	Build Team #1
PersonBuild3	P	Bld	Build Team #1
PersonBuild4	P	Bld	Build Team #1
PersonBuild5	P	Bld	Build Team #1
BuildTool1	T	Bld	Build Team #1
BuildTool2	T	Bld	Build Team #1
BuildTool3	T	Bld	Build Team #1
BuildTool4	T	Bld	Build Team #1
BuildTool5	T	Bld	Build Team #1
PersonTiresAssembly1	P	TirA	Assembly Team #1
PersonTiresAssembly2	P	TirA	Assembly Team #1
ToolTiresAssembly1	T	TirA	Assembly Team #1
ToolTiresAssembly2	T	TirA	Assembly Team #1
PersonBrakesAssembly1	P	BrkA	Assembly Team #1
PersonBrakesAssembly2	P	BrkA	Assembly Team #1
ToolBrakesAssembly1	T	BrkA	Assembly Team #1
ToolBrakesAssembly2	T	BrkA	Assembly Team #1
ToolBrakesAssembly3	T	BrkA	Assembly Team #1
ToolBrakesAssembly4	T	BrkA	Assembly Team #1
PersonEngineAssembly1	P	EngA	Assembly Team #1
PersonEngineAssembly2	P	EngA	Assembly Team #1
PersonEngineAssembly3	P	EngA	Assembly Team #1
ToolEngineAssembly1	T	EngA	Assembly Team #1
ToolEngineAssembly2	T	EngA	Assembly Team #1
ToolEngineAssembly3	T	EngA	Assembly Team #1
PersonWiring1	P	Wrng	Wiring Team #1
ToolWiring1	T	Wrng	Wiring Team #1
ToolWiring2	T	Wrng	Wiring Team #1

Table C.3: Simple Scenario - Resource Details

Resource Type	Capabilities	Available Resource Number
Person	Pt, Dsgn	2
Tool	Pt	2
Tool	Dsgn	2
Machine	Bld	1
Person	Bld	5
Tool	Bld	5
Person	TirA	2
Tool	TirA	2
Person	BrkA	2
Tool	BrkA	4
Person	EngA	3
Tool	EngA	3
Person	Wrng	1
Tool	Wrng	2

Table C.4: Simple Scenario - Resource Details Summary

Appendix D

User Input - Survey Questions and Results

D.1 Survey Questions

The following questions represent the user input survey that was conducted.

Question 1 - What is your educational background?

- Computer Science
- Engineering
- Mathematics
- Other

Question 2 - What is your graduate degree?

- Undergraduate
- Bachelors
- Masters
- PhD
- Other

Question 3 - How would you describe your level of knowledge of manufacturing concepts?

- Very High
- High
- Average
- Below Average

Question 4 - Do you work with customers?

- Yes
- No (move to question 6)

Question 5 - With how many customers do you estimate you have worked with?

- 1-10
- 10-50
- 50-100
- More then 100

Please state your level of agreement with the following sentences.
Question 6 - This project aids in planning overview.

- Strongly Agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly Disagree

Question 7 - This project aids in capacity planning and problems detection.

- Strongly Agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly Disagree

Question 8 - This project's approaches could be used for other types of planning such as material planning.

- Strongly Agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly Disagree

Question 9 - This project's visualization concepts could be used within Dynamics AX.

- Strongly Agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly Disagree

Question 10 - This project represents a step forward in visualization within manufacturing, compared to currently available tools.

- Strongly Agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly Disagree

Question 11 - This project's features would appeal to customers that perform production planning.

- Strongly Agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly Disagree

D.2 Survey Results

Question 1 - What was your educational background?

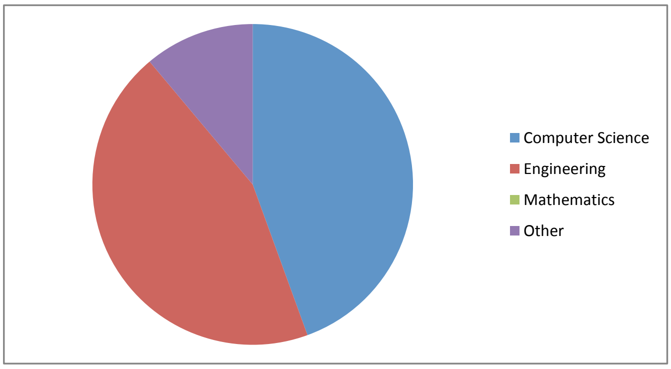


Figure D.1: Question 1 - Answers

Question 2 - What is your graduate degree?

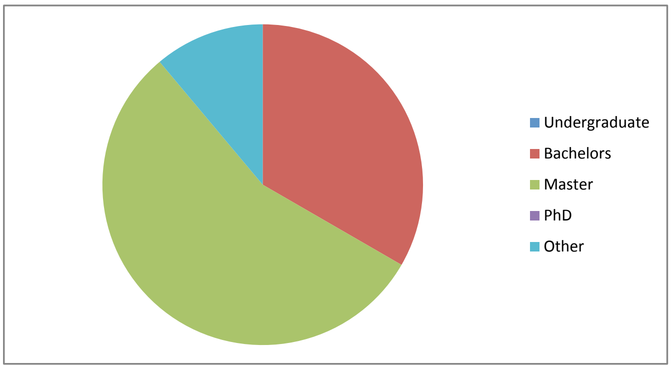


Figure D.2: Question 2 - Answers

Question 3 - How would you describe your level of knowledge of manufacturing concepts?

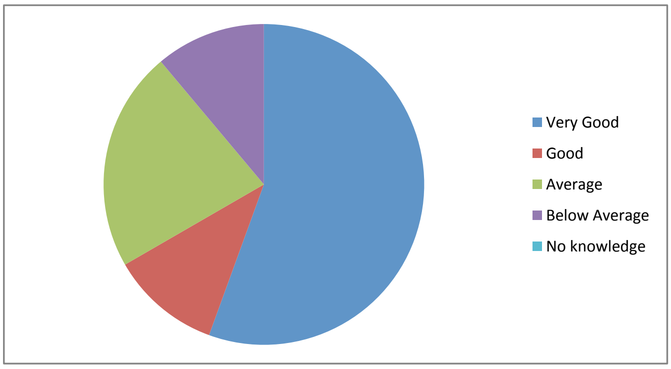


Figure D.3: Question 3 - Answers

Question 4 - Do you work with customers?

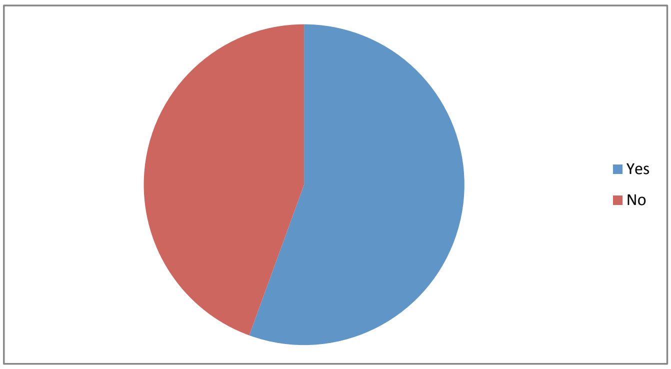


Figure D.4: Question 4 - Answers

Question 5 - With how many customers do you estimate you have worked with?

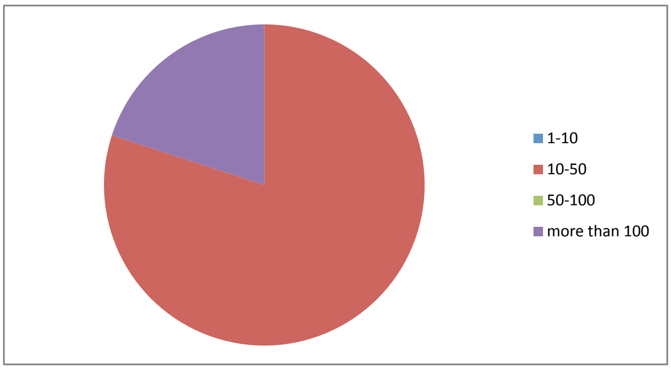


Figure D.5: Question 5 - Answers

Question 6 - This project aids in planning overview.

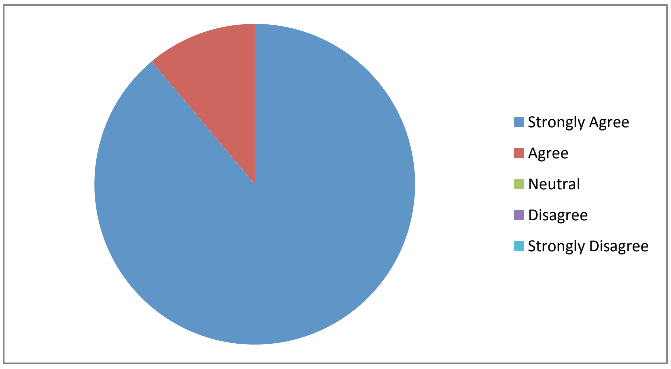


Figure D.6: Question 6 - Answers

Question 7 - This project aids in capacity planning and problems detection.

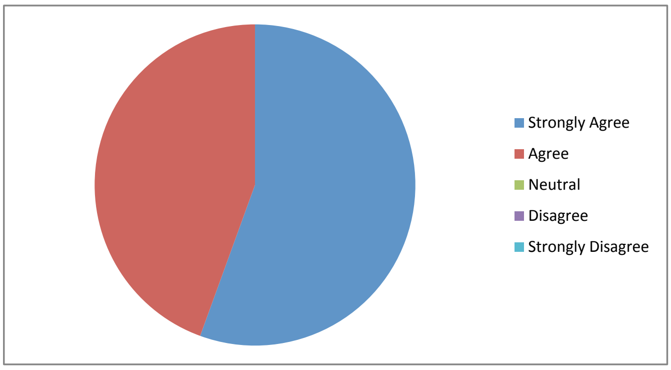


Figure D.7: Question 7 - Answers

Question 8 - This project's approaches could be used for other types of planning such as material planning.

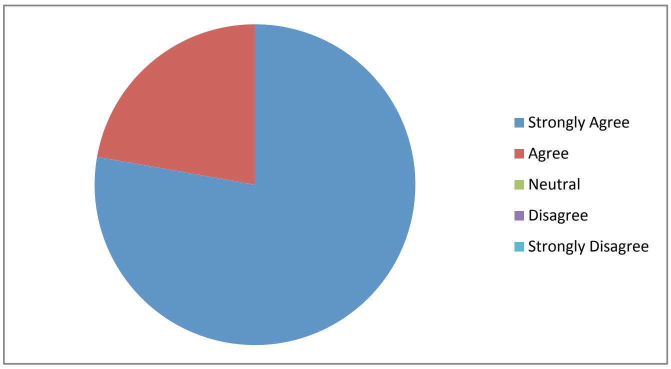


Figure D.8: Question 8 - Answers

Question 9 - This project's visualization concepts could be used within Dynamics AX.

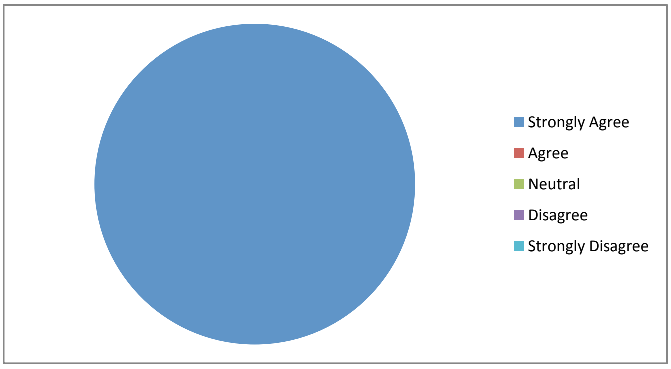


Figure D.9: Question 9 - Answers

Question 10 - This project represents a step forward in visualization within manufacturing, compared to currently available tools.

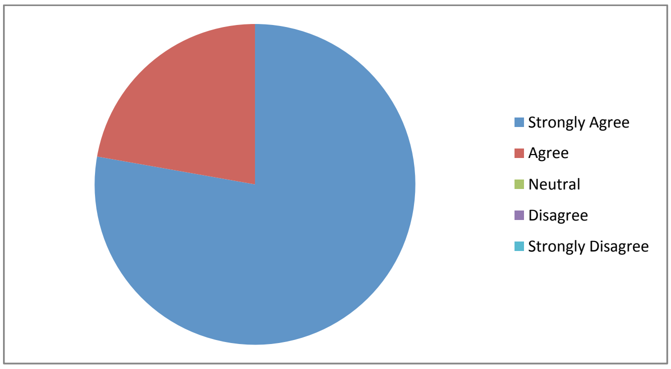


Figure D.10: Question 10 - Answers

Question 11 - This project's features would appeal to customers that perform production planning.

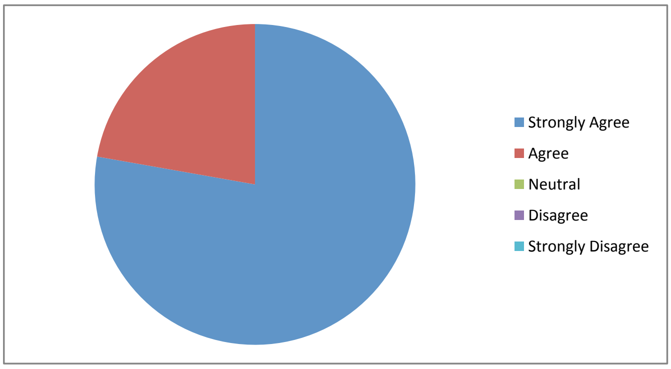
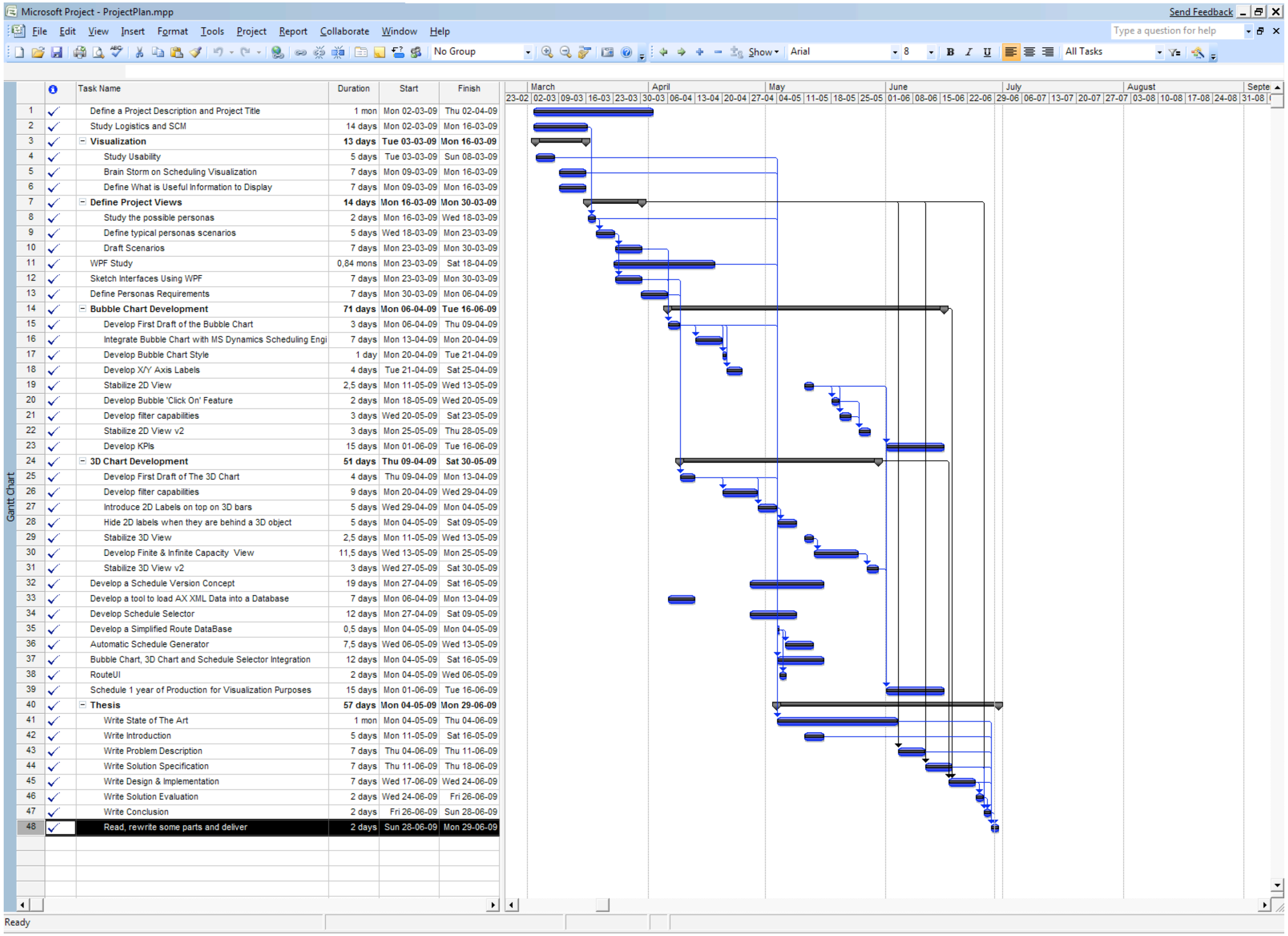


Figure D.11: Question 11 - Answers

Appendix E

Project Schedule



	ℹ	Task Name	Duration	Start	Finish
1	✓	Define a Project Description and Project Title	1 mon	Mon 02-03-09	Thu 02-04-09
2	✓	Study Logistics and SCM	14 days	Mon 02-03-09	Mon 16-03-09
3	✓	− **Visualization**	**13 days**	**Tue 03-03-09**	**Mon 16-03-09**
4	✓	Study Usability	5 days	Tue 03-03-09	Sun 08-03-09
5	✓	Brain Storm on Scheduling Visualization	7 days	Mon 09-03-09	Mon 16-03-09
6	✓	Define What is Useful Information to Display	7 days	Mon 09-03-09	Mon 16-03-09
7	✓	− **Define Project Views**	**14 days**	**Mon 16-03-09**	**Mon 30-03-09**
8	✓	Study the possible personas	2 days	Mon 16-03-09	Wed 18-03-09
9	✓	Define typical personas scenarios	5 days	Wed 18-03-09	Mon 23-03-09
10	✓	Draft Scenarios	7 days	Mon 23-03-09	Mon 30-03-09
11	✓	WPF Study	0,84 mons	Mon 23-03-09	Sat 18-04-09
12	✓	Sketch Interfaces Using WPF	7 days	Mon 23-03-09	Mon 30-03-09
13	✓	Define Personas Requirements	7 days	Mon 30-03-09	Mon 06-04-09
14	✓	− **Bubble Chart Development**	**71 days**	**Mon 06-04-09**	**Tue 16-06-09**
15	✓	Develop First Draft of the Bubble Chart	3 days	Mon 06-04-09	Thu 09-04-09
16	✓	Integrate Bubble Chart with MS Dynamics Scheduling Engi	7 days	Mon 13-04-09	Mon 20-04-09
17	✓	Develop Bubble Chart Style	1 day	Mon 20-04-09	Tue 21-04-09
18	✓	Develop X/Y Axis Labels	4 days	Tue 21-04-09	Sat 25-04-09
19	✓	Stabilize 2D View	2,5 days	Mon 11-05-09	Wed 13-05-09
20	✓	Develop Bubble 'Click On' Feature	2 days	Mon 18-05-09	Wed 20-05-09
21	✓	Develop filter capabilities	3 days	Wed 20-05-09	Sat 23-05-09
22	✓	Stabilize 2D View v2	3 days	Mon 25-05-09	Thu 28-05-09
23	✓	Develop KPIs	15 days	Mon 01-06-09	Tue 16-06-09
24	✓	− **3D Chart Development**	**51 days**	**Thu 09-04-09**	**Sat 30-05-09**
25	✓	Develop First Draft of The 3D Chart	4 days	Thu 09-04-09	Mon 13-04-09
26	✓	Develop filter capabilities	9 days	Mon 20-04-09	Wed 29-04-09
27	✓	Introduce 2D Labels on top on 3D bars	5 days	Wed 29-04-09	Mon 04-05-09
28	✓	Hide 2D labels when they are behind a 3D object	5 days	Mon 04-05-09	Sat 09-05-09
29	✓	Stabilize 3D View	2,5 days	Mon 11-05-09	Wed 13-05-09
30	✓	Develop Finite & Infinite Capacity View	11,5 days	Wed 13-05-09	Mon 25-05-09
31	✓	Stabilize 3D View v2	3 days	Wed 27-05-09	Sat 30-05-09
32	✓	Develop a Schedule Version Concept	19 days	Mon 27-04-09	Sat 16-05-09
33	✓	Develop a tool to load AX XML Data into a Database	7 days	Mon 06-04-09	Mon 13-04-09
34	✓	Develop Schedule Selector	12 days	Mon 27-04-09	Sat 09-05-09
35	✓	Develop a Simplified Route DataBase	0,5 days	Mon 04-05-09	Mon 04-05-09
36	✓	Automatic Schedule Generator	7,5 days	Wed 06-05-09	Wed 13-05-09
37	✓	Bubble Chart, 3D Chart and Schedule Selector Integration	12 days	Mon 04-05-09	Sat 16-05-09
38	✓	RouteUI	2 days	Mon 04-05-09	Wed 06-05-09
39	✓	Schedule 1 year of Production for Visualization Purposes	15 days	Mon 01-06-09	Tue 16-06-09
40	✓	− **Thesis**	**57 days**	**Mon 04-05-09**	**Mon 29-06-09**
41	✓	Write State of The Art	1 mon	Mon 04-05-09	Thu 04-06-09
42	✓	Write Introduction	5 days	Mon 11-05-09	Sat 16-05-09
43	✓	Write Problem Description	7 days	Thu 04-06-09	Thu 11-06-09
44	✓	Write Solution Specification	7 days	Thu 11-06-09	Thu 18-06-09
45	✓	Write Design & Implementation	7 days	Wed 17-06-09	Wed 24-06-09
46	✓	Write Solution Evaluation	2 days	Wed 24-06-09	Fri 26-06-09
47	✓	Write Conclusion	2 days	Fri 26-06-09	Sun 28-06-09
48	✓	Read, rewrite some parts and deliver	2 days	Sun 28-06-09	Mon 29-06-09

Figure E.1: Project Schedule