

Design of a fruit juice blending and packaging plant

Fey, J.J.H.

DOI:
[10.6100/IR540249](https://doi.org/10.6100/IR540249)

Published: 01/01/2000

Document Version

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Design of a Fruit Juice Blending and Packaging Plant

PROEFONTWERP

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
Rector Magnificus, prof.dr. M. Rem, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen op
dinsdag 7 november 2000 om 16.00 uur

door

Jeroen Fey

geboren te Maastricht, Nederland

Dit proefontwerp is goedgekeurd door de promotoren:

prof.dr.ir. J.E. Rooda

en

prof.dr.ir. A.A.H. Drinkenburg

Copromotor:

dr.ir. D.A. van Beek

Print: Universiteitsdrukkerij, Eindhoven.

[/stan ackermans institute, center for technological design](#)



The work in this thesis has been carried out under the auspices of the research school IPA (Institute for Programming research and Algorithms).

© Copyright 2000 J.J.H. Fey

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission from the copyright owner.

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Fey, Jeroen J.H.

Design of a Fruit Juice Blending and Packaging Plant / by Jeroen J.H. Fey.
Eindhoven: Technische Universiteit Eindhoven, 2000.

Proefschrift. - ISBN 90-386-2981-8

NUGI 831

Subject headings: fruit juice production / food processing industry
design method / group technology / scheduling

Vooraf

In maart 1997 rondde ik aan de TU in Eindhoven de ontwerpersopleiding Computational Mechanics af. Ik was inmiddels in dienst getreden bij Riedel. Jos Hannen, Koos Rooda en ik besloten toen om te starten met dit proefontwerp. Vier dagen per week van Eindhoven naar Ede en terug, ruim 125 duizend kilometer in auto en trein, maar geen moment heb ik spijt gehad. Terugkijkend op een periode van ruim drie jaar zijn er veel mensen die ik wil bedanken. Mensen die mij hebben gesteund met advies, discussies en gezelligheid, mensen van wie ik veel heb geleerd, waar ik intensief mee heb samengewerkt, koffie gedronken en gereisd, en mensen die veel tijd hebben gestoken in het reviewen van dit proefontwerp.

Bij Riedel wil ik bedanken: Jos Hannen voor het mogelijk maken van dit proefontwerp en voor de vrijheid die ik daarbij heb gekregen; Dirk van Wijk voor de tijd die hij heeft gestoken in het reviewen van dit proefontwerp, maar vooral voor zijn implementatie adviezen, operationele sturing en de soms noodzakelijke duw in de rug; Johan Janssen van Doorn en Jakob Strating voor de uitstekende samenwerking tijdens het herstructureringsproject; mijn collega's van het Bedrijfbureau voor de gezelligheid en hun collegialiteit; Arjan van Asselt voor saptchnologische adviezen en onderdak na de gezellige Wageningse avonden; Marc Luijsterburg en Pieter de Klerk voor de vele uren die we samen koffiedrinkend in auto's hebben doorgebracht tussen Eindhoven en Ede.

Bij de TUE wil ik Koos Rooda bedanken voor zijn aanmoedigingen, adviezen en vertrouwen; Bert van Beek voor de nuttige discussies en zijn review werk; Richard Peters en Nasko Hijdra voor hun bijdrage aan het ontwerp van de scheduling modules; Mieke Lousberg voor haar hulp en belangstelling gedurende de 6 jaar die ik betrokken ben geweest bij de sectie Systems Engineering; en Edgar van Campen, voor zijn grote bijdrage aan dit werk door onze vrijdagmiddagdiscussies en als co-auteur van hoofdstuk 3.

En tenslotte bedank ik Edith, met wie ik al meer dan 10 jaar samen ben en zonder wie ik deze mijlpaal nooit zou hebben bereikt.

Summary

In the last five years, the consumer market for fruit juices and fruit beverages has experienced increasing competition. The industry has to supply higher quality, larger variety, and better packaging at more competitive prices. From a manufacturing point of view, very high service levels have to be combined with short product lead times. Together with the strongly increased product variety this has set new performance standards for the manufacturing of canned fruit juice beverages.

Although a large variety of detailed design methods for a specific subset of design problems is available, literature does not present a universally accepted method for designing a fruit juice blending and packaging plant. Furthermore, literature disregards the fact that an industrial system evolves throughout its life-cycle and that optimizing and redesigning are essential tasks in the design process. Finally, although design of control and manufacturing system show a great deal of interaction, literature treats them separately. Integrating design of control and manufacturing system in one method can enhance the performance of the industrial system.

The objective of this dissertation is to provide a design method that structures the design process of fruit juice blending and packaging plants. This method is composed of a structure and a set of tools. The structure tells the user which design decisions need to be made and when to make them. The tools support the decision making process. The structure is generally applicable; the set of tools makes the design method specifically suitable for the design of fruit juice blending and packaging plants. The proposed design method incorporates design of the manufacturing system as well as design of the control system, where it affects the manufacturing system. It supports initial design as well as redesign of the plant operations.

The design method consists of four phases: objective definition, design of architecture, design of resources, and design of operations. First, the objectives and constraints are defined. Then, the architecture of the manufacturing and control system is designed. A first order approximation of the necessary resources is determined in the design of resources. Detailed design of the manufacturing system and control system is done in the design of operations.

The design method has been used to support a redesign project for Riedel's production facility. The production is divided in three successive stages: preparing, pasteurizing, and packaging. For juice preparation, a group technology structure is applied; for pasteurizing and packaging, a flow line structure is applied. In the design of resources, these structures are quantified. To support this process, two design tools have been developed. The result of the design of resources is a list of resources and a factory layout. In the design of operations, the resource design is refined and the control system is designed. To support design activities in this phase, two scheduling algorithms have been developed, and integrated in a design tool. This tool supports evaluation of 'what if' scenarios and the construction of detailed production schedules.

Contents

Summary	iii
1 Introduction	1
2 Production	5
2.1 Fruit juice industry	5
2.2 Technology	6
2.3 Riedel	9
2.4 Conclusion	13
3 Strategy	15
3.1 Structure	16
3.2 Objectives and constraints	19
3.3 Activities	21
3.4 Conclusion	24
4 Architecture	27
4.1 Functions	27
4.2 Processes	29
4.3 Conclusion	38

5 Resources	41
5.1 Pasteurizing and packaging	42
5.2 Preparation	56
5.3 Layout	62
5.4 Conclusion	64
6 Operations	65
6.1 Pasteurizing and packaging scheduling	66
6.2 Preparation scheduling	78
6.3 Refining the design of resources	85
6.4 Control system	88
6.5 Conclusion	91
7 Conclusions	93
Epilogue	97
Bibliography	101
Samenvatting	107
Curriculum Vitae	109

Chapter 1

Introduction

Commercial fruit juice production started in 1869 when Welch began bottling grape juice at Vineland, New Jersey. Welch had introduced the principles of heat sterilization, which, for the first time, enabled long preservation of fruit juices. Until the late 1920s, despite the revolutionary work of Welch, consumption of fruit juices was confined primarily to fresh juices. During the great depression at the end of this period, fruit and vegetables became popular home canning items for the American consumer. Consumers began to rely upon fruit juices as a source of Vitamin C. In the 1930s commercial fruit juice production developed rapidly in the U.S.A. due to major technological developments: flash pasteurization was perfected and applied commercially. This method of preservation was found to be superior to previous methods: the true fruit flavor could largely be retained and the method was relatively simple. The next technological breakthrough was established during World War II, fruit juice concentrates were prepared for the armed forces by evaporating the water under vacuum. Frozen concentrate was first introduced in the domestic marketplace in 1945-46. Commercial production of fruit juice beverages matured after World War II. It has often been stated that necessity is the mother of invention. In few instances is this cliché more applicable than in the juice processing industry. Social and technological developments during the great depression and World War II have boosted the fruit juice industry, and have enabled world-wide expansion of commercial fruit juice production¹.

In the beginning, the product variety was small: orange, apple, grape and grapefruit juice were the most popular fruit juice products, and packaging was mainly restricted to glass bottles. Expansion of demand was stimulated by the increasing popularity of tropical fruits in the 1980s. Development of convenient aseptic packaging systems has further accelerated availability and popularity of fruit juice beverages.

¹This paragraph is based on [Bro93]

Saturation of the consumer market of fruit juices in the 1990s, has forced companies to penetrate other market segments. Traditionally, the markets of fruit juices, soft drinks, and dairy products were clearly separated, but recent trends show that these markets are merging. As a result, the product range has diversified very rapidly in the number of recipes as well as the number of packaging types. The consumer focuses on health, freshness, no additives, less overprocessing and less overpackaging. Due to merging markets, the number of players in each market has strongly increased and the competition is tougher than ever. Together with the growing strength of retailers and supermarkets, this has set new standards for the fruit juice industry. The industry must supply higher quality, larger variety and better packaging at even more competitive prices [Fre95].

To be successful in today's highly developed consumer market of fruit juice beverages, a company is required to have a production system that supports a short time-to-market for high quality products. In order to realize a short time-to-market, a fruit juice blending and packaging plant needs to be flexible. However, as financial margins per product decrease, it also needs to produce cost efficient. In general, flexibility is expensive, and therefore, opposes cost efficiency. Excellent performance on both aspects is the ultimate challenge for every producer of premium quality fruit juice beverages. The cost efficiency and flexibility of a production system is largely determined in the design phase. Therefore, a high quality production system design provides a company competitive strength.

Until 1980 the variety of fruit juice beverages was small and fruit juice blending and packaging plants had a simple structure. Such plants consisted of a few receiver tanks, pasteurizers, filling machines, traypackers, and palletizers, all placed in-line, and some additional utilities. Order handling and production control was performed by hand on a few pieces of paper. The design of these production plants was done intuitively by process technologists and technicians.

Nowadays, a fruit juice blending and packaging plant contains dozens of tanks, blenders, filters, pasteurizers, packing lines, etc. Advanced production control tools have been introduced and administrative processes have been automated. A modern high quality juice plant costs several hundred million US dollars. Building such a complex and expensive plant based on nothing more than intuitive decision-making is hazardous. The design process needs to be structured to improve the quality of the production system design.

Scope of the thesis

By definition, every industrial system is a collection of products and a production system. A production system is further decomposed into four subsystems: the manufacturing system, the control system, the economical system, and the organizational system (see Figure 1.1, inspired by [Bra93]).

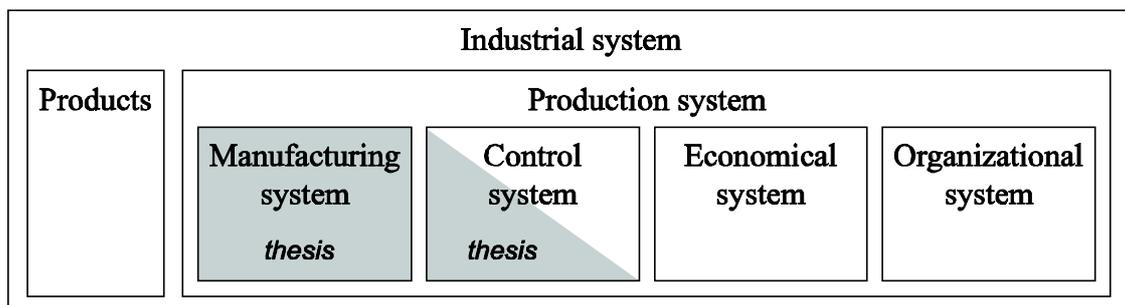


Figure 1.1: Decomposition of an industrial system and scope of this thesis.

The manufacturing system involves the transformation of materials into products, and is also referred to as the primary system. The control system, or secondary system, involves the information flow. The economical system, or tertiary system, involves the flow of values, compensating the flow of materials. The organizational system, or quaternary system, involves the people in the production system.

This thesis presents a method to support the design process of a fruit juice blending and packaging plant. The method focusses on the manufacturing system and on the control system, where it affects the design of the manufacturing system. Riedel's production system serves as a case study throughout this thesis.

The design method is composed of a structure and a set of tools. The structure tells the user which design decisions need to be made and when to make them. The tools support the decision making process. The structure presented in this thesis is generally applicable; Van Campen [Cam01], for instance, has successfully applied basically the same structure for the design of a wafer fabrication facility. The set of tools makes the design method more specific. In this case, the method is applicable for the design of all fruit juice blending and packaging plants.

Thesis outline

In Chapter 2 the development of the fruit juice industry and the production of fruit juice beverages are described. Furthermore, the company Riedel is introduced. Chapter 3 presents the structure of the design strategy. An overview of design literature is given and the structure of the design method is introduced. The proposed four-phase design method comprises the definition of objectives and constraints, and the design of architecture, resources, and operations. In the second part of Chapter 3 the objectives and constraints concerning Riedel's production system are discussed.

In the Chapters 4, 5, and 6 the latter three design phases are described. Chapter 4 describes the design of architecture. Chapter 5 describes the design of resources and Chapter 6 the design of operations. For each of these design phases the necessary tools have been developed. Chapter 7 presents the conclusions.

Riedel uses the proposed design method to redesign its production system. Besides modifications in the manufacturing and control system, the design has triggered large organizational changes. The implementation has been started in the summer of 2000 and will be completed in 2002. The epilogue briefly describes this redesign project.

Chapter 2

Production

This chapter discusses the main aspects involved in the production of canned fruit juices. In the second part of the chapter Riedel Drinks is introduced, and the Riedel juice plant is described.

2.1 Fruit juice industry

Fruit juice industry matured after World War II, as a result of two major technological breakthroughs: flash pasteurizing and evaporating water under vacuum. Flash pasteurization has established preservation of fruit juices while maintaining the flavor of the fruit. Evaporating under vacuum has enabled the use of high quality frozen concentrates in fruit juice production.

Nowadays, the majority of the fruit juices is made from frozen fruit juice and essence concentrates. There are several advantages associated with the use of frozen concentrates. Fruits are perishable and their production is seasonal, technologies for processing fruits into frozen concentrates have led to year-round availability of fruit juices for the food processing industry. The concentrated products are more stable and their quality is better maintained [Che93]. Furthermore, concentration of fruit juices has caused considerable savings in transportation and storage costs. In the early 1970s aseptic processing was commercialized on a large scale basis, this technology improved production, distribution and handling efficiency.

Frozen Concentrated Orange Juice (FCOJ) is the largest juice product in international trade. In recent years, Brazil has accounted for about 50% of the oranges processed in the world. Most of Brazil's oranges are made into frozen concentrate and shipped in bulk to the United States, Europe, and other developed parts of the

world. Distribution of FCOJ to production sites all around the world is done via import-export terminals in major ports. Other types of fruit juice traded in large amounts are: apple, grape, and grapefruit juice. Tropical fruit juices are exported from a large number of countries in Latin America, Africa, and Asia. The majority of these juices are distributed in drums as frozen concentrates or purees.

Three types of fruit juice beverages can be distinguished: 100% pure juices, nectars, and fruit juice drinks. Nectars are blends of fruit juices, sugar syrup, and citric acid; which should at least contain 50% pure juice. Fruit juice drinks usually contain at least 20% pure juice. An increasing part of fruit juice consumer products is ready-to-serve, it has replaced concentrated juice as the most popular product form.

2.2 Technology

The goal of a modern high quality juice plant is to reconstitute fruit juice in such a way that it approximates the color and flavor of fresh squeezed juice as close as possible. We can distinguish five factors which are essential for the quality of fruit juices: composition, microbiology, oxygen, temperature, and time. Because fruit juice is a low-acid food (pH 3.5-4), the majority of the harmful juice contaminants consists of molds, fermentative yeasts and acid tolerant bacteria [Fre95]. Minimizing microbial count is essential to increase shelf life and preserve the flavor of the fruit. The microbiology in acidic juices is adequately controlled by pasteurization [Dow95]. Oxidation degrades the color of the juice and the Vitamin C level. In order to eliminate oxidation, de-aeration of juice and juice concentrates is applied. Nevertheless, despite pasteurization, de-aeration, and good sanitation, chemical changes will occur on storage, which influence the shelf life and results in flavor deterioration. These changes occur more rapidly in concentrate, but can be minimized during storage at -18 degrees Celsius for as long as 5 years [Hen95]. At temperatures above 0 degrees Celsius fruit juices deteriorate rapidly: the growth of molds, yeasts, and bacteria accelerates. In order to guarantee a shelf life of several weeks, microbial count has to be minimized. During processing and packaging of juices the total of the five factors mentioned above has to be controlled very carefully in order to guarantee a high quality product.

The production of canned fruit juices and beverages can be subdivided in three production stages: preparation, pasteurizing, and packaging.

Preparation

Preparation is a recipe driven processing stage, which uses tanks, pipes, valves, manifolds, pumps, blenders, filters, discharging facilities, de-aerators, homogenizers and so on. The majority of the manufacturing resources used in this stage is multi-purpose. In general the production is batchwise. Preparation takes care of receiving ingredients, purchased in bulk, drums, sacks or boxes. Ingredients are dissolved, homogenized, blended, filtered and diluted, depending on the applied recipe and the quality constraints. When the product has been prepared, it is pumped to the pasteurizing and packaging equipment.

Pasteurization

Pasteurization technologies can be subdivided in thermal and non-thermal technologies. The first thermal pasteurization technology for fruit juice was developed around 1870 in the United States by Welch, he introduced the principles of heat sterilization. The first non-thermal pasteurization technology dates back to 1899. Ever since, thermal technologies have outclassed non-thermal technologies in applicability. Downes [Dow95] distinguishes three types of thermal pasteurization: in-pack pasteurization, batch pasteurization, and flash pasteurization.

In-pack pasteurization ensures product integrity, we can distinguish two variants: tunnel pasteurization and hot-fill. Tunnel pasteurizers transport packs through an enclosed tunnel while heating them by hot water sprays. The tunnel is often divided in three sections, in the first section the packs are slowly heated, in the second section the sprays are set at pasteurization temperature, and in the third section the packs are cooled down. When hot-fill is applied, the juice is pre-heated and filled into the pack at pasteurizing temperature. The pack is closed and cooled down after the pasteurizing period. A major drawback of in-pack pasteurization is that the heat load reaches levels at which the flavor of fruit juices strongly deteriorates. Furthermore, thermal pasteurization can only be applied to high temperature resistant containers, such as bottles or metal cans.

Batch pasteurizers pasteurize a batch of products in a large heating chamber. The system is thermally inefficient and can lead to damage to the juice due to prolonged heating [Dow95].

Flash pasteurization is carried out with a plate or tubular heat exchanger. The pasteurizer consists of four main parts: a generative section, a heating section, a holding section, and a cooling section. In the generative section cold incoming juice is preheated with exiting hot juice. In the heating section the juice is rapidly heated to

pasteurizing temperature with hot water or steam. In the holding section the juice is held at pasteurizing temperature. When the exit temperature of the pasteurized juice in the generative section is too high, a cooling section is needed. Flash pasteurization does not guarantee complete product integrity because micro-organisms can still enter at the packaging stage. Nevertheless, natural strength juices are best flash pasteurized [Hen95]. In order to minimize risks of micro-biological contamination, flash pasteurization is often combined with aseptic packaging equipment.

Recently, non-thermal technologies have gained renewed interest. Although modern state-of-the-art thermal pasteurizing technologies minimize the heat load, they still negatively influence the quality of fruit juices. Non-thermal pasteurizing technologies can play an important role in avoiding any heat load and further improvement of fruit juice quality. High pressure pasteurization and electroshock pasteurization are possible candidates for replacing thermal technologies in the future.

Packaging

The heart of every packaging system for fruit juices is the filling machine. Three fundamentally different filling concepts can be distinguished: hot filling, fresh cold filling, and aseptic filling systems.

Hot filling has been known as a pasteurizing and packaging method to achieve a shelf life of several months at ambient storage. As mentioned above, hot filling causes a strong heat load to fruit juices, which is unacceptable for a manufacturer of high quality fruit juice products. Hot filling is an option when the juice content in the beverage is low and the packaging is temperature resistant. Hot fill is often combined with a glass bottling line. Glass bottles are the oldest industrial packaging and are still very popular for packaging long life shelf stable products. Bottling lines can operate with many times the output of carton based lines. A disadvantage is the need for storing space for the bottles and complex logistics.

Fresh cold filling can be applied when a shelf life of several weeks is required. Infection of the finished product during packaging is not completely avoided. The distribution system needs to be refrigerated at 0-5 degrees Celsius. The need for refrigerated facilities, makes the system most suited for premium priced high quality products.

Aseptic filling enables packaging and storage at ambient temperatures, with a shelf life of several months. Distribution of aseptic products is therefore much more cost efficient than fresh cold filled products.

Besides the filling machine a packaging line contains downstream equipment such as conveyor systems, straw or cap applicators, handle applicators, tray packers, wrap

around machines, and palletizers, and in case hot fill is applied, a drying unit and a cooling tunnel.

2.3 Riedel

Riedel is a producer of ready-to-serve fruit juices, and fruit beverages in cartons. Riedel is a business unit of Friesland Coberco Dairy Foods, which is one of Europe's major companies in the dairy industry. Riedel has one production site which is located in Ede, The Netherlands. Its products are sold in The Netherlands and Belgium. Riedel produces only first class brands. Through its high quality products Riedel managed to become and remain a leader in the Dutch fruit juice market, in 1999 Riedel sold more than 330 million cartons of fruit juices and beverages. Despite the fact that competition in the fruit juice market is tougher than ever, Riedel has set itself the challenging task to increase market shares and sales volume every year. This goal has to be achieved in a cost efficient way in order to improve the economic competitiveness.

Riedel, as every other industrial system, is a collection of products and a production system. Below both subsystems are described.

Products

At the end of 1999, the product range of Riedel consisted of about 90 products or stock keeping units (SKU). These 90 products were composed out of 40 blends and 6 packaging varieties. In Figure 2.1 the sales volume of every product in 1999 is depicted. Figure 2.1 shows clearly that only a few products are produced in large quantities, the majority is produced in relatively small volumes. Financially, the same observations are made.

By far, the two most important products for Riedel are 1 liter cartons *Appelsientje* and *Goudappeltje*. Together they make more than 30% of the sales volume. Products with this kind of sales potential are not introduced every year. The majority of new products settle in the tail or the center of Figure 2.1. Nevertheless, Riedel strives to increase its market share by introducing a substantial number of new products every year.

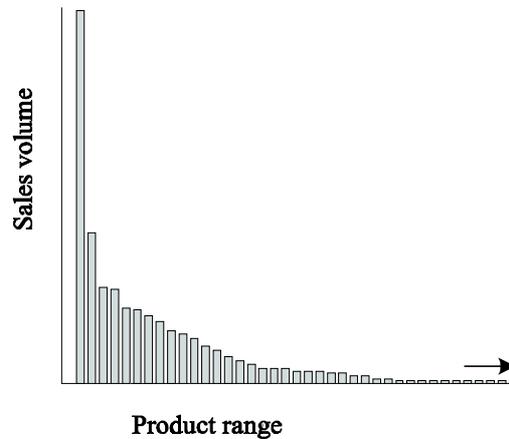


Figure 2.1: Sales volume per product.

Production system

Riedel's production system, is composed of four subsystems: a manufacturing system, a control system, an economical system, and an organizational system. This thesis focusses on the first two subsystems. In the following, Riedel's manufacturing, and control system are described briefly.

Manufacturing system. Production at the Riedel juice plant takes place five days per week, in three shifts. The production of juices is divided into three successive stages. As each stage is incorporated in a separate department, we can distinguish three production departments: the Preparation Department, the Pasteurizing Department, and the Packaging Department. In the Preparation Department the juice is prepared batchwise according to a specific recipe. When the preparation has been finished, the juice is pumped to the Pasteurizing Department, and subsequently to the Packaging Department, where it is packed in cartons of various appearances. The cartons are piled up on a pallet and transported to the warehouse. A simplified overview of the plant is depicted in Figure 2.2.

The major part of juice ingredients is delivered in bulk, the remaining part is delivered in drums, small containers, and bags. The laboratory tests whether the ingredients meet the specified quality requirements. The ingredients that pass the tests are pumped into receiver tanks. Preparation starts as soon as all ingredients are present and one of these receiver tanks is available.

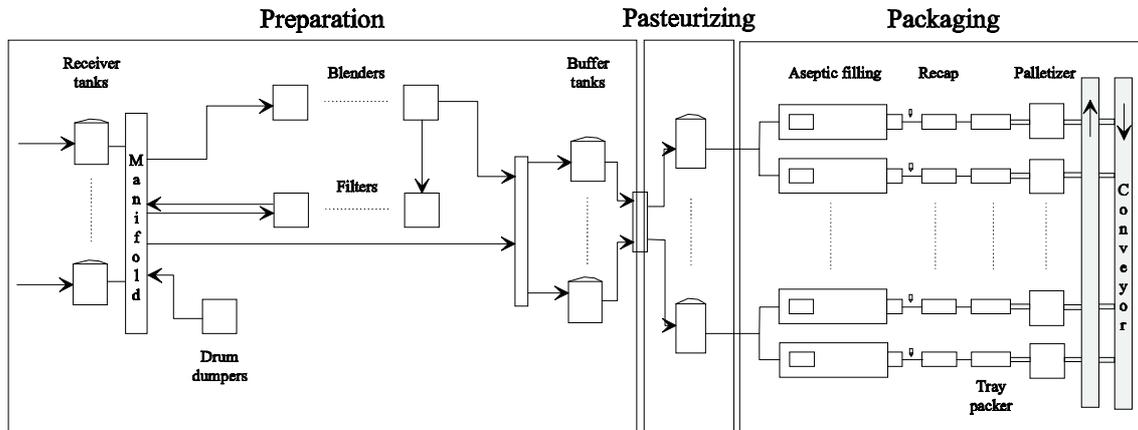


Figure 2.2: Simplified overview of the Riedel juice plant.

In general, two different ways of preparing can be applied. The first method is called preparation by in-tank dilution: first, all ingredients are pumped into one receiver tank, then the correct amount of water is added, finally the blend is stirred. The homogeneous natural strength juice is pumped to one or more pasteurizers and packed. In some cases the product is filtered. The second method is called preparation by in-line dilution by means of a blender: fruit concentrates are pumped into a tank and stirred. Next, the juice is pumped to a continuous blender to add water and in some cases sugar to the concentrate. The output of the blender is pumped into a buffer tank. In general the buffer tank is too small to buffer the complete batch, its main function is to damp the oscillatory output of the blender before the juice is pasteurized and packed. In the sequel, this method will be referred to as preparation by in-line dilution.

From a quality point of view, preparation by in-tank dilution is the most subtle method, and can be applied for all products. Preparation by in-line dilution cannot be applied for products that contain a large amount of floating fibres: these fibres would be crushed in the blender. Although preparation by in-line dilution is less subtle, it saves time and is advantageous when the size of a batch exceeds the size of a receiver tank.

The natural strength juice is fed to one or more pasteurizers; the Pasteurizing Department contains 12 pasteurizers. The objective of pasteurizing fruit juices is to minimize microbial count and to increase shelf life. Concisely speaking, these pasteurizers consist of a pipe and four tubular heat exchangers: two for heating, one for pasteurizing, and one for cooling the stream of juice. Depending on the type of possible harmful micro-organisms, the pasteurizing temperature is set. The pasteur-

izer produces a flow of aseptic/sterile juice which is pumped to one or two packaging lines.

The juice is packed using aseptic filling machines, cap or straw applicators, tray packers, and palletizers, in that sequence (see Figure 2.2). These machines are placed in a line. The Packaging Department contains 21 of these lines, which together produce 6 packaging variants. The cap or straw applicator glues a pouring cap or drinking straw, on the carton. The tray packer places a number of cartons in a tray, full trays are stacked up on a pallet with a palletizer. Pallets are transported to a machine, that wraps up the pallet with a plastic cover, and are subsequently transported to the warehouse. In order to guarantee satisfaction of high quality constraints, the products remain in the warehouse for a week, while being tested thoroughly.

Control system. We now consider the information flow that is related to order releasing, which is the process of the conversion of a client order into a production batch. A simplified overview of this information flow is depicted in Figure 2.3, several departments are left out for clarity reasons.

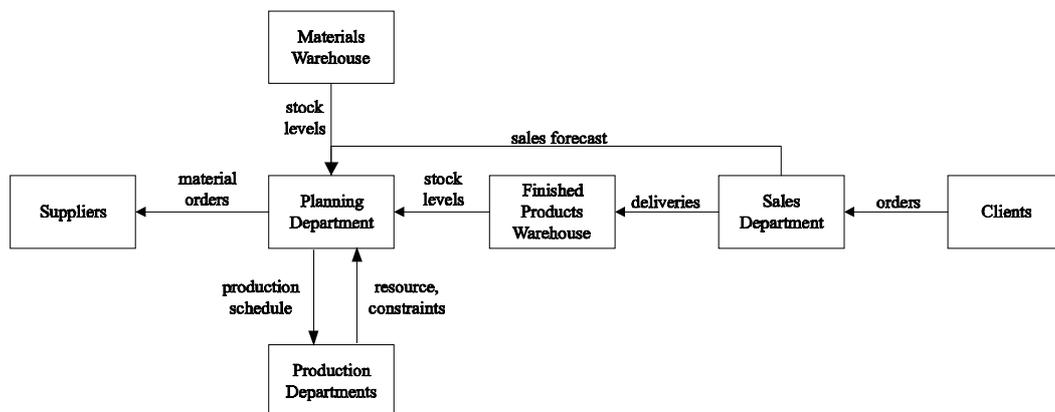


Figure 2.3: Information flow concerning order releasing at Riedel.

Clients contact the Sales Department and order products, subsequently deliveries are made within one or two days. Riedel applies a make-to-stock policy for nearly all products. Once a week, the Finished Product Warehouse hands the stock levels of every product to the Planning Department. At the same time, the Sales Department provides the latest sales forecasts. Subsequently, the Planning Department generates a Master Production Schedule (MPS), which is detailed for the next 6 weeks and coarse for the following 46 weeks. This MPS provides a tool to control the stock

levels and production resource requirements. When the MPS is available, a detailed production schedule for the packaging lines is generated. In this detailed production schedule all the constraints are taken into account. At the beginning of every week, the detailed production schedule is distributed as a Gantt chart. Twice a day, the production departments and the planning department meet to evaluate production progress; if necessary, the production schedule is modified.

2.4 Conclusion

The production of fruit juice beverages consists of three stages: preparing according to a recipe, pasteurizing to eliminate the majority of biological contaminants, and packaging. Technological developments in especially pasteurizing and packaging have enabled worldwide year-round consumption of high quality fruit juice beverages in convenient packaging variants.

Riedel is a producer of high quality ready-to-serve fruit juice beverages in aseptic cartons for the Dutch en Belgian consumer market. Riedel's product range contains 90 SKUs composed out of 40 blends and 6 packaging variants. Riedel strives to increase its market share by frequently introducing new and innovative products. Products are produced batchwise and visit the Preparation Department, Pasteurizing Department, and Packaging Department, in that order. Finally, products enter the warehouse before being transported to clients.

Chapter 3

Strategy

A high quality production system design is essential to be successful in today's highly developed consumer market for fruit juice beverages. Structuring the design process will lead to a better production system, and, as a consequence, improves competitive strength. In this chapter a strategy is presented for designing a modern fruit juice blending and packaging plant. Basically the same design strategy is also used by Van Campen [Cam01] to design a multi-process multi-product wafer fab. Although the industrial systems to be designed differ tremendously and the characteristics of both industries show almost no resemblance, the used design strategy is similar. Section 3.1 and Section 3.3 have been written in cooperation with Van Campen.

An industrial system is designed to serve certain objectives. With the emerging of new objectives or the changing of old ones, new industrial systems arise and old ones disappear. The life-cycle of an industrial system has been decomposed into five phases: orientation, specification, realization, utilization, and elimination, (see Figure 3.1).

The objectives of the industrial system are defined in the orientation phase. After this phase, one is aware of a certain need. The functions that have to be performed to satisfy this need are defined in the specification phase, also called the design phase. These functions are defined together with the required resources. The definitions are presented in an abstract system or model. In the realization phase the actual system is built and tested. The result is a working industrial system. In the utilization phase, the return on investments as specified by the exploiting company – made in the previous phases and to be made in the elimination phase – must take place. When the objectives have changed and the industrial system does not meet these objectives anymore, it has become obsolete. The last phase of the life-cycle concerns the elimination of the industrial system.

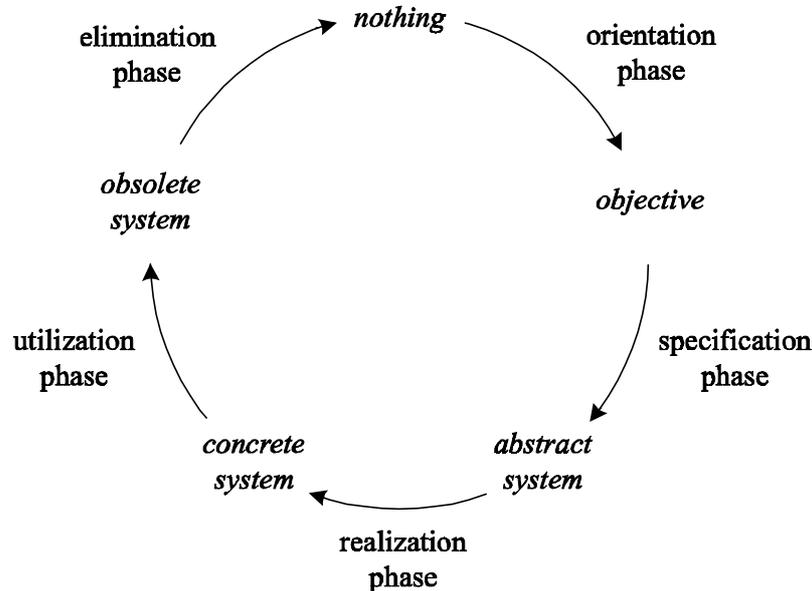


Figure 3.1: Life-cycle of industrial systems [Roo97].

In the remainder of this chapter the applied design strategy is discussed. First, a review of existing design methods is presented. The design strategy is deduced from these methods. The last part of this chapter discusses the design activities. Section 3.4 presents the conclusion.

3.1 Structure

Literature on designing industrial systems does not present a universally accepted methodology. It provides a large variety of detailed methods developed for a specific subset of industrial system designs. A large class of design methods focuses on one specific aspect of the relation between product and production system: Design for Assembly [Boo83], Design for Quality [Nic92], and Design for Recycling are a few of these 'Design for X'-methods. Another class approaches design problems from a production control perspective. The following methods can be named: load oriented manufacturing control [Wie95], group technology [Bur71], sociotechnics [Sit94], JIT [Shi81], lean production [Wom90], and OPT [Gol84].

The Business Process Reengineering approach (BPR) [Ham93] is another approach that can be used to redesign an industrial system. BPR advocates to make revolu-

tionary changes by focusing on the core business of the system. It is not a process that preaches continuous improvement. Therefore, it is typically a method in which different design methods are used to improve an existing system, rather than to design a new one.

The review above shows that much knowledge has been formalized in numerous design methods. Brandts [Bra93] opposes that the use of these methods is hindered by the fact that no structure exists in how to use the methods. A structured design method should point out what aspect to study at a certain moment and which design method should then be deployed. Brandts proposes a five phase structure of the industrial system design process: formalization of the objective definition, identification of basic sub-systems, phasing of the design processes of the various basic sub-systems, identification of the relevant attributes for every design phase, and selection or development of methods and techniques to support decision-making in the various design phases.

All design methods and structures mentioned above show a linear approach in that the design process always advances the process of realizing the designed system. This omits the fact that optimizing and redesigning are essential tasks, not only for a new design but also for an operational production facility. As the design process advances the realization process will be started. The level of detail in the design process increases as time proceeds. Design activities are not terminated when the specification phase is finished. On the contrary, the design process proceeds as long as the production facility is utilized.

The division of production systems in a manufacturing system and a manufacturing control system is often used to divide the design process in parts. Traditionally, both in research and in practice, production control has often been viewed in isolation from the manufacturing system design [Roy98]. Designing the control system then starts when the design of the manufacturing system has been completed. As the manufacturing system and the control system show large interaction, it is a difficult task to first completely design the manufacturing system and then the control system. The relation between the design processes of the manufacturing system and the control system is so strong, that they cannot be considered separately. Control issues have a direct impact on the manufacturing system design and vice versa. Johnston [Joh95] states that the effectiveness of control can be enhanced through reengineering of the manufacturing system.

A design strategy describes the sequence of activities that are carried out in the process of designing an industrial system. It provides a framework for intended activities. Within each activity design methods or design tools are used to obtain the desired result. We propose a four phase design strategy, see Figure 3.2.

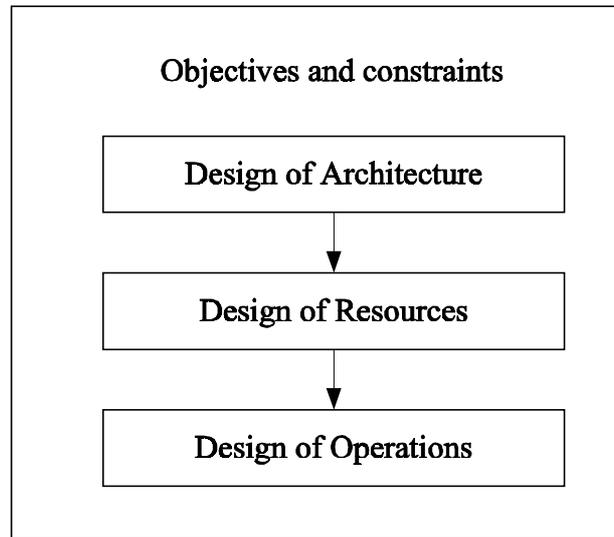


Figure 3.2: The four phase design strategy.

Each design starts with defining the objectives and constraints. This is to make clear what is expected from the design. Section 3.2 describes the objectives and constraints that concern Riedel Drinks. The next three phases represent the actual design activities, which are described in Section 3.3.

Conclusion

In this section a design method for the design of industrial systems has been proposed. This method consists of four phases: the objective definition, the design of architecture, the design of resources, and the design of operations. Development of the new design method is necessary for three reasons. First, literature provides a large amount of detailed design methods developed for specific problems rather than generally applicable methods for designing complete industrial systems. Second, literature disregards the fact that the majority of industrial systems evolve throughout its life-cycle, and that redesign is an essential task in the design process. Finally, although design of control systems and manufacturing systems show large interaction, literature treats them separately: design of the control system starts when design of the manufacturing system is finished. The proposed design method incorporates the design of the manufacturing system and the design of the control system where it affects the manufacturing system design.

3.2 Objectives and constraints

The success of a design can be measured by checking whether the resulting system satisfies all predefined objectives and constraints. Defining these objectives and constraints is the first phase in the process of designing an industrial system. However, objectives and constraints are not fixed, they remain subject of discussion during the design process. The design team and the company management need to evaluate objectives and constraints regularly.

Objectives

The industrial system to be designed needs to support a company in meeting business objectives. Business objectives need to be translated into design objectives. An important part of these design objectives needs to cover the companies marketing strategy. The design objectives need to reflect how the company wants to position its products in the market: an industrial system for low price, standardized products differs substantially from an industrial system for premium priced, high quality, custom-made products.

Constraints

An industrial system by definition is an open system that interacts with its environment, which consists of customers, suppliers, government, financiers, and labor market. This interaction generates constraints that need to be satisfied by the industrial system to be designed. Below, the business and design objectives and constraints of the Riedel juiced plant are formulated.

Riedel's objectives

Riedel is a producer of premium quality ready-to-serve fruit juices and fruit beverages for the Dutch and Belgian consumer market. Riedel is and wants to remain a leader in the Dutch fruit juice market. A diverse and innovative product range, exclusively composed of premium brands, and penetration of multiple market segments should guarantee continuity of business. Generating high profits is the main financial goal.

Riedel's production facility has to be tailored to the business objectives specified above. The production system needs to support a short product life cycle, and a great diversity of high quality products. As a consequence, Riedel's production

facility needs to be flexible, and excel in producing a large number of small batches, and a small number of large batches every week. In general, flexibility is not for free, and short production runs tend to decrease productivity. Nevertheless, the production system needs to enable high profits. Finding a global optimum between these opposing objectives is a major challenge in the design process.

Riedel's constraints

Riedel produces ready-to-serve fruit juice beverages for the Dutch and Belgium consumer market. For periods of one week and shorter, sales volume fluctuate strongly and are highly unpredictable. The majority of products are distributed to consumers via retailers. These retailers demand a delivery time of one day. In contrast with this, raw materials suppliers use a delivery time of one week. Riedel uses a large variety of perishable raw materials, which are delivered under non-aseptic conditions. In order to keep these raw materials in stock at Riedel's facility, special equipment needs to be installed, which is very expensive with respect to hardware investments as well as operational costs. Therefore, Riedel receives its raw materials Just In Time (JIT). A consequence of purchasing JIT is that batch sizes depend on the purchasing unit. As a result of all these constraints, Riedel has decided to produce on stock.

Distribution takes place at ambient temperatures, therefore, aseptic packaging is required to guarantee an acceptable shelf-life. A special range of premium quality products is distributed chilled, for these products non-aseptic packaging is an option.

Riedel's production facility is located near a residential area. To minimize noise for local residents, the government allows only a limited number of truck arrivals and departures. Consequently, trucks carrying raw materials or end products have to be loaded as full as possible.

Financiers are mainly interested in a high return on the capital invested. Riedel has a collective employment agreement with its personnel. In this agreement, working hours, salaries, and other employment conditions are determined. For 1999 Riedel and the labor union have agreed on a five day working week of three shifts, labor costs are about 30 US dollars per hour.

3.3 Activities

Architecture

Design of architecture is the second phase in the design process. Webster's collegiate dictionary defines *architecture* as: 'a unifying or coherent form or structure', or 'the manner in which the components of a computer or computer system are organized and integrated'. The term denotes the strive for coherent structures, something that the design process must achieve. In this thesis architecture is defined as the manner in which the components of a specific industrial system are organized and integrated.

It is important that a top-down approach is applied: the architecture of an industrial system has large influence on the specification of system components and their layout. Furthermore, a well designed architecture supports an industrial system in realizing manufacturing objectives. The need to excel in meeting objectives such as high flexibility, short lead times, high throughput, and high effectivity has to be reflected in the systems architecture. For example, a machine shop, where a large range of customer specific products are manufactured, is typically a production system where flexibility is essential. This is reflected in its architecture that is often focussed on the function of processes, which enables customer specific routing of products.

Design of architecture is divided into two levels of abstraction. It starts with designing the structure of manufacturing, or in other words, it starts with deciding how products are made. Technological developments play an important role at this step in the design process. Then the architecture for both the material flow as well as the control system are designed.

The first and most abstract step is concerned with the structure of manufacturing, the result of this step is a function model. The second step is concerned with the architecture of the material flow system and the control system. The result of this step is a process model. To illustrate the results of these two steps, a simplified function and process model for the production of canned orange juice are depicted below.



Figure 3.3: A function model for the production of canned orange juice.

Figure 3.3 depicts a function model for the production of canned orange juice. As described in Chapter 2, the production of canned juice starts with preparing the

blend, subsequently the juice is pasteurized, and finally the juice is packed in cartons. Now each of these functions can be represented by one or more processes. Figure 3.4 depicts a process model for the production of canned orange juice.

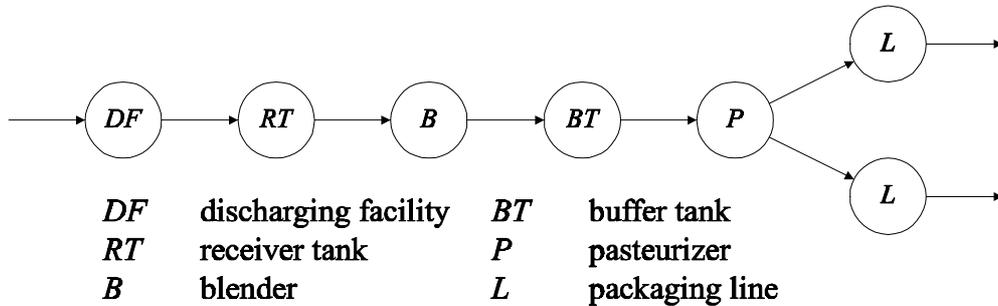


Figure 3.4: A process model for the production of canned orange juice.

The orange juice concentrate is discharged with discharging facility *DF* in receiver tank *RT*, and subsequently diluted with blender *B*, buffered in buffer tank *BT*, pasteurized with pasteurizer *P* and packed with packaging lines *L*.

The following step in the design process is to determine the necessary resources to perform the required functions, and satisfy the business objectives and constraints.

Resources

Design of resources is the third phase in the design process of an industrial system. As is captured in the name, design of resources deals with assets or means. In this thesis, design of resources focuses on quantifying the architecture designed in the previous design phase. This is necessary to concretize the design and obtain a good approximation of the necessary equipment and factory layout.

Where the business objectives have a large influence on the architectural design, the architectural design, on its turn, has a large impact on the number of resources and the factory layout. For example, a process oriented architecture in general may require less resources than a product oriented architecture. This is because the former has its resources grouped to functional operation, while the latter is structured and laid out towards the product flow, which may therefore lead to several machines in different places that are not fully utilized.

The resource design results in a list of resources and a factory layout. This result is accomplished in two successive steps: the first step is concerned with determination

of resources, the second step is concerned with laying out these resources on the shop floor. Often the available information which is needed as input for the determination of resources is based on expectations and not accurate. For example, sales volume can only be estimated and the product range is adapted rapidly to changing customer demands. In those cases, only a first order approximation of the necessary resources is possible at this stage of the design process.

A quantitative analysis of the system to be designed is made to come to a set of resources, e.g., machines or operators. The main tool for handling the capacity analysis problem is modeling. A model is an abstraction of reality, which stresses some aspects of reality while hiding irrelevant details. For example, consider a capacity analysis model that has the purpose of determining the necessary number of drilling machines in a machine shop. Relevant information which should be incorporated in the model is the capacity of one drilling machine, the number of available working hours and the total workload for the drilling section. Irrelevant aspects are for instance the size of the machines and its setup procedures. The art of modeling is to leave out as much information as possible without harming the relevance of the model. An important modeling activity when facing large and complex industrial systems is decomposition. The industrial system is decomposed into several smaller parts which are modeled and evaluated independently. A model consists of some sort of formal specification, e.g., a set of mathematical equations, or a discrete event specification. The capacity analysis problem is solved by solving the set of mathematical equations or executing simulation experiments and analyzing the results.

Resource design is critical to the success of a manufacturing facility, as it affects important performance measures such as lead time, equipment utilization, and system throughput. Once the layout has been determined, attention must be paid to how the control strategy is translated into operations. This is done in the design of operations phase.

Operations

Design of operations is the last activity of the design strategy. Webster defines operations as: ‘the way in which something works’. Design of operations is concerned with the design of the (detailed) way in which a plant works. The understanding of a plants operative performance and how it is influenced by strategic and operative decisions, provides companies a major advantage. For example, minimizing lead times, working at high equipment utilizations, and being able to adapt to changing product types, are crucial to ensure profitability of the plant.

In the resource design only a rough capacity analysis has been made, due to the absence of accurate and detailed information on the operation of the plant. During the process of designing a plant, more and more detailed and accurate information becomes available. This enables zooming in on the operation of the facility, and retrieving a more accurate image of its behavior. The objective of the operations design is twofold: validating the resource design, and supporting redesign of a utilized production system. The first objective deals with checking whether the resources calculated in the previous design phase provide sufficient capacity to operate the factory successfully. Compared with the resource design, more detailed information is taken into account. Furthermore, the dynamic interaction between processes is considered. The second objective is concerned with modifying the production system design in such a way that it is able to cope with changing market demands in a cost efficient way. The production facility or part of the production facility is modeled, analyzed, and optimized.

The outcome of the operations design are a valid capacity plan, control rules, scheduling rules, and operating procedures. Through the evaluation of models, a set of resources and rules can be determined. Rules describe the way in which a manufacturing facility is operated. Two examples are: where to store material on the shop floor when it is not in process and a set of scheduling rules. Determining rules can be referred to as optimizing. Implementing these rules leads to a better performing industrial system.

The process of specifying models, optimizing, and implementing rules does not take place just once, it is an iterative process that needs to be repeated often during the design and also during utilization of the industrial system. Figure 3.5 shows the circle of improvement.

The start of the cycle is the existing system or the system to be designed. By making a specification, knowledge and characteristics of the system are obtained and design decisions are made clear. The formal specification can be used as a model, for example as a discrete event simulation model, with which different scenario's are evaluated. In the optimization phase, the optimal set of rules is determined. Eventually, these rules are implemented, leading to a better performing system.

3.4 Conclusion

In this chapter a strategy for designing a fruit juice blending and packaging plant has been proposed. Design literature provides a large amount of detailed design methods developed for specific design problems rather than generally applicable methods for designing complete industrial systems. The proposed design strategy provides a

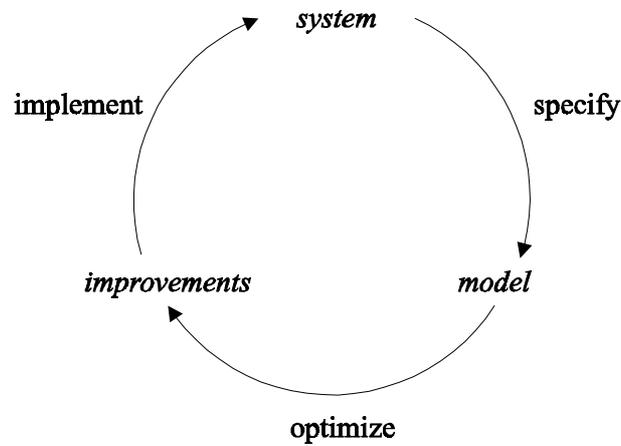


Figure 3.5: Circle of improvement.

structure that can be used throughout the specification and utilization phase of a fruit juice blending and packaging plant. It incorporates initial design as well as redesign of a utilized production facility. It consists of four phases: the definition of objectives and constraints, and the design of architecture, the design of resources, and the design of operations. Every design starts with identifying design objectives and constraints. In the design of architecture phase, the structure of the material flow system and the control system is designed. The design of resources is concerned with determining the required resources and constituting a floor plan. In the design of operations phase, detailed scheduling related variables are taken into account in order to validate the list of resources determined in the resource design.

Chapter 4

Architecture

In this chapter we discuss aspects involved with the second step in the design process of a fruit juice blending and packaging plant. The goal of this design step is to determine the structure of the manufacturing system. Technologies have a major impact on decisions taken at this stage of the design process. We consider state-of-the-art technologies which are commercially available at this moment, and mention implications of future developments, as far as they are within sight. In the production of canned fruit juices three basic functions can be distinguished: preparing, pasteurizing, and packaging. Section 4.1 is concerned with determining the sequence of these three functions, all feasible alternatives are discussed. By choosing one of these feasible sequences, the structure of manufacturing is determined. In Section 4.2 the structure of the manufacturing system is designed: every function is represented by one or a number of interacting processes. Unlike functions, processes have a sense of time, processes are more concrete than functions and can be used to model time behavior of industrial systems. In Section 4.2 a process model for Riedel's production facility is presented. Conclusions are presented in Section 4.3.

4.1 Functions

In the production of fruit juices we can clearly distinguish a flow line. There is relatively little back-flow and every production equipment in the line is visited once. In Chapter 2, we have seen that three production functions can be distinguished: preparing, pasteurizing, and packaging. The ordering of the three functions in the flow line has to be determined. In theory there are $3! = 6$ possible sequences, but not all of them are feasible. A sequence is not feasible when the function packaging

precedes the function preparing, as a consequence only three solutions remain (see Figure 4.1).

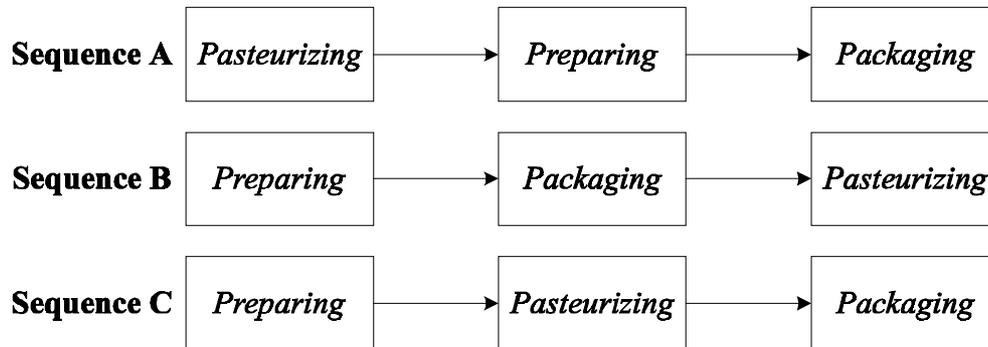


Figure 4.1: Feasible sequences of functions.

Sequence A. In this sequence, all ingredients that arrive at the juice plant are pasteurized first. The aseptic ingredients can be stored or used for production. An advantage of this alternative is that ingredients can be purchased in economic quantities and stored for a long time. Furthermore, the batch sizes are not dependent anymore on the size of a purchasing unit. When a product or an ingredient is pasteurized, every further contamination has to be prevented. In order to maintain sterile or very clean conditions throughout the whole production process, two options are available: the equipment downstream is aseptic, or the equipment downstream is very clean and chilled. Aseptic or very clean equipment is far more expensive than non-aseptic equipment, furthermore, the operational costs for both types of equipment are very high. Large investments and operational costs make this sequence unfit for economic production of packed fruit juice.

Sequence B. To minimize the need for aseptic or very clean equipment, we strive to pasteurize as late as possible. For this reason, alternative B is a very interesting one. At this moment there are a few technologies to pasteurize packed fruit juice, in Chapter 2 pasteurizing technologies are subdivided in two classes: thermal technologies, and non-thermal technologies. Thermal pasteurizing technologies applied to packed fruit juice cause a strong heat load to the product. From a quality point of view, technologies which do not use heat are better suited for pasteurizing packed fruit juice. Unfortunately, the processing costs are very high at present, and thus not suited for the cost efficient production of ready-to-serve juices [Fre95]. Sequence

B can be considered when the flavor of the fruit juice components in the beverage is not a main issue, and the packaging material is temperature resistant.

Sequence C. This sequence can be applied with common technologies: non-aseptic preparation equipment, flash pasteurizers, and depending on the required shelf life, fresh cold fillers or aseptic fillers. Fresh cold filled products have a shelf life of several weeks, the distribution system needs to be chilled. Aseptic filled products have a shelf life of several months, and can be distributed at ambient temperatures. Production and distribution costs, and marketing issues should be considered carefully before choosing one of the two concepts.

Conclusion

Sequence A is very costly, high investments as well as high operational costs make this sequence unfit for a juice plant. Sequence B is not applicable for the production of premium quality fruit juices. Thermal pasteurizing technologies for packed products cause an unacceptable heat load, which damages the flavor of the fruit. Non thermal pasteurizers are not commercially available. However, sequence B can be applied for the production of beverages with a small fruit juice content packed in high temperature resistant containers, such as glass bottles or metal cans. Sequence C is commonly applied for the production of milk and fruit beverages in cartons. The use of non-aseptic preparation equipment, flash pasteurizers, and fresh cold or aseptic fillers make this option a very cost efficient one.

Which sequence needs to be applied, is strongly dependent on the type of products that will be produced. As a producer of premium quality fruit juices in carton packages, sequence C is best fit for Riedel.

4.2 Processes

In the previous section, the sequence of functions is determined. Every function can be represented by one or a complex of interacting processes. In Figure 4.2 a model of an elementary juice plant is depicted.

The function preparing is represented by processes modeling a discharging facility (process *D*) and a receiver tank (process *RT*). The function pasteurizing is represented by process *P*, modeling a pasteurizer, the function packaging is represented by process *L*, modeling a packaging line. Utilities such as a de-aerator, cleaning equipment, and a water installation are not considered. The modeled production facility

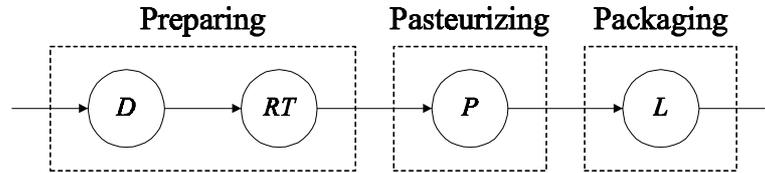


Figure 4.2: Iconic process model of an elementary juice plant.

consists of one production line capable of producing one blend at a time, and simultaneously package this blend in one type of package. Discharging facility D empties trucks carrying bulk. The ingredients are pumped to receiver tank RT , subsequently water is added in order to retrieve a natural strength juice. The blend is stirred until a homogeneous state is reached. The receiver tank feeds flash-pasteurizer P and subsequently packaging line L .

Riedel's current production facility

Depending on objectives specified in the business plan of a company and developments in the market, a plant evolves during its life-cycle in order to satisfy the objectives. The configuration of processes is highly dependent on the product range and changes often during the life-cycle of the plant. We now focus on the situation of Riedel and the current configuration of processes in Riedel's production facility. In Figure 4.3 a process model of the plant is depicted.

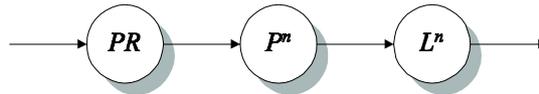
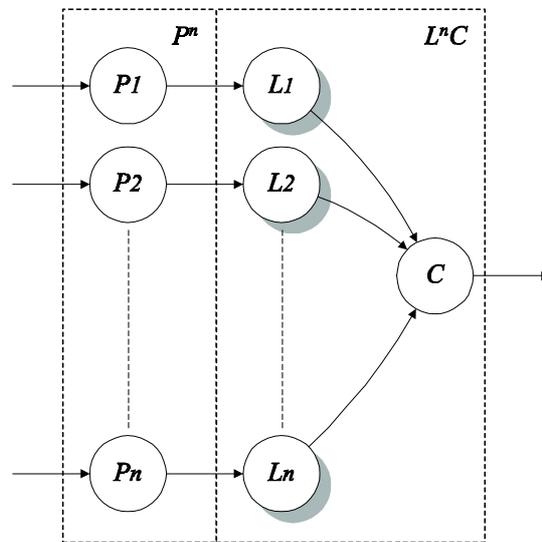


Figure 4.3: Process model of Riedel's plant.

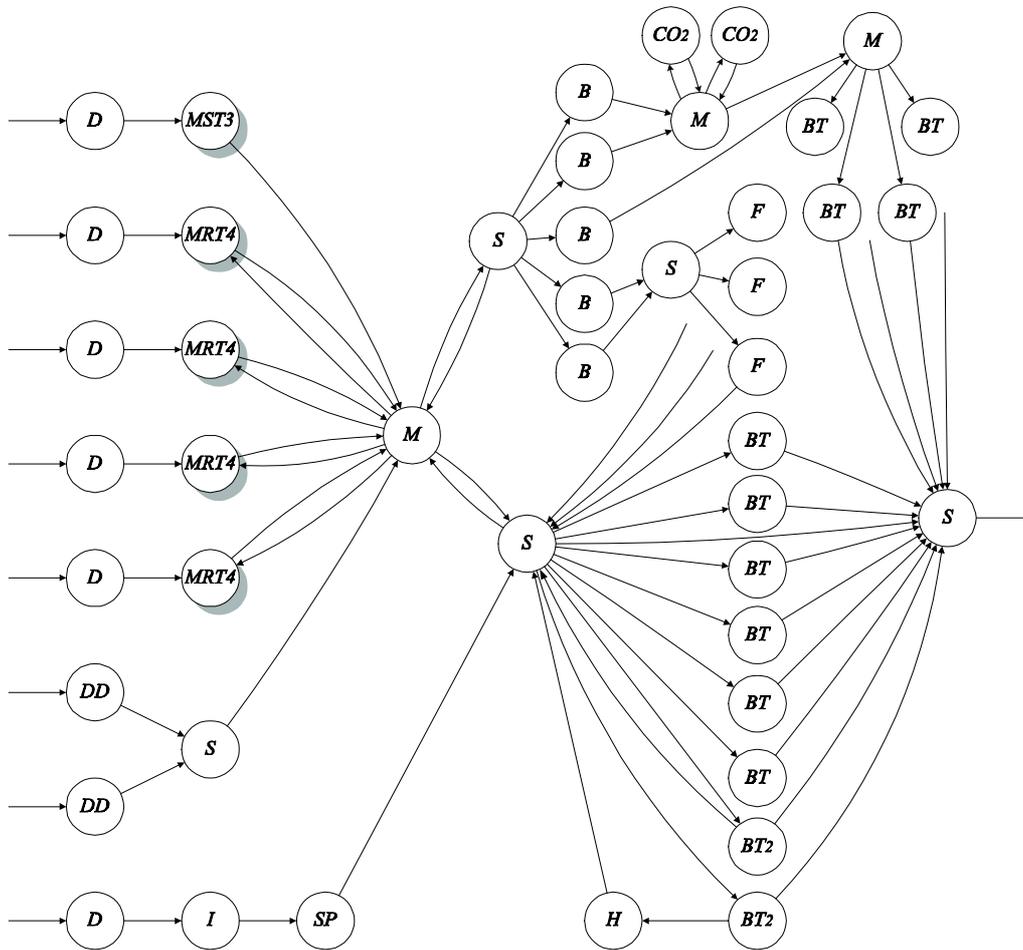
The model is composed of three subsystems, depicted as a shaded circle. System PR , represents the Preparation Department, system P^n represents the Pasteurizing Department, system L^n represents the Packaging Department. Every subsystem contains a complex of interacting processes. In Figure 4.4 system P^n and system $L^n C$ are depicted.

System P^n contains n pasteurizers P . System $L^n C$ contains n subsystems Li , where each subsystem contains one or two packaging lines, and a conveyor C which transports the pallets to the warehouse. Pasteurizer Pi , is connected to a subsystem

Figure 4.4: System P^n and system $L^n C$.

Li. Every pasteurizer P feeds one or a pair of packaging lines L , the structure of these systems and their interaction is relatively simple. System PR , representing the Preparation Department, by contrast, lacks an elegant structure. In Figure 4.5 the process structure of system PR is depicted.

System PR is composed of 40 processes, and 5 subsystems; subsystems $MRT4$ are composed of a manifold and 4 receiver tanks, and $MST3$ is composed of a manifold and 3 sugar sirup tanks. The production process starts with the discharging of raw materials received in bulk (process D) or drums (process DD). Discharging facilities D are connected to automated manifolds M to distribute the raw materials to receiver tank RT or, in case of sugar, to sugar tank ST . Special equipment, ion exchanger I , separator SP , and homogenizer H is installed to process ingredients containing protein. Drum dumpers DD are connected to manually operated switchboard S and to central automated manifold M . The drums are emptied and the concentrate or puree is pumped to one of the receiver tanks RT . Besides purchasing Just In Time, a number of ingredients is kept in stock in receiver tanks, e.g. apple juice, orange juice, and sugar sirup. These ingredients are pumped to a tank via central manifold M whenever needed. A receiver tank is filled with the necessary ingredients and subsequently stirred. When a pasteurizer and a packaging line is available, the natural strength blend is pumped to one or a number of pasteurizers P and packaging lines L . To create a route from a receiver tank to one or more pasteurizers, a number of manifolds M and switchboards S is used. Depending on the product, the route possibly contains blender B , carbon dioxide injector CO_2 , filter F , and buffer tank BT .

Figure 4.5: System *PR*.

The last few years the product range of Riedel has increased rapidly (see Section 2.3), therefore, increasing the flexibility of the plant has been an important issue. Furthermore, Riedel was convinced that flexibility enabled high equipment utilization and the possibility of anticipating on equipment breakdowns through rerouting. This flexibility policy is reflected in the large number of manifolds and switchboards. As a result, the material flow system is very complex and nontransparent. The complexity of the material flow system is reflected by the number of routes. When we only consider the production routes, i.e., the routes that lead from the receiver tanks to the packaging lines, there are more than 50 thousand possible routes. When the receiving of raw materials is taken into account the number of routes gets practically innumerable. For every production batch a route is created by switching valves and connecting pipes on switchboards. As a result, shop floor control is labor intensive

and failure sensitive. Automation of these manually operated switchboards can reduce labor costs significantly. Unfortunately, this is very expensive, e.g., automating a ten by ten switchboard costs more than 600 thousand US dollars, and the Preparation Department contains five switchboards of comparable size, and four smaller switchboards. The flexible but complex material flow system requires an advanced, extensive and therefore expensive shopfloor control system.

In general it can be stated that flexibility in a juice plant is expensive and demanding for hardware, software and labor. According to Burbidge [Bur96], companies with a flexible process organization tend to suffer from the ‘drunken spider’, or the ‘spaghetti’ syndrome. If the material flow system is too complex, controllability decreases and, as a result, flexibility vanishes.

For Riedel as well as any other juice plant simplicity needs to be the credo. Flexibility is only to be applied when it is functional and necessary. In discrete manufacturing environments analog problems have occurred. Below developments in discrete manufacturing technology, and their applicability in a juice plant are discussed.

Group Technology

Traditionally, the major part of discrete production environments were laid out either product focussed or process focussed. A product focussed layout (also called a flow shop) is a format in which the equipment is placed in a line, where a line is capable of performing all operations necessary for the completion of a product. A product layout is designed to produce high volume standardized products. Processes are adapted to the product, and physically arranged in the sequence required. Adopting a product layout makes sense when the batch size of a given product is large relative to the number of different products or parts produced [Cha98]. The product layout has found its greatest field of application in assembly, rather than in fabrication [Buf87], the major part of the discrete production environments were laid out process focussed (also called job shops). The manufacturing equipment is allocated according to its function in the production process. This results in a plant where we can clearly distinguish groups of similar equipment. During production, products in process are transported from one group to another in order to undergo the required operations, a product can visit a single group several times. The routing of the products in a process layout is dictated by individual product requirements; therefore, process oriented manufacturing causes a very complicated material flow system. Process focused layouts are applied when flexibility is essential; a large range of products are produced, often according to customer specifications. An example of a process focused manufacturing system is a machine shop, where the shop floor is laid out in

departments, each capable of performing a generic function such as drilling, planing, lathing, and assembling.

Unfortunately, nowadays, the major part of manufacturing systems have a diverse product range that contains neither strictly customer specified small volume products, nor strictly standardized large volume products. As a consequence, neither a product layout nor a process layout is a good fit. This was first realized by Mitrofanov [Mit66], who introduced the term Group Technology (GT) as the title for his research into the relationship between component-shape and processing methods in a machine shop. Later on GT was applied in other industries such as foundries, forges, welding, wood-working and sheet-metal shops. Today, GT is mainly applied in jobbing and batch production factories [Bur96]. A GT layout combines a product and a process layout, providing some of the advantages offered by each basic type [Buf87]. GT has as its objective the simplification of material flow, the development of a more rewarding work environment and the decentralization and simplification of shop floor control and scheduling [Moo95].

According to Moodie [Moo95] GT offers simplification of the material flow, shop floor control and scheduling, which is just what we are looking for to improve Riedel's plant. Now, the question arises: 'Is GT applicable in Riedel's production facility?'. Schonberger [Sch86] gives general guidelines for deciding whether or not applying GT makes sense: distinct parts families exist; there are several of each type of manufacturing equipment; the equipment is easily moveable. The first and second guideline are satisfied: the product range of Riedel comprises several distinct brands, each of them, in turn, comprises several blends. The major part of the equipment types is present more than once. The third guideline is violated: in general, the manufacturing equipment for the production of fruit juices is not easily moveable. Therefore, changing a juice plant with a process or product layout in a GT layout is very expensive. In this case GT is implemented in the shop floor control system rather than in the shop floor layout. Conclusively, it is stated that GT is applicable for the Riedel juice plant.

Applying GT in Riedel's production facility

The Riedel plant is laid out in three departments; the Pasteurizing and Packaging Department have a simple structure, therefore, the division in groups is restricted to the Preparation Department.

A challenging aspect in designing a GT plant is the division of products into groups. The amount of literature on this subject reflects its complexity. In the early days classification and coding systems (C&C) were very popular tools to solve this prob-

lem. C&C systems are designed for discrete machine shop like environments with thousands of parts to be produced. C&C systems are based on the assumption that parts with the same shape or function can all be produced by the same group of machines [Bur96]. In the preparation of fruit juice beverages, parts have no other function than taste, and shape is an irrelevant aspect, as we are dealing with fluids. In the packaging of fruit juices, the number of shapes, or in other words types of packaging, is very small compared with a machine shop. Therefore, C&C methods cannot be applied in a fruit juice blending and packaging plant.

Burbidge showed that Production Flow Analysis (PFA) is better suited for planning GT [Bur71]. In short, PFA analyses the routing of products in the manufacturing system and clusters products with similar routings. PFA involves four stages: (1) classifying processes, (2) collecting product routing information, (3) constructing one or more machine-component matrices, and (4) manipulating these matrices to form component groups. At this stage of the design process, we do not consider machines, but use the more abstract term processes. Furthermore, the term component is translated as blend.

Stage 1: Classifying processes. The collection of processes can be divided in a number of subsets, representing the main process groups: receiver tanks, blenders, carbon dioxide injectors, ion-exchangers, homogenizers, separators, filters, buffer tanks, utilities, drum dumpers, discharging facilities, and manifolds. The designer has to decide which of the process groups are relevant in the PFA; processes required for minor and ancillary operations have to be excluded from the analysis. An example of processes for ancillary operations for Riedel are utilities and manifolds. Furthermore, processes that are being used for the majority of blends, such as receiver tanks, discharging facilities and drum dumpers, are better excluded. When this rationalisation is not carried out, practice has shown that excessive distortion of the process-blend matrix in Stage 4 is likely to result [Kin95]. When the group formation has succeeded the processes excluded from the analysis are added.

Stage 2: Collecting routing information. For every product, the necessary processes are determined. Only the relevant process groups of Stage 1 need to be considered.

Stage 3: Constructing the process-blend matrix. The information gathered in the stages 1 and 2 are combined in a process-blend matrix. The matrix contains only binary valued cells. A cell entry $x_{ij} = 1$ if product j requires process i , else $x_{ij} = 0$.

Stage 4: Manipulating the process-blend matrix. Stage 4 deals with the formation of groups. In theory, the process-blend matrix has to be manipulated in such a way that mutually exclusive clusters of 1s are formed along the diagonal of the matrix. Each cluster represents a candidate group. Literature presents several strategies for finding such clusters. King [Kin80] has developed the Rank Order Clustering method (ROC) to generate diagonalized clusters. The ROC is a relatively simple procedure, which can be implemented easily. The method iteratively ranks the rows and subsequently the columns of the matrix in order of decreasing binary value. The binary value is determined by reading the pattern of the cell entries of each row and each column as a binary word. When the rank order of the rows and columns remains unchanged, the procedure terminates. In Figure 4.6(a), an example of a process-blend matrix is depicted, together with the binary value representation of every row and column. Figure 4.6(b) shows the matrix after manipulating the ordering of rows and columns with the ROC method.

		Processes						
		bin.	210	111	82	128	25	45
		bin.	1	2	3	4	5	6
Blends	36 A	1				1		
	56 B	1	1	1				
	17 C		1					1
	42 D	1		1			1	
	19 E		1				1	1
	17 F		1					1
	56 G	1	1	1				
	19 H		1				1	1

(a)

		Processes						
		bin.	240	128	112	111	28	15
		bin.	1	4	3	2	5	6
Blends	48 A	1	1					
	44 B	1		1	1			
	44 G	1		1	1			
	42 D	1		1			1	
	7 E				1	1	1	1
	7 H				1	1	1	1
	5 C				1			1
	5 F				1			1

(b)

Figure 4.6: A process-blend matrix, (a) initial and (b) diagonalized with ROC.

In practice, the resulting matrix will often not be exclusively composed of mutually exclusive clusters. In fact, when the ROC method is applied to Riedel's process-blend matrix, no mutually exclusive clusters are formed, as in Figure 4.6(b). However, blends with the same binary value require identical processes and form a candidate group, for instance blends *E* and *H*. The designer needs additional criteria to decide whether the group formation is realistic. We distinguish two additional criteria that play an important role in the process of group determination: utilization and packaging variants, in that order of importance. The utilization of processes in every group needs to be high for cost efficiency reasons. Through evaluation of the production volume per product and the routing of products in a group, process utilization needs to be estimated. If evaluation of the utilization does not solve the group formation

problem, the number of packaging variants per group are evaluated. Decreasing the number of packaging variants per group, leads to decreasing the interaction between product groups during pasteurizing and packaging, which is considered important when applying GT.

The process-blend matrix of Riedel is composed of forty blends and seven process groups. These forty blends are divided in six groups, four of them representing a distinct brand, the other two groups consisting of a large number of relatively small and medium volume blends that cannot be incorporated in one of the first four groups. In Table 4.1 this group division is depicted.

Table 4.1: Division of product range in groups.

Group	Brand
1	<i>Appelsientje</i>
2	<i>Goudappeltje</i>
3	<i>Carbonated drinks</i>
4	<i>Taksi</i>
5	<i>Rest, in-line dilution</i>
6	<i>Rest, in-tank dilution</i>

Groups 1 and 2 consist of one blend *Appelsientje* (orange juice), and *Goudappeltje* (apple juice), respectively. These blends are produced in such large volumes that they can form a complete group on their own. The *Carbonated drinks* and *Taksi* groups consist of seven and four blends, respectively. The two *Rest* groups contain products which do not fit in one of the first four groups. Products which can be prepared by in-line dilution are separated from products that can only be prepared by in-tank dilution. Figure 4.7 shows the production volume per blend for every group, and the cumulative production per group.

For every group the required processes are selected and configured according to the production procedures. In Figure 4.8 a process model of the Riedel juice plant with Group Technology is depicted, the majority of processes is assigned to one of the six groups.

Although the configuration of every group differs, the main structure is similar. The first processes in every group represent discharging facility D and in some cases drum dumper DD ; in the *Taksi* group the discharging facility is coupled with equipment for the processing of ingredients containing protein (process I , SP , BT , and H). The raw materials are distributed via a manifold to a receiver tank; the manifold and n receiver tanks are incorporated in system $MRTn$, which is depicted as a shaded circle. In the receiver tank the concentrated or natural strength blend is composed and

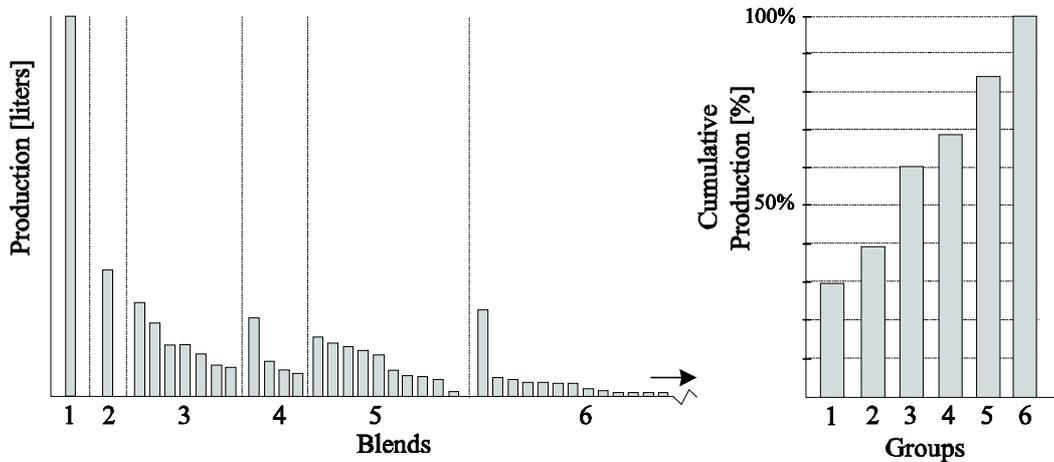


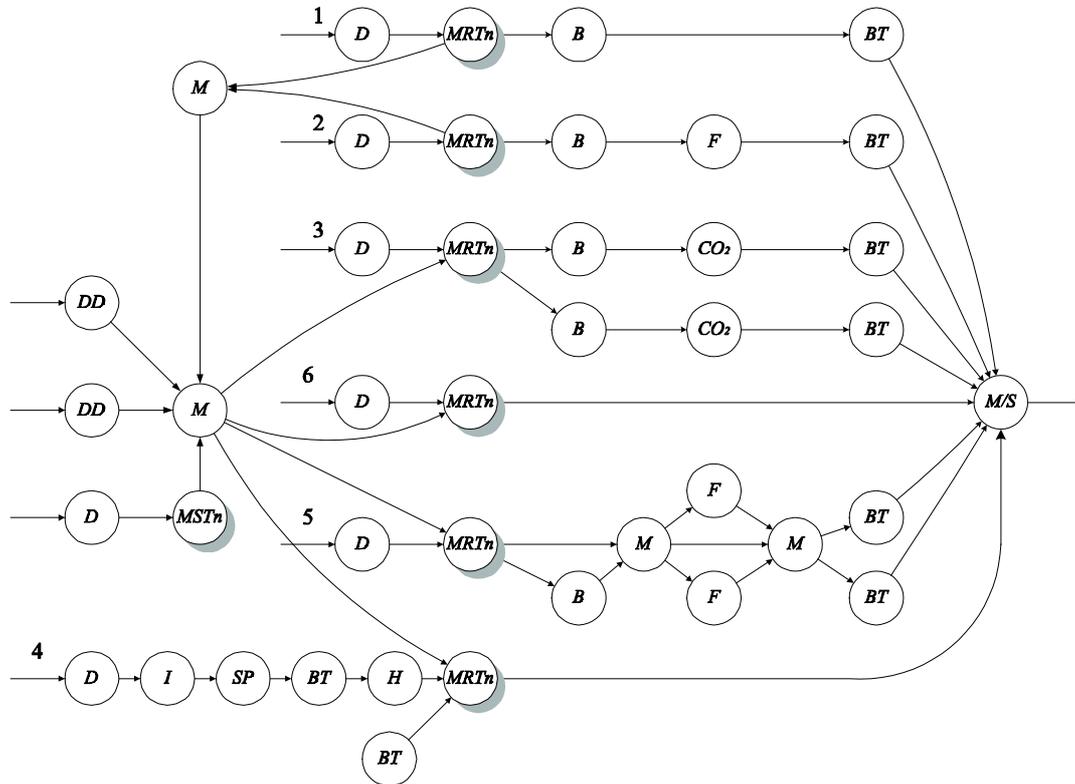
Figure 4.7: Production per blend and cumulative production per group.

stirred. From the receiver tank, the juice is pumped to pasteurizers and packaging lines, when necessary, via downstream equipment such as a blender, a filter, a carbon dioxide injector, and a buffer tank. For financial reasons, drum dumpers *DD* and sugar tanks *ST* are not incorporated in each of the groups separately, but operate as a central utility.

In contrast with the plant without GT, the focus on flexibility in the plant with GT is restricted to the groups 5 and 6. The equipment in these groups is configured in such a way that a large variety of products can be produced, and new products can be tested. The rerouting of products in case of equipment breakdowns remains possible, as products can often be transferred to the equipment of the groups 5 and 6.

4.3 Conclusion

In this chapter an architecture for Riedel's production facility has been designed. The design of architecture is divided into two subsequent steps: the sequencing of functions and the configuration of processes. For the blending and packaging of fruit juice beverages three basic functions are distinguished: preparing, pasteurizing, and packaging. These functions have to be performed sequentially. Important aspects in determining the best sequence are the product range and technological developments. For Riedel Drinks, the functions preparing, pasteurizing, and packaging are best performed in that sequence. The second step is determining the configuration of processes. The process structure responsible for pasteurizing and packaging clearly

Figure 4.8: System *PR* with group technology.

differs from the process structure responsible for preparing the beverages. The pasteurizing and packaging processes form a relatively simple flow line, the preparing processes have a far more complex and nontransparent structure. The focus on flexibility has led to a failure sensitive and labor intensive Preparation Department. In order to simplify the manufacturing system Group Technology has been applied. The product range has been divided into six groups and for every group the necessary processes have been selected and configured. The designed process model provides a manufacturing structure that is flexible where needed, but incorporates simple and cost efficient structures whenever possible.

Chapter 5

Resources

In the previous chapter an architecture for Riedel's production facility has been designed. In this chapter the architecture is quantified, the objective is to determine a list of resources and to constitute a floorplan. At this stage in the design process quantifying resources is a first order approximation. In general, little detailed information about the operation of the plant to be designed or the sales volume of products is available. However, even if a lot of detailed information is available, taking it into account at this stage works counter-productive. To support the design activities in the design of resources phase and to improve the quality of the design, two design tools have been developed.

In Chapter 4 a clear distinction has been made between the architecture responsible for pasteurizing and packaging, and the architecture responsible for preparing: the former has a flow line structure, the latter a Group Technology structure. In this chapter the same distinction is made. Quantifying the pasteurizing and packaging equipment precedes quantifying the preparation equipment. This is done for two reasons. First, packaging equipment is labor intensive and requires large investments, therefore, equipment utilization needs to be as high as possible to guarantee a high return of investments. As a consequence, packaging equipment has to be the bottleneck capacity and comes first in capacity analysis. Second, the number of packaging lines determines the number of blends packed simultaneously, which in turn is of large influence on the preparation equipment needed. In Section 5.1 the amount of packaging and pasteurizing resources is quantified. In Section 5.2 a list of resources for preparing the juice beverages is determined. Section 5.3 discusses the factory layout. Concluding remarks are presented in Section 5.4.

5.1 Pasteurizing and packaging

The objective of the design activity described in this section is to determine the number of pasteurizers and packaging lines needed, and the way they are configured. In the literature several techniques are presented to perform capacity planning. For example, [Vol92] describes four techniques, ranging from a relatively simple straightforward method called Capacity Planning Using Overall Factors which uses historical production data to estimate future requirements, to a more advanced method called Capacity Requirements Planning (CRP) which uses a Material Requirements Planning (MRP) to include timing aspects. The latter method is often used as a module in a Manufacturing Resource Planning system (MRPII), a production management system that extends the MRP with capacity requirements checks and additional features to cover business and financial planning. CRP is primarily used as a verification tool and not as a design tool [Bro96]. [Kra96] Describes a four-step procedure to support capacity decisions. Neither of these methods is fit for calculating the number of packaging and pasteurizing resources. First, these methods demand a production planning as input, as a result finished-good inventories are not taken into account. Second, they presuppose that the total amount of setup times are independent of the chosen resource configuration, for the considered pasteurizing and packaging resources this is not the case. Therefore, a nine-step procedure is developed to support the design process. For every packaging type this procedure is executed once.

To provide a proper framework for capacity analysis, time is divided into functional intervals. The available time t_a is partitioned into five parts: production time t_p , breakdown time t_b , preventive maintenance time t_m , setup time t_s , and idle time t_i (see Figure 5.1)

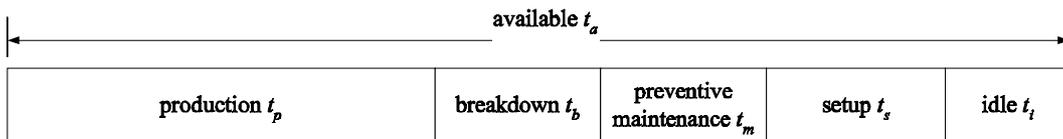


Figure 5.1: Division of available time.

or, for every time interval j

$$t_a(j) = t_p(j) + t_b(j) + t_m(j) + t_s(j) + t_i(j). \quad (5.1)$$

The basis for quantifying t_p are sales volume forecasts. Riedel is able to forecast sales volumes per stock keeping unit (in the sequel SKU) for four-week intervals for the next

two years with useful accuracy. In the remaining of this section, time interval j is considered to be a four-week interval, one year consists of thirteen of these intervals.

Step 1: Determine the lower bound for the number of packaging lines

Each SKU is packed on a specific type of packaging line. The sales volume v_s per packaging line type for every four-week interval j is determined by cumulating the sales volume of all SKUs packed on the considered type of packaging line. For each packaging line type, production volume v_p in each time interval, is calculated with

$$v_p(j) = i_e(j) - i_s(j) + v_s(j), \quad (5.2)$$

where v_s is the sales volume, i_s the inventory of finished products at the start of the interval, and i_e the inventory of finished products at the end of the interval. At this stage in the design process there is no detailed information available on the inventory levels i_e and i_s . We assume for all considered intervals j that

$$|i_e(j) - i_s(j)| \ll v_s(j), \quad (5.3)$$

or in other words, the inventory decrease or increase is negligible compared with the sales volume. Now, for the production volume v_p it follows that

$$v_p(j) \approx v_s(j). \quad (5.4)$$

For every interval j net production time t_p is calculated by

$$t_p(j) = \frac{v_p(j)}{c}, \quad (5.5)$$

where c is the packaging line capacity. Substituting (5.4) in (5.5) leads to

$$t_p(j) \approx \frac{v_s(j)}{c}. \quad (5.6)$$

Breakdown time t_b is determined by

$$t_b(j) = t_p(j) \cdot f_b, \quad (5.7)$$

where f_b is the machine breakdown factor with $0 \leq f_b < 1$. This breakdown factor is incorporated in the technical specifications provided by the suppliers of the machines in the line. Preventive maintenance t_m is some fraction of the production time t_p :

$$t_m(j) = t_p(j) \cdot f_m, \quad (5.8)$$

where f_m is the preventive maintenance factor with $0 \leq f_m < 1$. Again this factor is provided by the suppliers of the machines in the line. Substituting (5.6) in (5.7) and (5.8) results in

$$\begin{aligned} t_b(j) &\approx \frac{v_s(j)}{c} \cdot f_b, \text{ and} \\ t_m(j) &\approx \frac{v_s(j)}{c} \cdot f_m. \end{aligned} \quad (5.9)$$

To avoid undercapacity in an interval j , idle time t_i has to be nonnegative, hence

$$t_i(j) \geq 0. \quad (5.10)$$

Substituting the times t_p , t_b , t_m , and t_i from (5.6), (5.9), and (5.10) in (5.1) results in

$$t_a(j) \geq \frac{v_s(j)}{c} + \frac{v_s(j)}{c} \cdot f_b + \frac{v_s(j)}{c} \cdot f_m + t_s, \quad (5.11)$$

which is rewritten as

$$t_a(j) \geq \frac{1 + f_b + f_m}{c} \cdot v_s(j) + t_s(j), \quad (5.12)$$

The available time t_a equals the duration d of the considered interval multiplied by the number of available packaging lines n , or

$$t_a(j) = n(j) \cdot d(j). \quad (5.13)$$

Substituting (5.13) in (5.12) yields

$$n(j) \cdot d(j) \geq \frac{1 + f_b + f_m}{c} \cdot v_s(j) + t_s(j). \quad (5.14)$$

Subsequently, the number of packaging lines $n(j)$ has to satisfy

$$n(j) \geq \frac{1 + f_b + f_m}{c} \cdot \frac{v_s(j)}{d(j)} + \frac{t_s(j)}{d(j)}. \quad (5.15)$$

Now, to calculate the number of packaging lines $n(j)$, only the time for setups $t_s(j)$ needs to be determined. Unfortunately, the value of t_s is not only related to the

number of batches produced every interval, but also to the chosen configuration of pasteurizers and packaging lines. To provide a lower bound for the number of packaging lines, the time for setups is set to zero, and this is substituted in Inequality 5.15. The lower bound is for every considered interval j , the smallest nonnegative integer n that satisfies

$$n(j) \geq \frac{1 + f_b + f_m}{c} \cdot \frac{v_s(j)}{d(j)}, \text{ where } n \in \mathbb{N}, \quad (5.16)$$

or, in other words, the value of $\frac{1+f_b+f_m}{c} \cdot \frac{v_s(j)}{d(j)}$ rounded to the next nonnegative integer. The number of necessary packaging lines n is likely to differ per time interval j . However, Riedel is not able to change the number of installed packaging lines every four weeks. The number of packaging lines that should meet capacity requirements for every interval is denoted by n^* , and defined as the maximum value of n for all intervals $j \in \{1, \dots, 13\}$

$$n^* = \max_j n(j). \quad (5.17)$$

Step 2: Determine the available time for setups t_{sav}

Now, it has to be determined whether n^* packaging lines is enough to meet the capacity requirements when setup times are included. In this step the available time for setups is calculated, which is defined as

$$t_{sav}(j) = t_s(j) + t_i(j). \quad (5.18)$$

Although $t_s(j)$ as well as $t_i(j)$ are yet unknown, their sum, t_{sav} , can be determined. Rewriting (5.1) leads to

$$t_s(j) + t_i(j) = t_a(j) - t_p(j) - t_b(j) - t_m(j), \quad (5.19)$$

or,

$$t_{sav}(j) = t_a(j) - t_p(j) - t_b(j) - t_m(j). \quad (5.20)$$

Substituting (5.17) in (5.13) yields

$$t_a(j) = n^* \cdot d(j). \quad (5.21)$$

Substituting times t_a , t_p , t_b and t_m from the (5.21), (5.6), and (5.9) in (5.20) leads to

$$t_{sav}(j) = n^* \cdot d(j) - \frac{1 + f_b + f_m}{c} \cdot v_s(j), \quad (5.22)$$

which is used to evaluate $t_{sav}(j)$ for all intervals j . Where $t_{sav}(j)$ only depends on the number of packaging lines n^* , the necessary time for setups $t_s(j)$ depends on the complete pasteurizer-packaging configuration.

Step 3: Develop possible pasteurizer-packaging configurations

In this step, a number of alternative pasteurizer-packaging configurations is designed. This is done on the basis of n^* calculated in Step 1. Suppose $n^* = 2$, then there are two possible configurations. The first contains two pasteurizers and two packaging lines, where each pasteurizer feeds one packaging line. An alternative is one pasteurizer and two packaging lines, where the pasteurizer feeds both packaging lines simultaneously. Alternatives where the number of pasteurizers exceeds the number of packaging lines are not interesting, because the setup time of a packaging line is larger than the setup time of a pasteurizer, and the maximum flow generated by one pasteurizer easily feeds one or two packaging lines.

Step 4: Determine the necessary setup time t_s for each configuration

For each pasteurizer-packaging configurations the necessary setup time $t_s(j)$ is calculated for every time interval j . The necessary setup time is linearly dependent on the number of batches produced in a time interval according to

$$t_s(j) = n_b(j) \cdot d_s, \quad (5.23)$$

where $n_b(j)$ is the number of batches in interval j , and d_s is the average duration of a setup. In general, the influence of seasonality is restricted to the size of the production batches. The number of production batches per four-week interval is more or less constant. As a consequence, setup time is time independent and calculated by

$$t_s = n_b \cdot d_s. \quad (5.24)$$

For each configuration $t_{sav}(j)$ is compared with t_s to determine idle time $t_i(j)$ for each configuration in each time interval j . Idle time $t_i(j)$ reflects whether a configuration meets its capacity requirements.

Step 5: Determine idle time t_i per interval for each configuration

In this step idle time $t_i(j)$ is computed for all j , and for each configuration. The idle time $t_i(j)$ is calculated by

$$t_i(j) = t_{sav}(j) - t_s, \quad (5.25)$$

where $t_{sav}(j)$ and t_s are determined in Step 2 and Step 4, respectively. When for all intervals j the available setup time $t_{sav}(j)$ exceeds the necessary setup time t_s ,

then it follows that $t_i \geq 0$, hence, the considered configuration satisfies its capacity requirements. When all configurations satisfy the capacity requirements in all intervals, Step 6 is skipped and the procedure is continued in Step 7. When in an interval j , $t_i(j) < 0$ the configuration is under-capacitated for that interval. Furthermore, when

$$\sum_j t_i(j) < 0, \quad (5.26)$$

temporary under-capacity cannot be compensated with inventories: the configuration does not satisfy capacity requirements. Consequently, the number of packaging lines n^* is raised with one and the procedure is repeated starting at step 2.

However, when for one of the configurations

$$\sum_j t_i(j) \geq 0, \quad (5.27)$$

it is possible to compensate under-capacity in one interval with increase of inventories in preceding intervals. This type of inventory is often referred to as anticipation inventory. If for one or more of the time intervals $t_i(j) < 0$, the idle time is reallocated in Step 6 such that $t_i(j) \geq 0$ holds for every interval j .

Step 6: Determine idle times t_i , such that $t_i \geq 0$ holds for all intervals

Idle time in over-capacitated periods has to be used for production, to compensate for under-capacitated periods. In order to reallocate the idle times an algorithm has been designed. In Figure 5.2 a physical analogy of the algorithm is depicted. The analogy consists of a fixed shaft and a rotating disk. The disk has a mountainous surface, where a hill represents under-capacity, and a valley represents over-capacity. The shaft scrapes the hills from the surface and dumps them in the valleys, or in other words, it moves negative idle time from intervals with under-capacity to preceding intervals with over-capacity. The algorithm is able to perform its task when the cumulated area of the valleys is larger than or equal to the cumulative area of the hills, or in other words, the total amount of idle time is positive. It starts in the last interval 13, and moves back to interval 1. The algorithm is finished when the disk has rotated once without the shaft encountering a hill.

In Figure 5.3 the algorithm is specified as a function $T: \mathbb{R}^* \rightarrow \mathbb{R}^*$, where \mathbb{R}^* denotes a set of lists of reals. Function T is specified with the formalism χ [Are96], [Roo97], and [Bee98]. The disk of Figure 5.2 is represented as a list ts , the real valued elements of the list represent the idle times in the consecutive intervals. Initially, the idle time of the first interval is the head of the list. The value of the nonnegative integer n

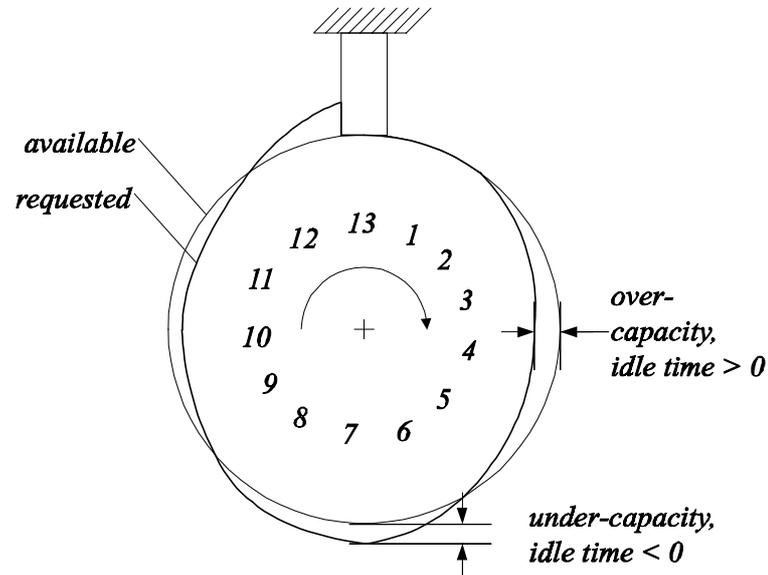


Figure 5.2: Physical analogy of the idle time reallocation algorithm.

```

func T(ts : real*) → real* =
  [ [ n : nat; t : real
    | n := 0
    ; *[ n < len(ts)
      → t := hr(ts); ts := tr(ts)
        ; [ t ≥ 0 → ts := [t] ++ ts; n := n + 1
          [ t < 0 → ts := [0.0] ++ tr(ts) ++ [hr(ts) + t]; n := 0
        ]
      ]
    ]
  ; ↑ ts
  ]

```

Figure 5.3: The idle time reallocation algorithm.

represents the number of successive intervals with a positive idle time. When n is larger than the length of the list ts , it means that the complete list is evaluated once without encountering a negative idle time, in other words, the algorithm has found a solution. As long as n is smaller than the length of list ts ($*[n < \text{len}(ts) \rightarrow \dots]$) the following loop is repeated. The algorithm selects the last element of list ts and

stores it in variable t ($t := \text{hr}(ts)$). Then this element is removed from the list ($ts := \text{tr}(ts)$). If t is larger than or equal to zero, it is added as first element to the list, and n is raised with one ($ts := [t] \uparrow ts; n := n + 1$). If t evaluates negative, its value is added to the value of the last element in the list, and zero is added as first element in the list ($ts := [0.0] \uparrow \text{tr}(ts) \uparrow [\text{hr}(ts) + t]$). The value of n is set to zero ($n := 0$). When n is equal to the length of list ts , the algorithm has encountered no negative valued element of list ts during a complete evaluation of the list, and as a result Function T returns the modified list ts ($\uparrow ts$).

Step 7: Determine production volume and inventory levels

In Step 7 the production volume and finished-good inventory levels are computed for each configuration. These inventory levels compensate for sales fluctuations between intervals, but not for fluctuations in intervals. As Riedel demands a service level of 98%, additional safety stocks are necessary to compensate for day-to-day fluctuations in demand. These safety stocks are not considered, hence, the inventory levels calculated here are lowerbounds.

When inventories are taken into account, (5.3) and (5.4) become invalid. From Equation (5.2) it follows that

$$i_e(j) = v_p(j) - v_s(j) + i_s(j). \quad (5.28)$$

Sales volume $v_s(j)$ is forecasted, production volume $v_p(j)$ is deducted from (5.1). Substituting the times t_a , t_p , t_b , t_m , and t_s from the respective Equations (5.21), (5.5), (5.7), (5.8), and (5.24) in (5.1) results in

$$n^* \cdot d(j) = \frac{1 + f_b + f_m}{c} \cdot v_p(j) + n_b \cdot d_s + t_i(j). \quad (5.29)$$

Rewriting this equation leads to

$$v_p(j) = \frac{c}{1 + f_b + f_m} \cdot (n^* \cdot d(j) - n_b \cdot d_s - t_i(j)). \quad (5.30)$$

The inventory at the end of an interval equals the inventory at the start of the next interval, hence

$$i_s(j + 1) = i_e(j). \quad (5.31)$$

With (5.30) the production volume $v_p(j)$ is calculated for all j . Next, $i_s(1)$ is set to zero, and applying (5.28) yields $i_e(1)$. Subsequently, (5.31) is used to determine

$i_s(2)$, and the procedure is repeated until $i_e(j)$ is determined for all j . Inventories compensate for temporary undercapacity when

$$i_e(j) \geq 0, \quad (5.32)$$

holds for all j . If not, $i_s(1)$ is raised such that Inequality (5.32) holds for all intervals.

Step 8: Determine inventory turnaround time t_t

The inventory consists of perishable products. Therefore, the turnaround time $t_t(j)$ (also called inventory average age) of the inventory in each interval is calculated with

$$t_t(j) = \frac{i_e(j) + i_s(j)}{2 \cdot v_s(j)}, \quad (5.33)$$

or, in words, the average inventory divided by the sales volume. In order to be able to guarantee a proper shelf life, t_t has to be smaller than some predefined maximum, which is product dependent. If this is not the case, n^* is raised with one, and the procedure is repeated starting at step 2.

Step 9: Economic evaluation of all remaining configurations and selecting the best

All remaining configurations provide enough production capacity. Now, economic evaluation is used for selecting one of the remaining configurations. Every capital expenditure proposal should contain evaluation of costs and benefits. Benefits are generated by selling products. The number of products sold is independent of the chosen alternative, hence, to compare the alternatives only costs have to be considered. Therefore, economic evaluation in this step is restricted to calculation of total cost.

Total cost consist of inventory and production related costs. Riedel's inventory holding cost solely originates from storing finished goods, because raw materials are delivered just in time, and processed immediately. Inventory holding cost includes interest, storage and handling costs, taxes, and insurance. The production related costs consist of costs due to fixed-asset investments and operational costs.

Total costs is one the most important criteria for selecting a configuration. Nevertheless, other criteria play a role, especially when two or more configurations are financially equally attractive. For instance, increasing the number of setups negatively influences the machine efficiency. Unfortunately, this effect is hard to quantify

and therefore cannot be integrated in the capacity analysis procedure. Nevertheless, it can be taken into account when a number of configurations appear equally attractive. Another criterium is that large inventories are often unwanted, although they may seem profitable. An important aspect in this matter is that large inventories negatively influence flexibility.

To illustrate this process of quantifying the packaging resources, a pasteurizer-packaging configuration is designed for Riedel's 1000ml CFA207 packaging lines.

Riedel example

Riedel's CFA207 packaging lines produce 22 SKUs. These packaging lines consist of a CFA207 filling machine, a cap applicator *CA*, a tray packer *TP* and a palletizer *PA* (see Figure 5.4).

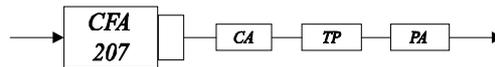


Figure 5.4: Schematic representation of a CFA207 packaging line.

Table 5.1 shows the relevant parameters for the CFA207 lines, where n_b is the number of production batches in a time interval and d_s the average duration of a setup.

Table 5.1: Parameters for the CFA 207 packaging line.

c	[cans/hour]	7000
f_b	[-]	0.25
f_m	[-]	0.05
n_b	[-]	60
d_s	[hours]	2

In Table 5.2 the forecasted sales volume v_s and the duration d are depicted for all four-week intervals j in a year. Below the nine-step procedure is applied to determine the necessary pasteurizing and packaging equipment.

Step 1. The necessary number of packaging lines n is calculated with Equation 5.16 and the result is depicted in Table 5.2. Table 5.2 clearly shows the seasonal effect that Riedel has to deal with: forecasted sales volume fluctuates from 4 to 5.6 million cans. This results in the need for 2 packaging lines in periods 1 up to 3 and 10 up to 12, and 3 packaging lines in periods 4 up to 9 and 13; as a result the lower bound $n^* = 3$.

Step 2. The available time for setups t_{sav} is calculated with Equation 5.22, for all j its value is depicted in Table 5.2. In a configuration with three CFA207 packaging lines, the time available for setups t_{sav} fluctuates from 109 hours in period 13 up to 604 hours in period 12.

Table 5.2: The available setup time t_{sav} per period.

j	1	2	3	4	5	6	7	8	9	10	11	12	13
v_s [10^6 cans]	4.0	4.4	4.4	4.8	5.1	5.3	5.6	5.5	5.4	5.0	4.6	4.1	5.6
d [hours]	431	455	455	431	407	431	455	455	455	455	455	455	383
n [—]	2	2	2	3	3	3	3	3	3	2	2	2	3
t_{sav} [hours]	550	548	548	383	274	290	325	344	344	436	474	604	109

Step 3. In Figure 5.5 three possible pasteurizer/packaging configurations are depicted, where P denotes a pasteurizer and TA denotes an aseptic tank. The aseptic tank is needed when a pasteurizer is connected to a single packaging line, to prevent the pasteurizer from stopping immediately whenever the packaging line breaks down.

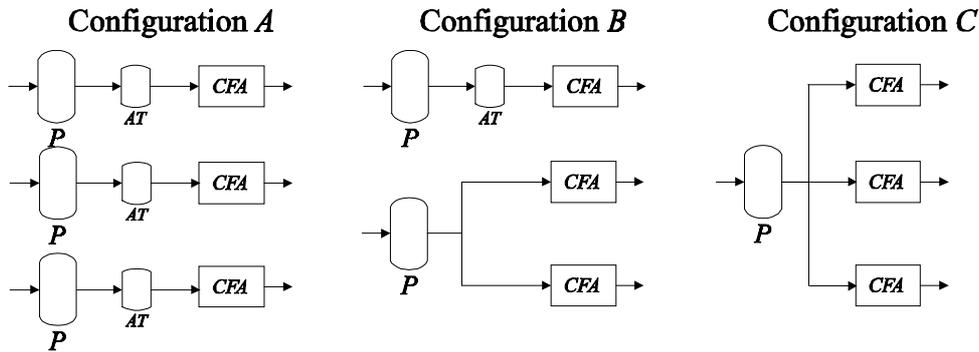


Figure 5.5: Possible CFA207 pasteurizer/packaging configuration.

Step 4. The total amount of setup time differs per configuration. Suppose the number of batches b equals 60 (see Table 5.1). Then, with configuration A, 60 setups need to be performed. With configuration C, on the other hand, every of the 60 batches is packed on all three packaging lines simultaneously, and, as a result, $3 \cdot 60 = 180$ setups need to be performed. The average setup duration is 2 hours and once every week the filling machine is cleaned thoroughly. This weekly setup takes 6 hours per filling machine and replaces a normal setup. The number of production batches per blend is constant throughout the year, only the batch size fluctuates.

Consequently, the amount of setup time is equal for all four-week intervals in a year. Table 5.3 shows the total amount of setup time per configuration per four-week interval.

Table 5.3: Total setup time per four weeks, in [hours].

Configuration	t_s
<i>A</i>	168
<i>B</i>	248
<i>C</i>	408

Step 5. From Equation 5.25, and the values of t_{sav} and t_s , idle time t_i is determined for all thirteen four-week intervals in a year (see Table 5.4).

Table 5.4: The idle time t_i for each configurations for all j , in [hours].

j	1	2	3	4	5	6	7	8	9	10	11	12	13	Σ
<i>A</i>	382	380	380	234	106	141	157	176	194	268	343	436	-59	3138
<i>B</i>	302	300	300	154	26	61	77	96	114	188	263	356	-139	2098
<i>C</i>	142	140	140	-6	-134	-99	-83	-64	-46	28	103	196	-299	18

Configurations *A* and *B* provide enough capacity in twelve out of thirteen intervals, and the sum of idle times is +3138 and +2098 hours, respectively. In interval 13 both configurations do not have enough capacity to produce the complete sales volume. On the other hand, in interval 12 both configurations are over-capacitated and are able to increase inventories temporarily to compensate for interval 13. Configuration *C* does not provide enough capacity in seven of the thirteen intervals, but the sum of idle times is +18 hours. On average *C* provides enough capacity, but inventories have to compensate for undercapacity in six adjacent intervals.

Step 6. In order to determine inventory levels and production volume per interval, idle time is reallocated for each configuration. Table 5.5 shows the results of executing Function T.

Table 5.5: Idle time t_i reallocated for each configurations for all j , in [hours].

j	1	2	3	4	5	6	7	8	9	10	11	12	13	Σ
<i>A</i>	382	380	380	234	106	141	157	176	194	268	343	377	0	3138
<i>B</i>	302	300	300	154	26	61	77	96	114	188	263	217	0	2098
<i>C</i>	0	0	0	0	0	0	0	0	0	18	0	0	0	18

Step 7. The maximum production volume v_p for each configuration is determined with Equation 5.30. With $i_s(1) = 0$, and alternately applying Equations 5.31 and 5.28, inventories are computed. In Table 5.6 the results are shown.

Table 5.6: v_s , v_p , and i_e for all configurations, in $[10^6 \text{ cans}]$.

j	1	2	3	4	5	6	7	8	9	10	11	12	13
v_s	4.0	4.4	4.4	4.8	5.1	5.3	5.6	5.5	5.4	5.0	4.6	4.1	5.6
A, v_p	4.0	4.4	4.4	4.8	5.1	5.3	5.6	5.5	5.4	5.0	4.6	4.4	5.6
A, i_e	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0
B, v_p	4.0	4.4	4.4	4.8	5.1	5.3	5.6	5.5	5.4	5.0	4.6	4.9	5.6
B, i_e	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.8	0.0
C, v_p	4.8	5.2	5.2	4.8	4.4	4.8	5.2	5.2	5.2	5.1	5.2	5.2	4.0
C, i_e	0.8	1.5	2.3	2.2	1.5	1.0	0.5	0.2	-0.1	0.0	0.6	1.6	0.0

The inventory level $i_e(9)$ is negative, raising the inventory at the start of the first interval, $i_s(1) = 0.1$, results in an inventory level that satisfies $i_e(j) \geq 0$, for all j . Figure 5.6 shows the inventory levels for the three configurations.

Step 8. The turnaround time per configuration is calculated with (5.33). In Table 5.7 the results are shown. The maximum average turnaround time is two weeks, which is acceptable for the products produced on the CFA 207 packaging lines.

Table 5.7: Average turnaround time t_t for each configurations, in $[\times \text{ four weeks}]$.

j	1	2	3	4	5	6	7	8	9	10	11	12	13
A	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
B	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0
C	0.1	0.3	0.4	0.5	0.4	0.3	0.2	0.1	0.0	0.0	0.1	0.3	0.2

Step 9. Annual holding costs are 35 euro per pallet, whereby each pallet contains about 1000 cans. The average inventory levels for configurations A , B , and C are 23, 62, and 1000 pallets, respectively. In Table 5.8 the annual holding costs and depreciation of pasteurizing and packaging equipment are depicted for the three configurations.

Configuration A is the least attractive configuration, because of the large annual depreciation of assets. Configurations B and C have comparable financial consequences. With respect to production capacity, B clearly outperforms C . When it turns out that the sales volume is underestimated, configuration C does not provide additional capacity to maintain a high service level. Furthermore, configuration B is

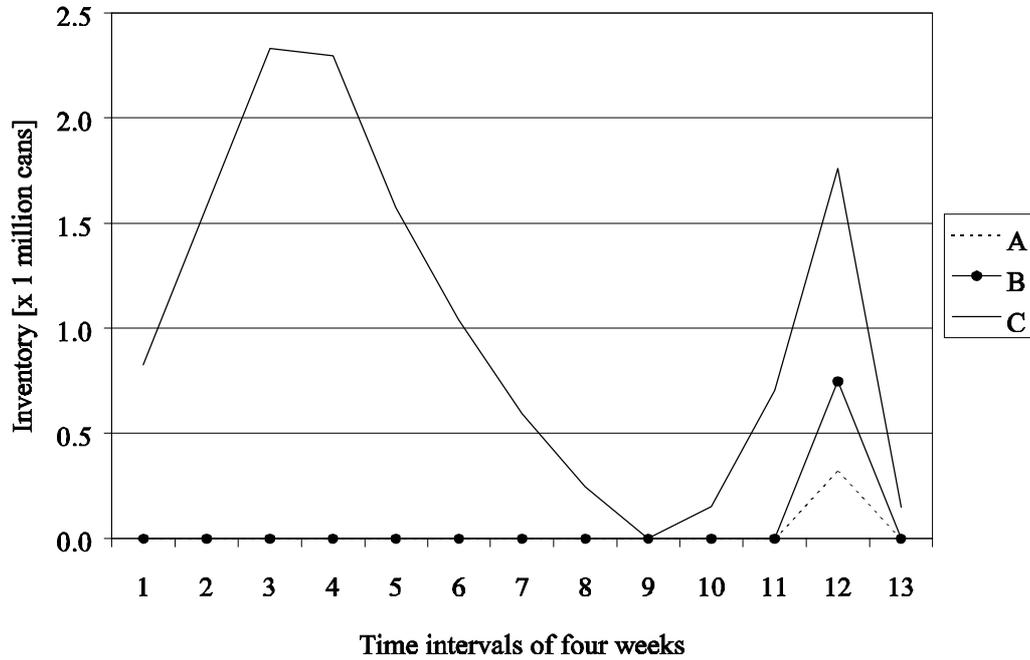


Figure 5.6: Inventory levels as function of time for each configuration.

Table 5.8: Annual costs for each configuration, in [$\times 1000$ US dollars].

	A	B	C
Depreciation of assets	122	86	46
Holding costs	1	2	35
	123	88	81

able to minimize inventory levels, which positively affects the lead time of products, and, as a consequence, product quality. It is concluded that configuration *B* is best fit for the production of the *Carbonated drinks* group.

The calculations above assume that a setup is without costs unless it is performed in overtime. If this is not the case, the costs of setups have to be incorporated in the financial analysis of each configuration.

□

5.2 Preparation

Applying GT in Riedel's Preparation Department simplifies the material flow, and, in addition, simplifies capacity analysis. The capacity analysis problem is reduced to seven independent problems: the Preparation Department is divided in six product groups, and one utility group. Each group is considered separately. Figure 5.7 shows an abstract representation of such a group.

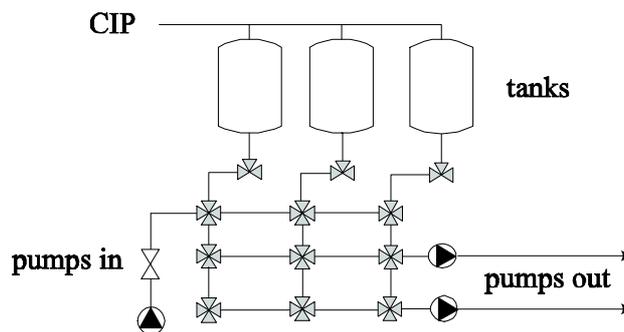


Figure 5.7: Abstract representation of a resource group.

A group consists of a number of ingoing pumps, tanks, outgoing pumps, and cleaning-in-place circuits (CIP). Now, the task is to determine the number of tanks, pumps, CIPs, and their capacity. Subsequently, additional resources such as blenders, filters, and buffer tanks can be determined straightforward.

As mentioned before, packaging line capacity comes first in capacity analysis. As a consequence, the number of outgoing pumps, i.e., pumps which transport the juice via downstream processing equipment to the pasteurizers and packaging lines, depends on the number of blends packed simultaneously. This can be either hardware restricted, or control restricted. For example, consider a product group which consists of four blends, and suppose there are three pasteurizer/packaging line pairs relevant for this product group. When the hardware is used as a restriction, three out of four blends can be packed simultaneously, and as a consequence, three outgoing pumps need to be installed. On the other hand, when control issues dominate, for instance when the scheduler decides that no more than two blends will be packed simultaneously, only two outgoing pumps need to be installed.

Discrete event model. In order to be able to calculate the number of tanks and pumps per product group a discrete event model is constructed with the specification

formalism χ . In χ a system is composed of several subsystems and/or processes that work concurrently. Processes communicate with each other via channels. In Figure 5.8 an iconic representation of the developed model is depicted. Processes and channels are represented by circles and arrows, respectively.

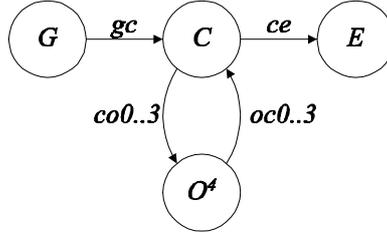


Figure 5.8: Iconic representation of the χ model.

The model consists of seven processes: generator G , exit E , controller C , and four operation processes O_0 up to O_3 (denoted by O^4). The four operation processes represent the four activities that have to be performed to produce a batch of juice, i.e., filling the tank with ingredients, stirring, emptying, and cleaning. The processes are connected through channels gc , ce , $co0..3$, and $oc0..3$, where $co0..3$ denotes a bundle of 4 channels $co0$ up to $co3$ which connect process C with the processes O_0 up to O_3 , respectively. Process G generates orders and sends these orders to controller C via channel gc . An order contains a unique order identity number, a batch size, and for monitoring purposes a start time. Controller C receives the order and puts the order in a list. The length of this list is limited to prevent overloading the controller. Controller C tries to execute an order whenever there is one available. Before execution can start, C checks whether the necessary resources are available. Every order starts with receiving raw materials, in order to be able to execute receiving raw materials, an ingoing pump and a tank have to be available. For every type of resource, controller C keeps up a list of available resources. The length of this list represents the number of available resources of that type. Before simulation starts, controller C reads configuration files and initiates the resource lists. When the length of the lists representing the ingoing pumps and the tanks are both larger than zero, the receiving of raw materials can start. From both lists one element is taken and these two elements are linked with the order; from that moment on it is called a job. The completion time is calculated and the job together with the completion time is sent to process O_0 which represents the receiving of raw materials. In process O_0 the actual processing takes place, modeled as a time delay. Process O_0 receives the job, puts it in a list, and sorts the list ascending according to completion time. When simulation time equals the completion time of the first job in the list, this job is finished and sent back to C . Controller C extracts the ingoing pump from the job

and adds it to a list of cleaning jobs. When a CIP circuit is available, the ingoing pump and the CIP circuit are sent to process O_3 to start the cleaning operation. When the cleaning job is finished, it is sent back to C and the pump and the CIP circuit are added to the corresponding lists: they are available again. The tank is now filled with raw materials and needs to be stirred. Stirring the tank is done in process O_1 . When the tank is stirred, the juice can be packed. A packing job can start when a tank, which is filled with homogeneous juice, and an outgoing pump are available. The tanks is emptied in process O_2 . When the tank is empty, the tank and the outgoing pump have to be cleaned in process O_3 . When the cleaning is finished a new batch can be prepared in the tank. The finished order is sent to exit E .

Data management. As mentioned above, the capacity analysis problem is decomposed into several smaller problems. This reduces the complexity of the problem, and in addition, the complexity of the data management. To facilitate case studies, the tool provides possibilities for changing the configuration of the simulation model. The user can easily study the effect of adding a tank, removing a pump, or changing the flow generated by a pump. Communication between the user and the simulation model is done via computer files which are generated by a graphical interface. The simulation model reads from these files to initiate the simulation. Initially, process G reads from a computer file the number of orders that have to be generated throughout the simulation, the batch size per order, and the minimal time between generation of two orders. Process C reads from a file the hardware configuration, for all resource types a resource list is generated. When we consider the example of Figure 5.7, controller C distinguishes four types of resources: ingoing pumps, tanks, outgoing pumps, and cleaning in place circuits. For every type C administrates a list, which initially has length 1, 3, 2, and 1, respectively.

Visualization. Essential for effective use of a capacity analysis tool is a user interface, which provides adequate visualization of the specified problem and simulation results. A specification of the problem consists of an equipment configuration and simulation parameters. The equipment configuration contains the number of ingoing pumps and their capacity, the number of tanks and their size, the number of outgoing pumps and their capacity, and the number of CIP circuits and their cleaning time. The simulation parameters consist of the number of batches to be simulated, the batch size, the time between two arrivals of raw materials, and the stirring time of a batch in a tank. For both, the equipment configuration and simulation parameters, a tabular format is employed, for simulation results tabular as well as graphical formats are employed. Equipment utilization, throughput, manufacturing lead time

and work in process are depicted, all as function of time. In addition, simulation results are depicted in a Gantt chart. The number of simulated batches, the workload per batch, the makespan, and the average equipment utilization are shown in a table.

To illustrate its use, the design tool is applied to determine the preparation equipment for Riedel's product group *Carbonated drinks*.

Riedel example

The product group Carbonated drinks is composed of six blends which are packed in three different packaging types. Suppose that a maximum of two blends is packed simultaneously. When this is translated to the representation of Figure 5.7, the number of *pumps out* is two. The tanks, ingoing pumps and the CIP circuits have to take care of feeding these two pumps with batches of carbonated drinks in such a way, that the equipment downstream can be utilized for 100%. A blend is packed in three packaging types simultaneously: two cans of 250 ml. and one of 1000 ml. As a result, the maximum flow generated by an outgoing pump equals 17 thousand liters per hour. Carbonated drinks are prepared by in-line dilution, the concentrated blend in the preparation tank has one-fifth of the volume of the end product. The ingredients are sugar sirup and a number of fruit concentrates purchased in drums. The ingredients are discharged with 15 thousand liters per hour. Every time a pump or tank has been used, it is cleaned with the CIP circuit. A cleaning procedure takes 0.5 hour, and a CIP circuit can handle one cleaning action at a time. On average, the batch size of a carbonated product is expected to be 300 thousand liters of single strength product.

After simulating a few scenarios, a hardware configuration is found that satisfies the needs. In Figure 5.9 and 5.10 the simulation results of this scenario are depicted.

The configuration is composed of one ingoing pump, three receiver tanks of 70 m³, two outgoing pumps, and one CIP circuit. Twenty batches of 300 thousand liters have been simulated. In the Gantt chart of Figure 5.10 the operations necessary for the production of one product are marked with a letter. For example, the production of product *A* starts with filling *tank 1* with ingredients using *pump in*. This operation takes four hours and uses *pump in* and *tank 1* synchronously. After that *pump in* is cleaned with the CIP circuit, which takes thirty minutes. At the same time the ingredients are stirred in *tank 1* for three hours. At time 7 the stirring is completed and *tank 1* is emptied with *pump out 1*. Finally, *tank 1* and *pump out 1* are cleaned

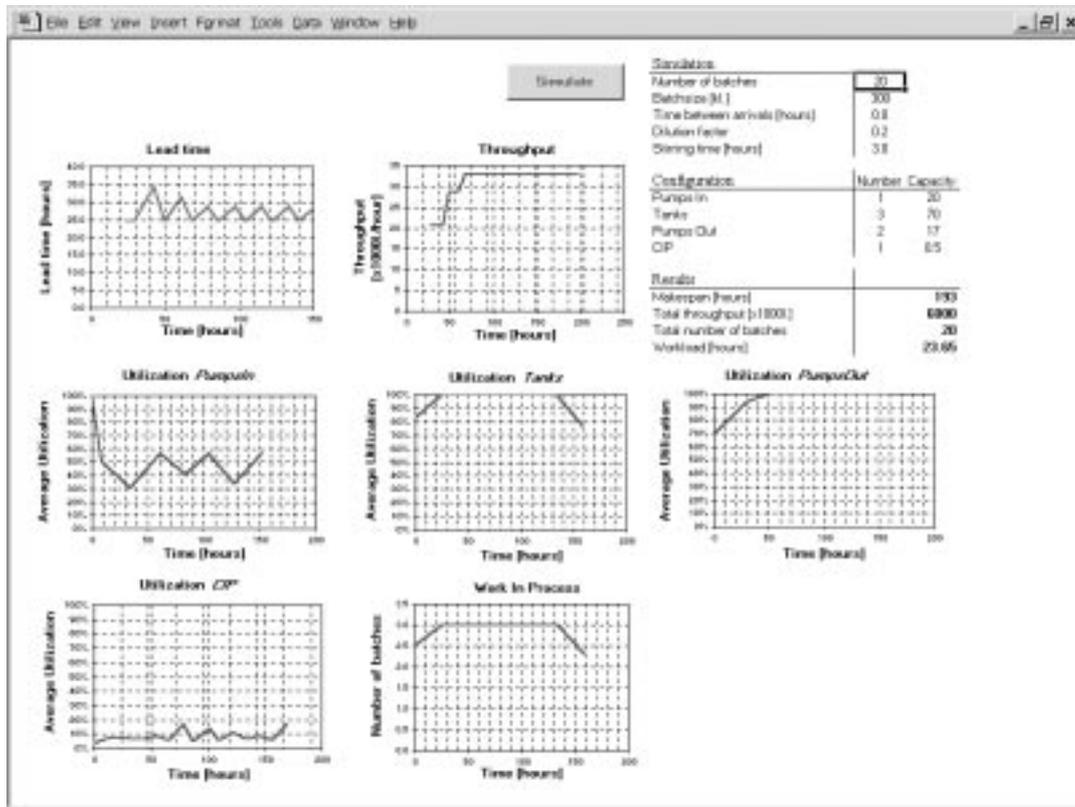


Figure 5.9: Interface of the capacity analysis tool.

with the CIP circuit and both resources are available again, next *tank 1* and *pump out 1* proceed with product *D* and product *C*, respectively.

Figure 5.9 shows that each batch has a workload of 23.65 hours, the average lead time of a batch is 27 hours. Gantt chart 5.10 shows that after 30 hours, the throughput of the system settles at 33.1 thousand liters per hour, the system has reached a steady state. The utilization of the tanks is 100% in steady state, which means that a tank never waits empty. The Gantt chart shows that in steady state a tank waits for 6 hours every other batch for an outgoing pump to become available. The utilization of the outgoing pumps, and in addition downstream equipment such as pasteurizers and packaging lines, is 100%. The CIP circuit and the ingoing pump provide enough capacity in this configuration: their utilization is low.

Now, necessary processing equipment downstream is determined. Carbonated products contain a small amount of carbon dioxide. This is injected after diluting the concentrate with an in-line blender. A buffer tank is used to damp the fluctuating

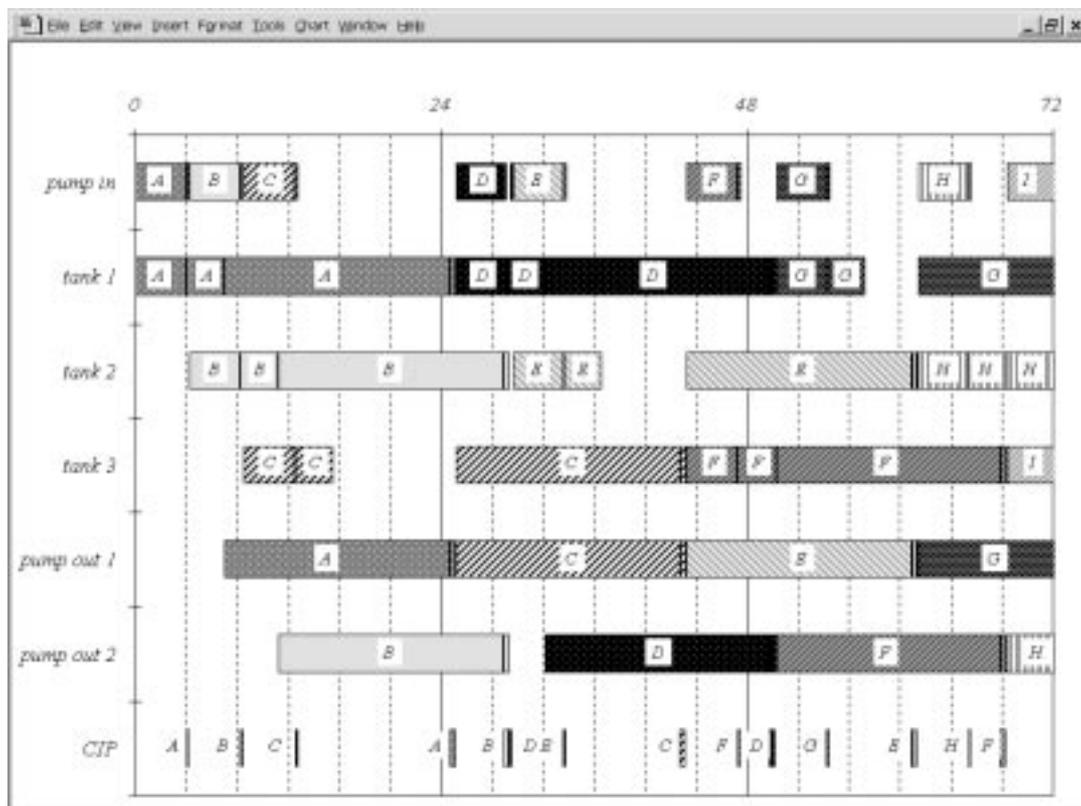


Figure 5.10: Gantt chart depicted by the capacity analysis tool.

output of the blender. The blender needs to generate an output of at least 17 thousand liters per hour. The size of the buffer tank depends on the detailed specifications of the blender. The buffer tank is equipped with a carbon dioxide blanket to maintain the gas concentration in the product. For every outgoing pump a blender, carbon dioxide injector, and a buffer tank is installed.

The configuration presented above provides the necessary hardware to produce two carbonated blends simultaneously. Capacity of tanks and downstream equipment is nicely balanced. In Figure 5.11 the complete configuration is depicted.

□

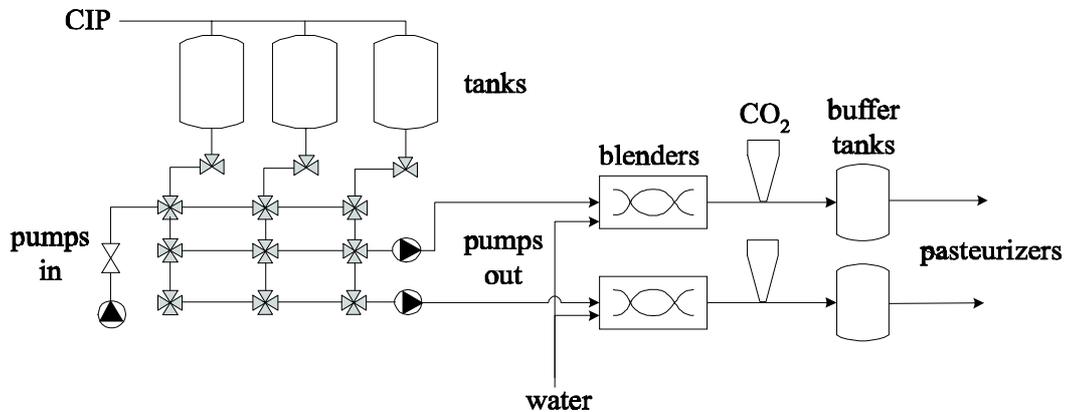


Figure 5.11: Hardware configuration for preparing carbonated products.

5.3 Layout

In accordance with capacity analysis, constitution of the factory layout is divided into two parts: determining a floor plan for pasteurizing and packaging equipment, and for preparation equipment. In this section a global layout pattern is determined.

Pasteurizing and packaging equipment

Packaging equipment consists of a filling machine, cap or straw applicator, tray packer and palletizer. Compared with the preparation equipment, the pasteurizing and packaging equipment comprise a simple flow line structure. The pasteurizer precedes the packaging equipment and feeds on or more packaging lines. With respect to financial investments, the filling machine represents the heart of the packaging equipment and is placed first in line. The cap or straw applicator, tray packer and palletizer are located downstream, in that sequence. The filling machine is failure sensitive and has to be the bottleneck capacity in the packaging line. A conveyor system transports the cartons downstream, and at the same time forms buffer capacity. In this thesis we will not discuss the size of these buffers.

Preparation equipment

In Chapter 4 the architecture for the Preparation Department has been designed. To reduce material flow complexity a group technology structure is applied. The product

range is divided in six groups and for each of these groups the necessary preparation equipment is selected. Literature presents three common group technology layout patterns: the GT flow line, the GT cell, and the GT center [Buf87]. GT flow line is closely related to the flow shop, and GT center is closely related to the job shop; the GT cell concept provides a compromise between the GT flowline and the GT center.

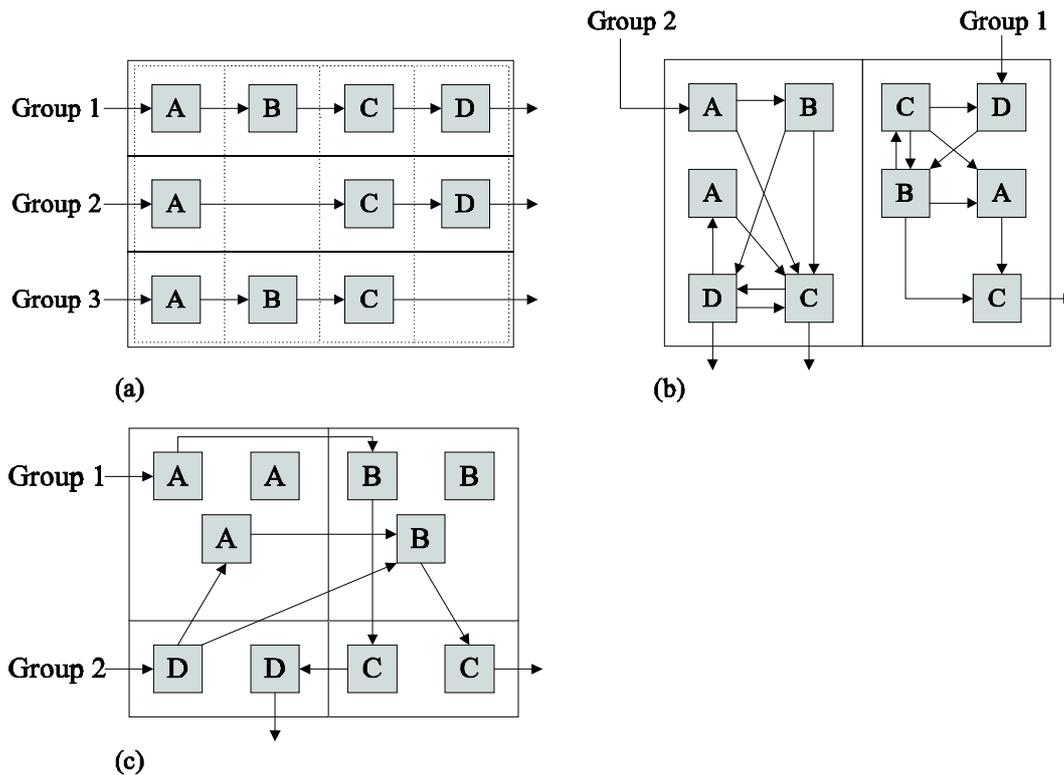


Figure 5.12: (a) GT flow line layout, (b) GT cell layout and (c) GT center layout.

The GT flow line pattern is applicable when each product or product group follows nearly the same processing route. Manufacturing equipment necessary for the production of one group is placed in-line (see Figure 5.12a), but, in contrast with a flow shop, equipment of the same type is located in processing centers. When the product range changes, products are easily re-routed without changing the floor plan.

The GT cell is applicable when the processing route for families of parts differs. The operations necessary for the completion of one or more part families are executed in a cell, which contains the necessary equipment (see Figure 5.12b). The GT center (also called *virtual GT cells*) has a floor plan similar to that of a job shop (see

Figure 5.12c), but the allocation of operations is performed according to a GT philosophy. This means that the routing of parts in a family is such, that they are processed by the same equipment as much as possible.

In Chapter 4, it has been mentioned that in the production of canned fruit juices a clear flow line can be distinguished. There is relatively little backflow, and every processing equipment is visited once. Riedel products follow nearly the same route. Figure 4.8 shows that the ordering of processes for every product group is equal, for example, blending always precedes filtering and filtering always precedes buffering. It is concluded that the GT flow line pattern is best qualified for Riedel's Preparation Department. Utilities, however, cannot be incorporated in one of the groups, and are placed separately. The product life-cycle of Riedel products tend to shorten, therefore, the processing equipment is located in processing centers, which provides the necessary flexibility: these functional processing centers enable easy reallocation of processing equipment to another product group.

5.4 Conclusion

In this chapter the architecture of Chapter 4 has been quantified. This capacity analysis problem has been partitioned into two: first, the packaging and pasteurizing resources are determined, then the preparation resources. The packaging and pasteurizing equipment are calculated analytically, where the preparation equipment is calculated by means of dynamic simulation. To support determining the necessary pasteurizing and packaging equipment, a nine-step procedure has been developed. For every packaging type, this procedure is executed once. For the preparation equipment a design tool has been designed that provides the means to calculate the necessary equipment per product group. The graphical interface and embedded simulation model enable fast evaluation of scenario's. The final task in the resource design phase is constituting a layout. It is determined that a GT flow line pattern is best fit for laying out Riedel's preparation equipment. The pasteurizing and packaging equipment is laid out in-line. In the next chapter detailed information about the operation of the plant is introduced. The influence of scheduling related variables on the performance of the plant is analysed and it is checked whether the resources determined in this chapter provide sufficient capacity.

Chapter 6

Operations

In the previous chapter a list of resources has been determined and a floor-plan has been constituted. The objective of the operational design phase is refining the design of resources, and designing the control system. The result is an initial design which triggers the realization phase. The realization phase comprises the building and testing of the industrial system.

Completion of the initial design does not terminate the design process. On the contrary, in today's highly developed markets, companies increasingly face erratic consumer behavior and reduced prediction horizons. Together with technological developments these aspects have a large impact on the operation of a plant. Therefore, continuation of the design process during the utilization phase is essential to keep up with or outperform the competition. To support this iterative design activity, an operational design tool has been developed.

Shah [Sha95] states that the design and operation of multipurpose batch and semi-continuous plants are severely complicated by the need to take account of detailed production scheduling. The same accounts for the operational design of Riedel's production facility. Scheduling related variables such as delivery restrictions of raw materials, product specific allocation constraints, change-over times, and preparation procedures, have a significant impact on the performance of the plant. When refining the resource design, these variables have to be taken into account. To incorporate the effect of scheduling related variables into the design process, a scheduling procedure has been designed. This procedure should be seen as a first attempt to solve the severely complicated scheduling problem for Riedel's production facility.

To break down the complexity of the scheduling problem, it is decomposed into two parts. One part takes care of pasteurizing and packaging scheduling, the other part takes care of preparation scheduling. These two parts cannot be considered indepen-

dently: the pasteurizing and packaging schedule directly influences the preparation schedule and vice versa. However, to reduce computational complexity further, the interaction between the two parts is considered to be a one way street: the packaging and pasteurizing schedule influences the preparation schedule, not the other way around. As a result, a pasteurizing and packaging schedule is generated first. Then an instance of the preparation scheduling problem is generated, and subsequently, a preparation schedule is computed.

Section 6.1 involves scheduling of pasteurizing and packaging equipment. Section 6.2 deals with the scheduling of preparation equipment. Both scheduling problems are described and for each problem an algorithm has been designed. These algorithms have been integrated in a design tool that is used to refine the resource design. This refinement is described in Section 6.3. Section 6.4 deals with designing the control system. In Section 6.5 concluding remarks are presented.

6.1 Pasteurizing and packaging scheduling

Below, the scheduling problem for the pasteurizing and packaging equipment is described. The scheduling problem is considered static and deterministic, in other words, all characteristics of the operations and resources are known in advance.

Informal problem description

The Packaging Department consists of several machine clusters, where each cluster contains one or more packaging lines. All packaging lines in a cluster produce the same type of carton. For the majority of products, the lines in a cluster are interchangeable, however, exceptions have to be dealt with. Each line consists of a filling machine, a cap or straw applicator, a tray packer and a palletizer. Although a carton is processed by each of these machines, the complete packaging of a product is considered as one operation. A pasteurizer has been installed in front of the filling machine and in general this pasteurizer feeds two identical packaging lines. As a consequence, these coupled packaging lines always produce the same product. Therefore, together with the pasteurizer, they are considered as one resource with doubled capacity. One batch in the Preparation Department is often split in multiple packaging batches. For quality reasons, the lead time of the batch in the Preparation Department, the non-aseptic part of production, needs to be minimized. Therefore, the packaging operations of a blend on different lines need to be synchronized. A synchronization between two operations is successful when the largest operation completely over-

laps the smallest operation. The capacity of a resource is not constant over time, preventive maintenance can reduce the capacity for a certain period of time.

The Pasteurizing and Packaging Scheduling Problem (PPSP) is now defined as the problem of assigning resources to operations over time. A resource is not allowed to process more than one operation at a time. Furthermore, in between two operations, a sequence dependent setup time needs to be taken into account. The capacity of a resource can be reduced temporarily due to preventive maintenance actions. The set of operations is subdivided in operations for which preemption is allowed, and operations for which preemption is not allowed. When preemption is allowed, a packaging operation can be split in multiple packaging operations. For each operation a set of feasible resources is available. An operation is not allowed to start before its release date or finish after the end of the scheduling horizon. The objective is to generate a feasible schedule that minimizes:

- total tardiness,
- sum of sequence dependent setup times,
- number of missed synchronizations.

Formal problem description

Definition 6.1 Formally, an instance *PPSP* of the considered Packaging and Pasteurizing Scheduling Problem is defined as a thirteen tuple

$$PPSP = (\mathcal{OP}, \mathcal{ON}, \mathcal{R}, \mathcal{RS}, fr, sz, dd, rd, cp, pm, co, sy, H),$$

where \mathcal{OP} is a set of operations for which preemption is allowed, \mathcal{ON} is a set of operations for which preemption is not allowed, \mathcal{R} a set of resources, and $\mathcal{RS} \subseteq \mathcal{P}(\mathcal{R})$ a set of resource sets. For notational convenience we define $\mathcal{O} = \mathcal{OP} \cup \mathcal{ON}$. For each operation there is set of feasible resources, given by function $fr: \mathcal{O} \rightarrow \mathcal{RS}$; a size, given by function $sz: \mathcal{O} \rightarrow \mathbb{N}$; a due date, given by function $dd: \mathcal{O} \rightarrow \mathbb{R}$; and a release date, given by function $rd: \mathcal{O} \rightarrow \mathbb{R}$. Each resource has a maximum capacity (number of cans per hour), defined by $cp: \mathcal{R} \rightarrow \mathbb{N}$, and for each resource there is a set of preventive maintenance actions, determined by function $pm: \mathcal{R} \rightarrow \mathcal{P}(\mathbb{R} \times \mathbb{R} \times \mathbb{N})$ where a preventive maintenance action is a three tuple of two reals representing start and end time, and a nonnegative integer representing the capacity of the resource during the action. Partial function $co: \mathcal{O} \times \mathcal{O} \times \mathcal{R} \mapsto \mathbb{R}$ gives the sequence dependent change-over time for each combination of operations. A change-over time is only defined for feasible resources, i.e., $\text{dom}(co) = \{(o, o', r) \mid o, o' \in \mathcal{O}, r \in fr(o) \cap fr(o')\}$.

Non-preemptive packaging operations that process the same blend but use different types of packaging, are often combined in the preceding Preparation Department. To minimize the lead time of the blend in the Preparation Department, these packaging operations need to be synchronized. Partial function $sy: \mathcal{ON} \mapsto \mathcal{ON}$ defines the synchronization partner of each operation that is synchronized. The lead time is minimized when the largest synchronization partner completely overlaps the smallest. $H \in \mathbb{R}$ is the scheduling horizon. \square

Preemption is allowed for all operations $o \in \mathcal{OP}$. Suppose an operation $o \in \mathcal{OP}$ is split in n operations $o_0 \dots o_{n-1}$, then

$$\sum_{0 \leq i \leq n-1} sz(o_i) = sz(o). \quad (6.1)$$

Furthermore, for each $o_i \in \{o_0, \dots, o_{n-1}\}$ it holds that function $fr(o_i) = fr(o)$, $dd(o_i) = dd(o)$, and $\forall o' \in \mathcal{O}, \forall r \in \mathbb{R} \Rightarrow co(o', o_i, r) = co(o', o, r) \wedge co(o_i, o', r) = co(o, o', r)$. In discrete manufacturing often an additional constraint is formulated, which states two operations $o_i, o_j \in \{o_0, \dots, o_{n-1}\}$ are not allowed to be processed at the same time, because a single product cannot be present at two resources at the same time. In the case that production deals with large series of discrete products, liquids, or gasses this is not a problem.

Definition 6.2 Schedule s for instance $PPSP$ is a function $s: \mathcal{O} \rightarrow \mathcal{R} \times \mathbb{R}$, which for each operation defines a tuple containing the assigned resource and the start time. The set of all schedules for instance $PPSP$ is denoted by \mathcal{S}_{PPSP} . \square

Definition 6.3 Let $s \in \mathcal{S}_{PPSP}$ be a schedule for instance $PPSP$. We define for schedule s four functions that are used below. Function $ra_s: \mathcal{O} \rightarrow \mathcal{R}$, that defines for each operation the resource assignment of schedule s . For each $o \in \mathcal{O}$, $ra_s(o) = \pi_1(s(o))$, where π_1 denotes a function that returns the first element of tuple $s(o)$. Function $st_s: \mathcal{O} \rightarrow \mathbb{R}$, that for each operation defines the start time. For each $o \in \mathcal{O}$, $st_s(o) = \pi_2(o)$. Function $ct_s: \mathcal{O} \rightarrow \mathbb{R}$, which for each operation defines the completion time. The completion time of an operation is determined by its start time, the assigned resource and its maximum capacity, and the set of preventive maintenance actions for that resource. Function ct_s is not further specified here. Furthermore, we define function $ad_s: \mathcal{O} \times \mathcal{O} \mapsto \{\text{true}, \text{false}\}$, which for each pair of operations in schedule s defines whether or not they are scheduled adjacent. Two operations are adjacent if no operation assigned to the same resource is scheduled in between. Function ad_s is only defined for a pair of nonidentical operations assigned to the same resource, i.e., $\text{dom}(ad_s) = \{(o, o') \mid o, o' \in \mathcal{O}, o \neq o', ra_s(o) = ra_s(o')\}$. Function ad_s is specified as:

$$\begin{aligned}
ad_s(o, o') &= \{ o'' \mid o'' \in \mathcal{O} \wedge ra_s(o'') = ra_s(o) \\
&\quad \wedge (st_s(o'') \geq ct_s(o) \wedge ct_s(o'') \leq st_s(o')) \\
&\quad \vee st_s(o'') \geq ct_s(o') \wedge ct_s(o'') \leq st_s(o) \} \\
&= \emptyset.
\end{aligned} \tag{6.2}$$

□

A schedule s is feasible if:

- An assigned resource is a feasible resource, i.e., for all $o \in \mathcal{O}$,

$$ra_s(o) \in fr(o). \tag{6.3}$$

- No two adjacent operations assigned to the same resource are processed at the same time (precedence relation), and a change-over time is taken into account, i.e., for all $o, o' \in \mathcal{O}$,

$$\begin{aligned}
ad_s(o, o') &\Rightarrow ct_s(o) + co(o, o', ra_s(o)) \leq st_s(o') \\
&\quad \vee ct_s(o') + co(o', o, ra_s(o)) \leq st_s(o).
\end{aligned} \tag{6.4}$$

- No operation is started before its release date, and completed after the end of the scheduling horizon, i.e., for all $o \in \mathcal{O}$,

$$st_s(o) \geq rd(o) \wedge ct_s(o) \leq H. \tag{6.5}$$

Definition 6.4 The quality of every feasible schedule $s \in \mathcal{S}_{PPSP}$ is evaluated with a cost function $f_s: \mathcal{S}_{PPSP} \rightarrow \mathbb{R}$. Function f_s is composed of three performance measures: the total tardiness $t_s: \mathcal{S}_{PPSP} \rightarrow \mathbb{R}$, the sum of sequence dependent setup times $sds_s: \mathcal{S}_{PPSP} \rightarrow \mathbb{R}$, and the number of unsuccessful synchronizations $nsy_s: \mathcal{S}_{PPSP} \rightarrow \mathbb{N}$. The total tardiness t_s of a schedule $s \in \mathcal{S}_{PPSP}$ is given by

$$t_s(s) = \sum_{o \in \mathcal{O}} \max(ct_s(o) - dd(o), 0.0). \tag{6.6}$$

The sum of sequence dependent setup times sc_s is given by

$$sds_s(s) = \sum_{o, o' \in \mathcal{O} \wedge ad_s(o, o')} co(o, o', ra_s(o)). \tag{6.7}$$

The number of missed overlaps is given by

$$\begin{aligned}
 nsy_s(s) = \text{size} (\{ o \in \mathcal{ON} & \tag{6.8} \\
 | ct_s(sy(o)) < ct_s(o) \wedge st_s(sy(o)) < st_s(o) & \\
 \vee ct_s(o) < ct_s(sy(o)) \wedge st_s(o) < st_s(sy(o)) & \\
 \}), & \tag{6.9}
 \end{aligned}$$

where the function size returns the number of elements in a set. A synchronization is successful when the largest operation completely overlaps the smallest. Now, function f_s is given by

$$f_s(s) = t_s(s) + sds_s(s) + nsy_s(s). \tag{6.10}$$

□

The objective is to find a feasible schedule $s \in \mathcal{S}_{PPSP}$ that minimizes f_s .

Computational complexity

Rinnooy Kan et al. [Rin75] have designed an enumerative algorithm that is tested for a range of single machine weighted tardiness problems. Their main conclusion is that ‘the problem of minimizing total weighted tardiness remains a very difficult one’. Du and Leung [Du90] have shown that the problem of minimizing total tardiness on one machine is NP-hard. For a survey of total tardiness scheduling literature, we refer to Koulamas [Kou94]. If the single machine problem is NP-hard, the PPSP must also be NP-hard. To solve instances of NP-hard problems, in the worst case the solution space has to be enumerated completely in order to guarantee that the best solution found is the optimal solution. The size of the solution space of an NP-hard problem grows exponentially with the size of the instance. Consequently, it cannot be guaranteed that an optimal solution is found within a reasonable amount of time.

Solving the PPSP

As the PPSP is NP-hard, it is likely that an optimal solution cannot be found within a reasonable amount of time. However, this is not considered a serious drawback. Because of erratic consumer behavior accurate information about future developments is not available. Solving an instance of the PPSP to optimality is hardly worth the effort, if the instance is likely never to become reality. To support the process of refining the design of resources, it is important that solutions are generated rapidly, in order to be able to evaluate a complete set of instances. The quality of the solutions has to be ‘good’ rather than optimal.

In case near-optimal solutions for complex scheduling problems have to be generated in a reasonable time, approximation algorithms are often applied successfully. Literature distinguishes two classes of approximation algorithms: constructive algorithms and local search algorithms.

A constructive algorithm generates a solution by iteratively extending a partial solution until a full solution is obtained [Nui94]. Examples of approximation algorithms are constraint guided search, priority dispatching, random sampling and probabilistic dispatching. For the latter two we refer to [Bak74] and [Fre82].

When a constraint guided search algorithm is applied, the problem is modeled as a special case of the constraint satisfaction problem (CSP) [Mon74]. To solve an instance of the CSP, generally applicable algorithms are available, an overview is presented by Kumar [Kum92]. The PPSP can be modeled as an extension of the Time and Resource Constrained Scheduling Problem (TRCSP) formulated by [Nui94], where the TRCSP is a special case of the CSP. Where the TRCSP is a search problem, the PPSP is an optimization problem, i.e., the PPSP uses an objective function to evaluate the quality of a solution. Furthermore, the PPSP considers a combination of preemptive and non-preemptive orders, sequence dependent setup times, and resource capacity that is not constant over time. Nuijten [Nui94] has designed algorithms that can handle the TRCSP. These algorithms have been implemented in a commercially available optimization engine, known as the ILOG solver. There is one drawback of using a commercially available solver to refine the resource design: it prevents exploiting problem-specific knowledge. Especially in the design phase, gaining insight in the problem at hand is regarded very important. Therefore, it has been decided to develop and implement an home-made algorithm.

A priority dispatching algorithm uses some priority rule to select and dispatch operations. A large number of priority rules have been developed, Panwalker and Iskander [Pan77], and Haupt [Hau89] present a survey. Peters [Pet99] has designed a priority dispatching algorithm for minimizing the total tardiness of operations with release dates for multiple resources. This algorithm is a modified version of the Insertion Priority Rule for Total Tardiness criterion (IPRTT) proposed by Chu and Portmann [Chu92]. In the sequel, this modified version of the IPRTT will be referred to as MIPRTT. The original IPRTT algorithm of Chu and Portmann has been designed for sequencing operations on a single machine, where each operation has a due date and a release date. Its objective is to generate a schedule that minimizes the total tardiness. The algorithm has been modified to perform its task for multiple machines. The MIPRTT algorithm is based on the function $prtt: \mathbb{R} \times \mathcal{O} \rightarrow \mathbb{R}$. The function $prtt$ of operation $o_i \in \mathcal{O}$ at time t is defined as:

$$prtt(t, o_i) = \max(r_i, t) + \max(\max(r_i, t) + p_i, d_i),$$

where p_i is the processing time of operation o_i , r_i its release date, and d_i its due date. The function $prtt$ is a priority rule: the smaller the $prtt$ value of an operation at a given time, the higher its priority. Its value equals the sum of the earliest start time $\max(r_i, t)$ and a modified due date $\max(\max(r_i, t) + p_i, d_i)$. The MIPRTT algorithm is treated later on in this section.

A local search algorithm starts its search from an initial solution and explores the solution space through iteratively modifying the current solution. In every iteration, a new solution is chosen from a set of neighboring solutions. The neighborhood of the current solution is generated with a neighborhood function, which defines what modifications or *moves* are allowed. Neighboring solutions are evaluated with a cost function, and the search strategy defines which from the neighboring solutions is selected as starting point for the next iteration. When a stop criterium is met, the search is terminated and the best solution is returned. Examples of local search algorithms are: iterative improvement, threshold accepting, simulated annealing, tabu search, and genetic algorithms.

Iterative improvement is the most primitive of all local search algorithms. Iterative improvement starts from a given initial solution and explores its neighborhood. When an improvement is found, it is selected as starting point for the next iteration, otherwise the algorithm terminates and returns the last solution, which is by definition a local minimum. To avoid getting stuck in local minima, other local search algorithms have more advanced search strategies, which enable selection of worse quality solutions to escape from local minima. Genetic algorithms differ from the other local search algorithms as they use a population of solutions instead of a single solution as starting point for each iteration. A new solution is created when two solutions are combined with a crossover operation.

Flexibility and ease of implementation of local search algorithms have led to the successful handling of many complex real-world problems [Aar97]. Anderson et al. [And97] have compared the performance of different local search algorithms on machine scheduling problems. Genetic algorithms tend to perform worse than other local search algorithms unless problem specific knowledge is incorporated. Anderson et al. find that tabu search algorithms are among the most competitive algorithms, although no firm conclusions can be drawn. Therefore, a tabu search algorithm is designed to solve the PPSP.

Tabu search algorithms avoid getting stuck in local minima by selecting the best neighboring solution, even if it is worse than the current solution. In order to prevent cycling tabu search algorithms administrate solutions that have been visited recently. These solutions are stored in a tabu list. The algorithm is not allowed to select a solution that is listed in the tabu list. To improve efficient use of computer memory, solutions are listed indirectly by storing forbidden moves instead of com-

plete solutions. The tabu list has a limited length, when the list is full, the oldest element is removed. A disadvantage of storing moves instead of solutions is that a tabu move may lead to an unvisited solution. To prevent that high quality solutions are missed, a tabu move is forbidden, unless it leads to a solution that satisfies some predefined condition, often called aspiration criterion. A tabu search algorithm terminates when a predefined stop criterion is met, and returns the best solution found. For more information on tabu search we refer to Glover [Glo89], [Glo90] and Glover et al. [Glo93].

In Figure 6.1 a procedure is presented to solve the PPSP problem. Here as well as in the rest of this chapter, procedures are presented in a pseudo-code. To reduce the complexity of the scheduling problem, the production lines are scheduled groupwise. First, the machine group $g \in \mathcal{G}$ with the highest workload is selected with the function *highest utilization*. For each operation $o \in \mathcal{O}$, the corresponding machine group is determined by the function *machinegroup*. Subsequently, all the operations are selected that have to be assigned to one of the machines in this group g . These operations are scheduled with procedure SCHEDULE MACHINE GROUP. With the resulting machine group schedule s , the due dates and release dates of the remaining operations are modified with the function *modify*. The schedule s is added to the set of machine group schedules \mathcal{S} , and the next machine group is selected. This procedure is repeated until all machine groups are scheduled.

```

proc SCHEDULE PASTEURIZING PACKAGING
   $\mathcal{G} ::=$  'set with all machine groups';
   $\mathcal{S} ::= \emptyset$ ;
  while  $\mathcal{O} \neq \emptyset$  do
     $g ::=$  highest utilization( $\mathcal{O}, \mathcal{G}$ );
     $\mathcal{O}' ::= \{o \mid o \in \mathcal{O}, \text{machinegroup}(o) = g\}$ ;
     $\mathcal{O} ::= \mathcal{O} - \mathcal{O}'$ ;
    SCHEDULE MACHINE GROUP( $\mathcal{O}'$ );
     $\mathcal{O} ::=$  modify( $s, \mathcal{O}$ )
     $\mathcal{S} ::= \mathcal{S} \cup \{s\}$ 
  end while
endproc

```

Figure 6.1: The SCHEDULE PASTEURIZING PACKAGING procedure.

Procedure SCHEDULE MACHINEGROUP is a scheduling procedure in five steps, see Figure 6.2. The core of the procedure is a local search algorithm called TABU SEARCH. In the implemented scheduling procedure the tabu search algorithm is applied to schedule the non-preemptive operations. These non-preemptive operations, collected

in set \mathcal{ON} , are selected with the boolean function *preemption*. The initial schedule s_{init} for these non-preemptive operations is calculated with the MIPRTT algorithm. The schedule s_{init} is used as input for the TABU SEARCH algorithm, that optimizes the initial schedule and returns a schedule s . Finally, the preemptive operations \mathcal{OP} are inserted in schedule s , with the function *insert*.

```

proc SCHEDULE MACHINEGROUP( $\mathcal{O}'$ )
   $\mathcal{ON} := \{o \mid o \in \mathcal{O}', \neg \text{preemption}(o)\}$ ;
  MIPRTT( $\mathcal{ON}$ );
  TABU SEARCH( $s_{init}$ );
   $\mathcal{OP} := \{o \mid o \in \mathcal{O}', \text{preemption}(o)\}$ ;
   $s := \text{insert}(s, \mathcal{OP})$ ;
endproc( $s$ )

```

Figure 6.2: The SCHEDULE MACHINE GROUP procedure.

The MIPRTT algorithm starts with a set of packaging lines and a set of non-preemptive operations that need to be scheduled on one of the packaging lines. The total workload of the non-preemptive operations is calculated, and subsequently, the number of necessary resources is estimated. From the set of resources the necessary resources are selected, preferring the ones without preventive maintenance actions. Then the operations are assigned arbitrarily to these resources, such that the workload is divided equally and resource constraints are met. The resources are now scheduled one after another. The procedure iteratively schedules each of the operations, until all operations assigned to a resource have been scheduled, or the completion time of the last operation exceeds the end of the planning period. At each iteration, the operation with smallest *prtt* value is definitively scheduled, ties being broken by choosing first the operation with the smallest number of legal resources, and second the operation that can be scheduled the earliest. Every time an operation is scheduled, it is checked if there exist unscheduled operations that can be inserted in the idle time before o . If there exist such operations, the operation with the smallest start time is scheduled, breaking ties by choosing the operation with the smallest *prtt* value, until there is no job left that can be inserted. In between two operations a sequence dependent setup time is inserted. If there are operations that cannot be scheduled without exceeding the planning horizon, these operations are scheduled on an empty resource which initially has not been selected, or, if such resource is not available, iteratively added to the resource with the smallest completion time. When all operations are scheduled, the procedure terminates. The MIPRTT algorithm is developed to minimize total tardiness on one machine, but is not equipped to optimize the sequence dependent setup times, let alone the PPSP. However, this procedure provides an initial schedule for the tabu search algorithm within a few seconds.

A tabu search algorithm has to be tailored to the details of the problem at hand. Unfortunately, there is little theoretical knowledge that guides this tailoring process [Aar97]. The TABU SEARCH algorithm implemented to solve the PPS is a best improvement algorithm: all neighboring solutions are generated and subsequently the best solution is selected. The objective of the tabu list is to prevent cycling. When the size of the tabu list is too small, cycling is not prevented, but on the other hand, when it is taken too large, it restricts the search. However, it is often impossible to determine a tabu list size that meets both aspects [Her97]. Laguna and González Velarde [Lag91] propose a tabu search algorithm for the multi-machine weighted earliness problem with deadlines, and use a tabu list length of $\lfloor \sqrt{n} \rfloor + 5$, where n is the number of operations, and $\lfloor \sqrt{n} \rfloor$ denotes the smallest integer larger or equal to \sqrt{n} . In the TABU SEARCH algorithm presented here, the size of the tabu list is set to $\lfloor n/2 \rfloor$, where n is the number of operations to be scheduled on a machine group. Peters [Pet99] has performed experiments, which have shown that this is an acceptable size. The TS algorithm terminates when no neighboring solutions are found, or the number of successive iterations in which no best solution ever is determined, exceeds a specified maximum.

The tabu search algorithm is depicted in Figure 6.3. It starts with an initial solution s_{init} . First, a number of variables is initialized. The walk through the solution space is started with the generation of a set of possible moves \mathcal{M} . If set \mathcal{M} is nonempty, a random move m' is selected ($m' := pick\mathcal{M}$) and the cost of the corresponding neighboring solution is calculated ($f' := cost(s \oplus m')$, where $s \oplus m'$ denotes applying move m' to schedule s); in the first iteration, s equals s_{init} . Subsequently, all moves in \mathcal{M} are evaluated. A move m is accepted ($b := true$), when the cost of the corresponding schedule is less than f' , and m is not an element of the tabu list ($m \notin \mathcal{TL}$). However, when m is tabu, but applying the move leads to a solution that is better than any solution found so far, we want to accept it. Hence, when the cost of the corresponding schedule f , satisfies the aspiration criterium $f < f_{best}$, move m is accepted. If the complete neighborhood is evaluated ($\mathcal{M} = \emptyset$), the best move m' is applied to generate a new current solution ($s := s \oplus m'$). To prevent returning to the previous solution, the inverse of move m' is added to the tabu list ($\mathcal{TL} := \mathcal{TL} \cup \{inverse(m')\}$). When the size of the tabu list exceeds a maximum ($size(\mathcal{TL}) > maxsize$), the oldest element is removed ($\mathcal{TL} := tail(\mathcal{TL})$). The new current solution may have a cost value that is less than the best solution so far, in that case it is stored in s_{best} , furthermore, f_{best} is modified, and the number of iterations is reset ($iter := 0$). When the number of iterations $iter$ exceeds a specified maximum $itermax$, the algorithm terminates and returns the best solution so far s_{best} .

An important part of the tabu search algorithm is generation of neighboring solutions. The function *moves* defines the set of moves that are allowed to generate neighboring

```

proc TABU SEARCH( $s_{init}$ )
   $s := s_{init}; s_{best} := s_{init}; f_{init} := cost(s_{init});$ 
   $f_{best} := f_{init}; \mathcal{TL} := \emptyset; iter := 0;$ 
  repeat
     $\mathcal{M} := moves(s);$ 
    if  $\mathcal{M} \neq \emptyset$  then  $m' := pick \mathcal{M}; f' := cost(s \oplus m');$ 
      repeat
         $m := pick \mathcal{M}; \mathcal{M} := \mathcal{M} - \{m\};$ 
         $f := cost(s \oplus m);$ 
        if  $f < f'$  then
          if  $m \notin \mathcal{TL}$ 
            then  $b := true$ 
            else  $b := f < f_{best}$ 
          endif;
          else  $b := false$ 
          endif;
          if  $b$  then  $m' := m; f' := f$  endif;
        until  $\mathcal{M} = \emptyset;$ 
         $s := s \oplus m'; iter := iter + 1;$ 
        if  $f' < f_{best}$  then  $s_{best} := s; f_{best} := f'; iter := 0$  endif;
         $\mathcal{TL} := \mathcal{TL} \cup \{inverse(m')\};$ 
        if  $size(\mathcal{TL}) > maxsize$  then  $\mathcal{TL} := tail(\mathcal{TL})$  endif;
      endif;
    until  $iter \geq itermax;$ 
  endproc( $s_{best}$ )

```

Figure 6.3: The TABU SEARCH algorithm.

solutions. In the tabu search algorithm a composite neighborhood is applied. Three neighborhoods are combined:

- Swap: interchange two operations assigned to different resources by reversing their resource assignment;
- Transpose: interchange two adjacent operations on the same resource;
- Reassign: assign an operation to another resource.

Some of the moves are expected to generate solutions with high cost and are excluded from the set of moves. For example: swapping two operations o_1 and o_2 , when the

start time of o_2 in the current schedule is smaller than the release date of o_1 will very likely generate a poor quality neighbor. Furthermore, reassigning an operation to an empty resource is not allowed.

The efficiency of a local search algorithm benefits from simple calculations in each step. Unfortunately, the calculation of cost values requires intricate computations. As a result, the number of neighborhood solutions that can be visited is limited. Hertz et al. [Her97] indicates that calculation of the change of the cost value induced by a move is often easier than calculation of the absolute cost value. Laguna & González Velarde [Lag91] refer to this change of cost as *move value*. Although the implementation presented here likely will benefit from this approach, it is not yet implemented.

Computational results

The scheduling procedure is implemented with the specification formalism χ . Peters [Pet99] has performed experiments in order to provide a first impression of the performance of the scheduling algorithm. The algorithm has been fed with real life data. The generated tabu search schedules have been compared with the schedules constructed by Riedel's human scheduler. The schedules have been evaluated with the minimization criteria: total tardiness, sum of sequence dependent setup times ($\sum sds$) and number of missed synchronizations. Peters has compared six schedules. In Table 6.1 the results are depicted.

Table 6.1: Computational results (averages).

Criteria	Tabu search	Human scheduler
Total tardiness [hours]	1.5	4.7
$\sum sds$ [hours]	77	72
Missed synchronizations [%]	1%	25%

The time necessary to generate a legal schedule with the designed algorithm varied between 1 and 10 minutes, dependent on the number of operations which varied between 40 and 50 operations. The human scheduler used about 8 hours per schedule. The experiments show that the designed algorithm outperforms the human scheduler on total tardiness and missed synchronizations. However, no firm conclusions can be drawn, more elaborate experiments have to be done. The first impressions are that the designed algorithm seems to generate schedules with acceptable quality.

6.2 Preparation scheduling

In this section the scheduling problem for preparation equipment is described. The Preparation Department consists of a large number of receiver tanks, blenders, filters, buffer tanks, and specialized equipment such as a homogenizer and a separator. Preventive maintenance and change-over times are not taken into account, because they are considered to be of minor importance. Raw materials enter the production system via a drum dumper, or a discharging facility. Transportation in the system takes place with pumps, pipes and valves. For the preparation of a product a number of operations need to be executed. Precedence relations determine the ordering of these operations. Preparation starts with filling a receiver tank with the necessary ingredients. The tank is stirred until a homogenous blend has been retrieved. When the assigned pasteurizer and packaging line have become available, the homogenous blend is pumped to the pasteurizer and the packaging line. In the sequel, filling, stirring and emptying a tank is considered to be one operation. The route from the tank to the pasteurizer is created by switching a number of valves. When necessary, a blender, filter, and buffer tank are incorporated in this route. Although the provided flexibility is high, the infrastructure of pipes and valves has its restrictions. Therefore, downstream resources are accessible to a limited number of upstream resources. The resources on the route are claimed until the packaging operation has been completed. As a consequence, the processing time of the packaging line directly determines the processing time of the equipment upstream. Moreover, when a batch is packed using more than one packaging line, the processing time of the equipment upstream is determined by the packaging schedule. In Figure 6.4 a schedule for the production of blend A in two packaging types is depicted as a Gantt chart.

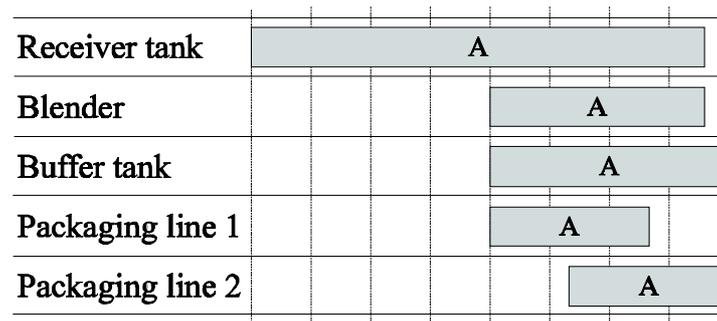


Figure 6.4: Example of a schedule.

The processing times for the receiver tank, blender and buffer tank are completely determined by the packaging line schedule. Furthermore, the end of the packaging

operation at line 2 defines the due date of the emptying, blending, and buffering operation. The operations responsible for emptying, blending, and buffering have to be synchronized, e.g., when in the process of scheduling, an operation is shifted in time, the other operations need to be shifted with the same amount.

Informal problem description

The Preparation Scheduling Problem (PRSP) is defined as the problem of assigning resources to non-preemptive operations over time. A resource is not allowed to process more than one operation at a time. The assigned resources have to be feasible and accessible. Furthermore, the size of an operation has to be smaller than the capacity of the assigned resource. Whenever two operations are linked with a synchronization constraint, the start and end times of these two operation cannot be moved relatively to each other. The definition of synchronization in the PRSP problem differs from that used in the PPSP problem, for an exact definition we refer to the formal problem description. An operation is not allowed to start before 0 or finish after the end of the scheduling horizon. The objective is to generate a feasible schedule that minimizes the mean absolute lateness. The absolute lateness of an operation is defined as absolute difference between the due date of an operation and its completion time. This performance criterium penalizes tardiness as well as earliness, it is also referred to as the absolute deviation problem.

Formal problem description

Definition 6.5 Formally, an instance *PRSP* of the considered Preparation Scheduling Problem is defined as an eleven tuple

$$PRSP = (\mathcal{O}, \mathcal{R}, \mathcal{RS}, fr, sz, dd, pt, cp, sy, ro, H), \quad (6.11)$$

where \mathcal{O} is a set of operations, \mathcal{R} a set of resources, and $\mathcal{RS} \subseteq \mathcal{P}(\mathcal{R})$ a set of resource sets. For each operation there is set of feasible resources, given by function $fr: \mathcal{O} \rightarrow \mathcal{RS}$; a size, given by function $sz: \mathcal{O} \rightarrow \mathbb{N}$; a due date, given by function $dd: \mathcal{O} \rightarrow \mathbb{R}$; and a processing time, given by function $pt: \mathcal{O} \rightarrow \mathbb{R}$. Function $cp: \mathcal{R} \rightarrow \mathbb{R}$ gives for each resource the maximum capacity, which can be the volume in the case of a tank or the maximum flow in case of a continuous process. To prepare a product, a sequence of operations needs to be performed. Each sequence is defined by a number of binary synchronization relations. Function $sy: \mathcal{O} \times \mathcal{O} \rightarrow \{\text{true}, \text{false}\}$, determines for each pair of operations whether or not a synchronization relation exists. Each synchronization relation is unidirectional, i.e., for all $o, o' \in \mathcal{O}$, $sy(o, o') \Rightarrow \neg sy(o', o)$.

When each operation in a sequence is assigned to a feasible resource, a routing is created that includes all the assigned resources. Due to hardware restrictions, the flexibility to create routings is limited. Function $ro: \mathcal{R} \rightarrow \mathcal{RS}$ defines for each resource $R \in \mathcal{R}$, a set of resources $RS \in \mathcal{RS}$ that are accessible to R . $H \in \mathbb{R}$ is the scheduling horizon. \square

Definition 6.6 Schedule s for instance $PRSP$ is a function $s: \mathcal{O} \rightarrow \mathcal{R} \times \mathbb{R}$, which for each operation defines a tuple containing the assigned resource and the start time. The set of all schedules for instance $PRSP$ is denoted by \mathcal{S}_{PRSP} . \square

Definition 6.7 Let $s \in \mathcal{S}_{PRSP}$ be a schedule for instance $PRSP$. We define three functions that are related to schedule s . Function $ra_s: \mathcal{O} \rightarrow \mathcal{R}$, that defines for each operation the resource assignment of schedule s . For each $o \in \mathcal{O}$, $ra_s(o) = \pi_1(s(o))$, where π_1 denotes a function that returns the first element of tuple $s(o)$. Function $st_s: \mathcal{O} \rightarrow \mathbb{R}$, that for each operation defines the start time. For each operation $o \in \mathcal{O}$, $st_s(o) = \pi_2(s(o))$. Function $ct_s: \mathcal{O} \rightarrow \mathbb{R}$, which for each operation defines the completion time. For each $o \in \mathcal{O}$, $ct_s(o) = st_s(o) + pt(o)$. \square

A schedule s is feasible if:

- An assigned resource is a feasible resource, i.e., for all $o \in \mathcal{O}$,

$$ra_s(o) \in fr(o). \quad (6.12)$$

- The size of each operation is less than or equal to the capacity of the assigned resource, i.e., for all $o \in \mathcal{O}$,

$$cp(ra_s(o)) \geq sz(o). \quad (6.13)$$

- For each pair of operations for which a synchronization relation holds, the respective assigned resources can be included in the same routing, i.e., for all $o, o' \in \mathcal{O}$,

$$sy(o, o') \Rightarrow ra_s(o') \in ro(ra_s(o)). \quad (6.14)$$

- A resource is not allowed to process more than one operation at a time, i.e., for all $o, o' \in \mathcal{O}$,

$$o \neq o' \wedge ra_s(o) = ra_s(o') \Rightarrow ct_s(o) \leq st_s(o') \vee ct_s(o') \leq st_s(o). \quad (6.15)$$

- Each pair of operations for which a synchronization relation holds, has to be synchronized according to the following relation: for all $o, o' \in \mathcal{O}$,

$$sy(o, o') \Rightarrow ct_s(o) - dd(o) = ct_s(o') - dd(o'). \quad (6.16)$$

- No operation is started before 0, and is completed after the end of the scheduling horizon, i.e., for all $o \in \mathcal{O}$,

$$st_s(o) \geq 0 \wedge ct_s(o) \leq H. \quad (6.17)$$

Definition 6.8 The quality of every feasible schedule $s \in \mathcal{S}_{PRSP}$ is evaluated with a cost function $f_s: \mathcal{S}_{PRSP} \rightarrow \mathbb{R}$. As optimality criterium the total absolute lateness of the schedule is used. Lateness penalizes earliness and tardiness equally. For each scheduled operation the lateness is given by $|ct_s(o) - dd(o)|$. Function f_s is defined as the total absolute lateness of a schedule $s \in \mathcal{S}_{PRSP}$, which is calculated by

$$f_s(s) = \sum_{o \in \mathcal{O}} |ct_s(o) - dd(o)|. \quad (6.18)$$

□

The objective is to find a feasible schedule $s \in \mathcal{S}_{PRSP}$ that minimizes f_s .

Computational complexity

The mean absolute lateness is a nonregular performance measure, which means that insertion of idle time may improve the performance of a schedule. As a result, the search for an optimal solution cannot be restricted to a finite set such as the nondelay schedules. Baker and Scudder [Bak90] present a review of the scheduling literature concerned with minimizing total tardiness and earliness penalties. They discern two classes of earliness/tardiness problems: the first class involves scheduling problems with a common due date for all jobs, the second class permits distinct due dates. Problems with common due dates tend to be significantly easier to solve than problems with distinct given due dates, such as the PRSP. Garey et al. [Gar88] have shown that the earliness/tardiness problem with distinct given due dates is NP-complete. The PRSP comprises further complicating aspects such as multiple resources, and synchronization of operations. Therefore, the computational complexity of the PRSP is considered to be NP-hard.

Solving the PRSP

A single machine absolute lateness problem with distinct given due dates can be solved in two steps: sequencing the operations and subsequently determining the idle time to insert. Inserting idle time for a given sequence can be formulated as a linear programming problem, which can be solved to optimality [Bak90]. Garey et

al. [Gar88] and Fry et al. [Fry87] have developed simple heuristic procedures which find an optimal solution in $O(n \log n)$ time. To determine the best sequence, Fry et al. [Fry90] present an adjacent pairwise interchange procedure (API). Fry has tested this API method in combination with an idle time insertion algorithm on 192 problem instances and finds that it averages 2.49% worse than the optimal. In the initial schedule, the operations are sequenced according to smallest slack (SLK).

The PRSP involves multiple identical resources. Sundararavaghan and Ahmed [Sun84] treat absolute lateness multimachine scheduling problems with common due dates. As far as we know, literature does not treat algorithms for the multimachine scheduling problem with distinct given due dates. Hijdra [Hij98] has designed a procedure to solve the PRSP on the basis of the single machine algorithm developed by Fry et al. [Fry87], [Fry90].

To reduce the complexity of the PRSP the complete set of preparation resources is divided in four subsets: receiver tanks, blenders, filters, and buffer tanks. Products visit one or more resources in one or more of these subsets. The set of legal resources for each operation to be scheduled is a subset of only one of these four subsets. The four subsets are scheduled one after another starting with the receiver tanks ($i := 1$). In Figure 6.5 procedure SCHEDULEPRSP is shown. Set \mathcal{O} denotes the set of all operations to be scheduled. First, all operations are selected that need to be scheduled on the resource set under consideration, these operations are stored in set \mathcal{O}' . Initially, the set of operations is sequenced according to the number of legal resources (NLR), the operation with the smallest number first. Operations with an equal number of legal resources are sequenced according to smallest slack (SLK), breaking ties by choosing the operation with the longest processing time (LPT). Now, all operations are scheduled one after another, starting with the head of \mathcal{O}' . Then, the set of legal resources \mathcal{R} of o is determined. Furthermore, cost value f and resource r_{best} are initialized. Operation o is assigned to each legal resources. First o is added to the schedule of resource r , and the operations are sorted according to smallest slack. Subsequently, this schedule is optimized with the adjacent pairwise interchange procedure API and its cost f is determined. If the cost value of s_r is smaller than the best cost value f found, the cost f and resource r_{best} are modified. The operation o is removed from the schedule s_r and o is assigned to the next resource in \mathcal{R} . Finally, the operation is assigned to the resource with the lowest cost schedule. When \mathcal{O}' is empty, all operations are scheduled. When the absolute lateness of the schedule is zero, or the maximum number of iterations has been reached, the next resource set is scheduled ($i := i + 1$). If the absolute lateness is larger than zero ($cost(s_{r_{best}}) > 0$), the schedule is not feasible and the schedule data are modified on the basis of the retrieved schedule $s_{r_{best}}$. The function *modify* modifies the due date of some operations in set \mathcal{O} in order to be able to take care of synchronization and modifies the set of feasible resources of some operations on

the basis of resource assignments in schedule $s_{r_{best}}$. It has to be mentioned here that as a result of the modification, the scheduling problem is altered, and therefore, a slightly different problem is solved. The procedure terminates when all four resource sets \mathcal{R}_{rt} , \mathcal{R}_b , \mathcal{R}_f , and \mathcal{R}_{bt} are scheduled.

```

proc SCHEDULE PRSP
   $\mathcal{O} ::=$  'set of all operations';  $\mathcal{R}_{rt} ::=$  'set of all receiver tanks';
   $\mathcal{R}_b ::=$  'set of all blenders';  $\mathcal{R}_f ::=$  'set of all filters';
   $\mathcal{R}_{bt} ::=$  'set of all buffer tanks';
   $rt := (\mathcal{R}_{rt}, \mathcal{R}_b, \mathcal{R}_f, \mathcal{R}_{bt}); i := 1; iter := 20;$ 
  while  $i \leq 4$  do
     $\mathcal{O}' := \{o \mid o \in \mathcal{O}, \text{feasible resources}(o) \subseteq \pi_i(rt)\};$ 
     $\mathcal{O}' := \text{sequence}(\mathcal{O}', \text{NLR/SLK/LPT});$ 
    while  $\mathcal{O}' \neq \emptyset$  do
       $o := \text{head}(\mathcal{O}'); \mathcal{O}' := \text{tail}(\mathcal{O}'); f := \infty;$ 
       $\mathcal{R} := \text{feasible resources}(o); r_{best} := \text{pick } \mathcal{R};$ 
      while  $\mathcal{R} \neq \emptyset$  do
         $r := \text{pick } \mathcal{R}; \mathcal{R} := \mathcal{R} - \{r\}; s_r := \text{add}(s_r, o);$ 
         $s_r := \text{sort}(s_r, \text{SLK}); \text{API}; f := \text{cost}(s_r);$ 
        if  $\text{cost}(s_r) < f$  then  $f := \text{cost}(s_r); r_{best} := r$ 
        endif;
         $s_r := \text{remove}(s_r, o);$ 
      end while;
       $s_{r_{best}} := \text{add}(s_{r_{best}}, o);$ 
    end while
    if  $\text{cost}(s_{r_{best}}) > 0$  and  $i > 1$  and  $iter < 20$ 
      then  $i := 1;$ 
      else  $i := i + 1;$ 
    endif;
     $\mathcal{O} := \text{modify}(\mathcal{O}, s_{r_{best}}, i);$ 
  end while
endproc

```

Figure 6.5: The SCHEDULE PRSP algorithm.

Figure 6.6 shows the API algorithm in pseudo-code. Initially, schedule s_{init} is optimized with the procedure INSERT IDLE TIME. For an elaborate description of this procedure, we refer to Fry et al. [Fry87]. The cost f of this solution is calculated with the function cost . The number of operations n in s is determined with the function size , and i is initialized. Now, the algorithm iterates as long as $i < n$. The function transpose swaps operations i and its successor. When this swap results in

a solution with better quality, this solution s' is accepted as best solution ever and i is reset, the procedure starts all over again. When $n - 1$ successive iterations have failed to produce a solution of better quality, the algorithm terminates and returns schedule s .

```

proc API
  INSERT IDLE TIME;
   $f := cost(s); n := size(s); i := 1;$ 
  while  $i < n$  do
     $s' := transpose(s, i);$ 
    INSERT IDLE TIME;
    if  $cost(s') < f$  then
       $s := s'; f := cost(s'); i := 1;$ 
    endif;
  end while;
endproc

```

Figure 6.6: The API algorithm.

Computational results

Hijdra [Hij98] has performed experiments to test the quality of the designed scheduling algorithm. In contrast with pasteurizing and packaging schedules, no real life preparation schedules were available at that time. Therefore, Hijdra has constructed four optimal preparation schedules, that consist of about forty operations. The utilization of the schedules varies from 87% to 105%, which means that the schedules are heavily loaded. The scheduling data for these schedules are fed to the preparation scheduler and the results have been compared with the four optimal solutions. An optimality ratio

$$optimality\ ratio = \frac{total\ workload - total\ absolute\ lateness}{total\ workload},$$

has been defined to analyze the results. The results show that the algorithm has generated schedules with an average optimality ratio of 92%. More elaborate experiments need to be done to quantify the algorithms performance. However, the algorithm has been used for six months to generate weekly schedules for the Preparation Department. Riedel's Planning Department and the operators of the Preparation Department have appreciated the schedules that are generated. In most cases, only a few modifications have been made, through relaxing the constraints. Generally, generating a preparation schedule takes less than five minutes, with a Pentium

250Mhz personal computer. The number of operations varies between thirty and forty.

6.3 Refining the design of resources

In the former sections, two scheduling procedures have been designed. These procedures have been integrated in a design tool. The objective of this operational design tool is to refine the resource design of Chapter 5, and to support redesign projects. It enables fast evaluation of ‘what if’ scenarios.

Solving the scheduling problem consists of four steps: generating an instance for the pasteurizing and packaging scheduling problem (PPSP), solving this PPSP instance, generating an instance for preparation scheduling problem (PRSP), and solving this PRSP instance. In Figure 6.7 the configuration of the operational design tool is depicted.

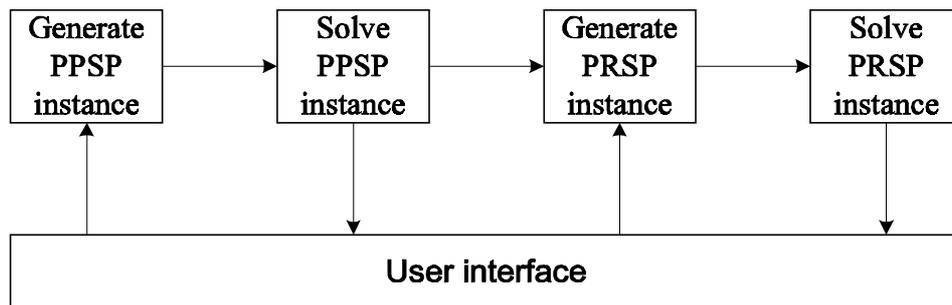


Figure 6.7: Configuration of the operational design tool.

Generating an instance for the PPSP is done on the basis of a Master Production Schedule (MPS). The MPS contains for each product the production volume and inventory levels per time interval of one week, for at least one year ahead. In Section 5.1 a procedure is described to calculate the total production volume and inventory levels per time interval. This procedure needs to be extended to calculate the production volumes per product. However, this is not treated in this thesis, a MPS is considered to be available. From the MPS a large number of problem instances can be generated. For every instance, a set of orders to be scheduled is extracted from the MPS and combined with necessary manufacturing data. An instance for the PPSP is solved with the algorithm specified in Section 6.1.

Next, the solution is used to generate an instance for the PRSP. When generating such an instance, a few things have to be reckoned with. Two or more pasteurizing and packaging operations which process the same blend can possibly be combined in one preparation batch (see Figure 6.4). However, the maximum volume of this preparation batch depends on the product recipe and the preparation procedures in combination with the capacity of the feasible resources. Furthermore, the lead time of the preparation batch is not allowed to exceed a specified maximum, as the of fruit juice quality degrades fast under non-aseptic conditions. On the other hand, a pasteurizing and packaging operation may be split in several preparation batches, because of its size. Again this depends on the recipe, the preparation procedures, the capacity of the feasible resources, and the lead time of the batch. A procedure to generate an instance for the PRSP has been designed and implemented. When an instance for the PRSP is constructed, it is solved with the algorithm specified in Section 6.2.

Data management and visualization. All procedures integrated in the tool have been coded with the specification formalism χ and compiled with the χ -engine. The executables have been embedded in a Microsoft Excel shell. This shell provides an interface which allows the user to specify the necessary scheduling data and edit the problem instances. The scheduling data are composed of information concerning production orders, resources and their configuration, manufacturing procedures, recipes, change-over matrices, and various time and resource constraints. Communication between the user and the scheduling procedures is done via computer files, generated by the interface. Visualization of scheduling data is done in a tabular format and visualization of the schedules is done via Gantt charts.

To illustrate its use, the operational design tool is applied to the carbonated products group presented on page 59.

Riedel example

The configuration calculated in the resource design phase consists of one discharging pump, three receiver tanks of 70 cubic meters, two outgoing pumps, two in-line blenders with two carbon dioxide injectors and two buffer tanks. In the original configuration, this preparation equipment needed to feed two pasteurizer/packaging simultaneously. Now, suppose Riedel plans to introduce an additional packaging variant for the carbonated blends. The question arises whether Riedel has installed enough preparation equipment to feed three packaging lines simultaneously. To answer this question, the operational design tool is applied. First, the manufacturing equipment for the carbonated products is isolated from its environment, and

considered as a separate plant. A model of this plant is generated with the tools user-interface. When the tool has been configured, experiments are conducted. In general, every experiment deals with an order list of one week, which is extracted from the MPS. In Table 6.2 an example of such a list of orders is depicted. The filling machines CFA112, CFA207, and TBA1500 produce cans of 250 ml., 1000 ml., and 1500 ml., respectively.

Table 6.2: An example of an order list.

Blend	Packaging	Quantity	Due	Blend	Packaging	Quantity	Due
	line	[$\times 1000$ cans]	date		line	[$\times 1000$ cans]	date
DFa	CFA112	200	Tue 6pm	Xa	CFA112	125	Fri 6pm
DFa	CFA207	315	Tue 6pm	Xa	CFA207	175	Fri 6pm
DFb	CFA207	125	Fri 6pm	Xb	CFA112	150	Fri 6pm
DFc	CFA112	300	Fri 6pm	Xb	CFA207	200	Fri 6pm
DFc	CFA207	160	Fri 6pm	Xc	CFA112	100	Fri 6pm
DFd	CFA207	100	Thu 6am	Xe	TBA1500	140	Fri 6pm
DFd	TBA1500	130	Thu 6am	Xf	TBA1500	100	Fri 6pm

Together with the order list, product and manufacturing characteristics need to be fed to the scheduler. These characteristics concern resource capacity and configuration (network of pipes and valves), product ingredients, preparation procedures, change-over matrices, and various time and resource constraints.

To illustrate the influence of constraints, we mention one of the time constraints, which is related to the discharging of drums. Production management prefers to operate the drum discharging facility in one shift. As a result, drums can be discharged from 8 am until 4 pm. This influences the start time of operations in the Preparation Department. In Figure 6.8 the resulting schedule is depicted.

All carbonated products contain ingredients purchased in drums. Therefore, the time constraint mentioned above has to be met. However, as the Gantt chart shows, the receiver tank operation for product DFa starts at the beginning of the scheduling horizon. The special chemical characteristics of this product are such, that fast deterioration of product quality under non-aseptic conditions can be prevented. As a result, this product can be discharged at Fridays, and stored during the weekend. All other receiver tank operations start between 8am and 2pm, because they contain ingredients, purchased in drums. All packaging operations have been finished before their due dates, the total tardiness equals zero. From five possible synchronizations of packaging operations, three have been met. The sum of sequence dependent change-over time equals 25 hours, where the optimal value is 21 hours. For the preparation operations the total absolute lateness equals zero. On the basis of this experiment,

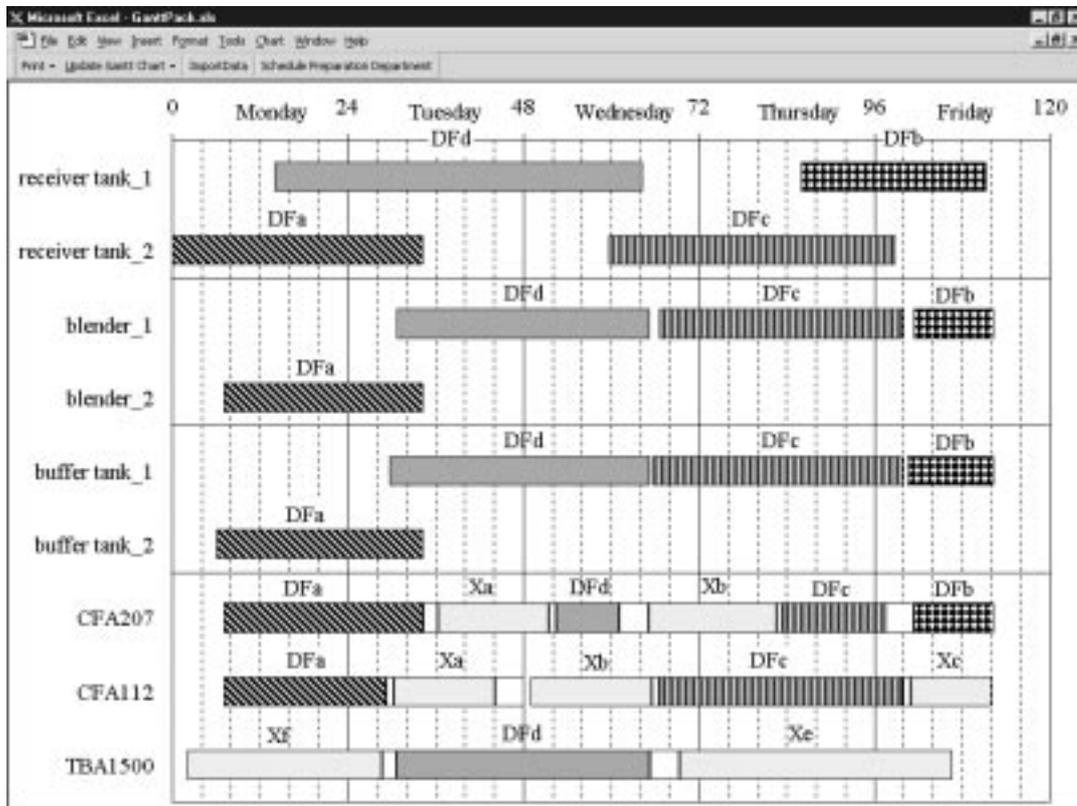


Figure 6.8: An example of a Gantt chart.

Riedel would not invest in new preparation equipment. However, no firm conclusion can be drawn from a single experiment.

□

6.4 Control system

With respect to manufacturing control this thesis focuses on control aspects that directly influence the design of the manufacturing system, i.e., the control philosophy (Chapter 4), production volumes and inventory levels (Chapter 5) and detailed scheduling procedures (Chapter 6). Other control tasks, such as shopfloor control and master production scheduling are not treated in detail. However, to provide a control framework, this section briefly discusses the control system that is now being implemented at Riedel.

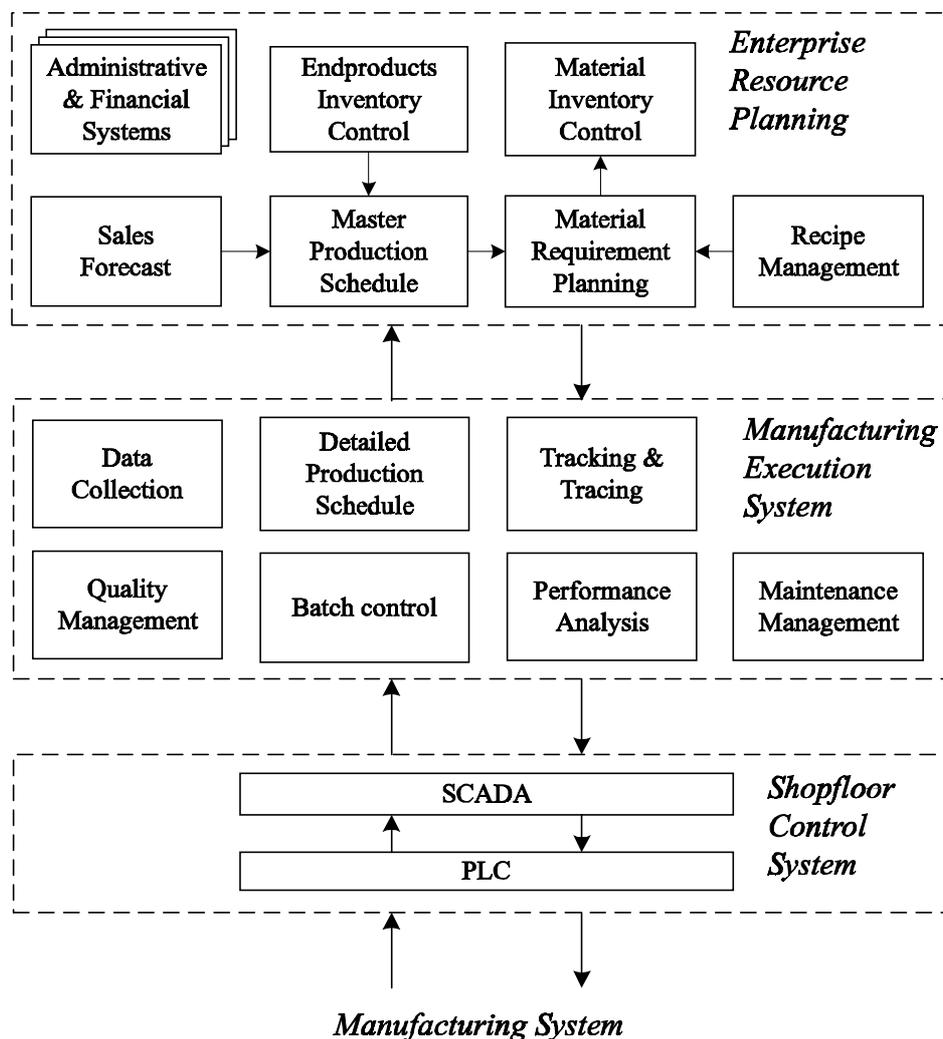


Figure 6.9: Riedel's control system in the near future.

The control system is composed of three parts: the Enterprise Resource Planning (ERP), the Manufacturing Execution System (MES) and the Shopfloor Control System (SCS). In Figure 6.9 this control system is depicted.

ERP takes care of the so-called business functions, i.e., administration and finance, sales forecasting, inventory control, material requirement planning and production planning. With respect to manufacturing control the Master Production Schedule (MPS) function is the core of ERP. The MPS translates sales forecast and inventory levels into production volumes per product per week for at least one year ahead. On

the basis of the MPS, the Materials Requirement Planning (MRP) determines when and in which amounts materials have to be available. Some materials are purchased just in time, others are available on stock. Materials Inventory Control checks the corresponding inventory levels and replenishes inventories when necessary. Every week the MPS specifies which orders need to be scheduled in detail. These orders are communicated with the MES.

MES is a set of functions for the control and monitoring of product quality and production performance. A large number of functions can be integrated in the MES. For Riedel, the most important functions are depicted in Figure 6.9. The Data Collection function stores production monitoring information in a database. The Quality Management function analyses real-time the monitoring information from this database to guarantee a high product quality. The Tracking and Tracing function registers for every batch all necessary data, such that materials can be traced in the endproducts in case of quality problems. Furthermore, the exact routing of every batch is registered, together with the involved operators, and possible alarms and exceptions. The Batch Control function controls all batches, including their routing. With the operation procedures, an order from the MPS can be translated into a sequence of operations. Every week, the Detailed Production Scheduling function schedules all operations. Detailed scheduling of the Riedel plant has been treated in Sections 6.1 and 6.2. Scheduling procedures have been developed and implemented in an operational design tool. However, in addition to support the design process, these procedures have been applied successfully as a Detailed Production Scheduling module. The detailed scheduling procedures support generating a schedule for the production of one week. Performance Analysis provides production management with the necessary reports concerning the performance of the plant. Performance indicators are directly compared with historical data and targets. Maintenance Management informs maintenance personnel on the technical status of all manufacturing equipment and schedules preventive maintenance actions.

The Shopfloor Control System (SCS) takes care of material flow control. For example, when an amount of orange juice concentrate has to be pumped from one tank to another, the Batch Control module in the MES checks whether the assigned pipes, pumps and destination tank have been cleaned, and the resource tank has been filled. If all safety constraints are met, the SCS switches the correct valves, starts the pump. During the transfer process the SCS monitors its progress and displays necessary information to the operator. When the correct amount of juice has been pumped to the destination tank, the SCS stops the pump, switches valves and if necessary, initializes a cleaning procedure.

In the case of Riedel, the SCS is composed of a Supervisory Control And Data Acquisition (SCADA) application and a layer with Programmable Logic Controllers (PLC).

The PLCs take care of direct equipment control, via input/output devices. The SCADA application provides an interface with the operator and transforms raw process data in information which can be used for control purposes. SCADA/PLC systems have proven their fitness for the semi-processing industry. Other possibilities for shopfloor control are a Distributed Control System (DCS) or a soft control system. In a DCS, PLCs are programmed indirectly, code is generated at a high level and distributed to the PLC. Two or more PLCs with the same functionality can be programmed at once. As a result, modifications and extensions are done relatively easy. However, a large part of Riedel's processing equipment already contains an integrated PLC, with equipment specific functionality. Transferring this code in a DCS system is very expensive. In a soft control system, there are no PLCs, personal computers have taken over their tasks. However, robustness of a soft control system is still considered as a weakness.

The control philosophy is directly reflected by the SCS. When a group technology philosophy is applied, the SCS is structured according to these groups. This starts with the internal coding structure of manufacturing equipment and ends with the operator screens. The operators are able to access screens that provide all relevant information concerning equipment in a product group, rather than screens concerning all blenders or tanks.

6.5 Conclusion

In this chapter, the resource design has been refined. This means that the influence of scheduling related variables on the performance of the plant is taken into account. In order to support the process of refining the design of resources, an operational design tool has been designed. With this tool, future manufacturing and control scenarios are evaluated, and redesign processes are supported. The operational design tool contains two scheduling modules, one for the pasteurizing and packaging equipment, and one for the preparation equipment. The latter has been used extensively in industrial practice for six months and performs very well. The pasteurizing and packaging module has been tested briefly by Riedel's scheduler. First impressions are that the module generates schedules with good quality. It takes two working days to generate a complete production schedule by hand, the operational design tool takes about ten minutes to generate a pasteurizing and packaging schedule, and an additional five minutes to generate a preparation schedule.

A framework for Riedel's manufacturing control system has been given. The control framework consists of three levels: ERP, MES, and SCS. ERP takes care of the business processes, MES provides functions for the control and monitoring of product quality

and production performance, and SCS takes care of real-time material flow control. The architecture that has been designed in this thesis is reflected in the SCS. The scheduling procedures are integrated in the MES.

Chapter 7

Conclusions

In this thesis a method is presented to support the design process of a fruit juice blending and packaging plant. Development of such a method is important for two reasons: first, the ever increasing competition, diversification of the product range and high service levels stress the need for a high quality production system design, and second, design literature does not provide a structure that adequately supports the design process of such types of plants. The Riedel juice plant is used as a case study.

Design literature provides a large amount of detailed design methods developed for specific design problems rather than generally applicable methods for designing production systems. Furthermore, designing production systems is treated as a static problem and the design process is finished before the actual building starts. In practice, production systems evolve throughout their life-cycle, due to ever changing market demands. As a result, the design process continues as long as the industrial system is used. Finally, in research as well as in practice, production control has been viewed in isolation from the manufacturing system design. However, as manufacturing and control show large interaction, the quality of the production system design can be enhanced by considering them simultaneously.

The proposed design method can be used throughout the specification and utilization phase of a production system. It supports initial design as well as redesign of a utilized production facility and regards the design of the manufacturing system as well as the design of the control system, where it affects the manufacturing system. The design method is composed of a structure and a set of tools. The structure defines which decisions need to be made and when to make them. The tools support the process of decision making. The structure is generally applicable for the design of industrial systems, the set of tools makes the method fit for the design of a fruit juice

blending and packaging plant. The method consists of four phases: the definition of objectives and constraints, and the design of architecture, resources, and operations.

Every design process starts with identifying design objectives and constraints. The design objectives are deduced from the business objectives of a company. During the design process, these objectives and constraints remain subject of discussion. Riedel's business objectives state that Riedel is a producer of ready-to-serve fruit juices and fruit beverages for the Dutch and Belgian consumer markets. Through a diverse, innovative, and high quality product range, exclusively composed of distinct brands, Riedel aims at being a leader in multiple market segments. Generating high profits is the main financial goal. As a result, the design objectives state that Riedel has to produce a small number of large batches and a large number of small batches every week, in a cost efficient way. Furthermore, it needs to support a short time-to-market for new and innovative products.

An industrial system is an open system that interacts with an environment. This environment imposes constraints on the system to be designed. In this thesis, these constraints have been divided in five categories: customers, suppliers, government, financiers, and labor market. Constraints related to each of these five categories are translated into constraints for the production system. For Riedel this has resulted in, amongst others, JIT deliveries of ingredients, aseptic packaging, on-stock production, and ambient and chilled distribution.

In the design of architecture phase, the structure of the manufacturing system and the control system is designed. The design of architecture is divided into two steps: the sequencing of functions and the configuration of processes. For the blending and packaging of fruit juice beverages three basic functions are distinguished: preparing, pasteurizing, and packaging. These functions have to be performed sequentially. Two essential aspects in determining the best sequence are product quality and technological developments. For Riedel, the functions preparing, pasteurizing, and packaging are best performed in that sequence. The second step deals with determining the configuration of processes. The process structure for pasteurizing and packaging clearly differs from the process structure for preparing the beverages. The pasteurizing and packaging processes form a relatively simple flow line, the collection of preparation processes has a far more complex structure. The current Preparation Department has a process oriented structure, the biased focus on flexibility has lead to a failure sensitive and labor intensive production facility. In order to simplify the material flow, Group Technology (GT) is applied. The product range is divided into six groups and for every group the necessary processes have been selected and configured. The designed process model represents an architecture that is flexible where needed, but simple and cost efficient whenever possible. Designing a production architecture is an essential phase in the design process, as it strongly influences the controllability

of the manufacturing system, the structure of the control system, the number and type of resources and the floorplan. It prevents the engineering habit of ‘thinking’ in equipment too soon in the design process. The function and process models provide adequate tools for specifying a production architecture.

In the design of resources phase, the architecture is quantified. As the designed architectures for preparing and for pasteurizing and packaging clearly differs, the capacity analysis problem is divided into two: first, the packaging and pasteurizing resources are determined, then the preparation resources. The number and configuration of packaging and pasteurizing resources is calculated with a nine-step procedure. For each packaging type, this procedure is executed once. The procedure is new with respect to two aspects: it incorporates calculation of finished goods inventory levels and it takes into account the fact that setup times are configuration dependent. The preparation resources are calculated by means of dynamic simulation. A design tool has been developed which provides the means to calculate the necessary equipment per product group. The graphical interface and embedded simulation model enable fast evaluation of scenarios. The final task in the resource design is to constitute a floorplan. The pasteurizing and packaging equipment is laid out in-line and the preparation equipment is laid out according to a GT flow line pattern.

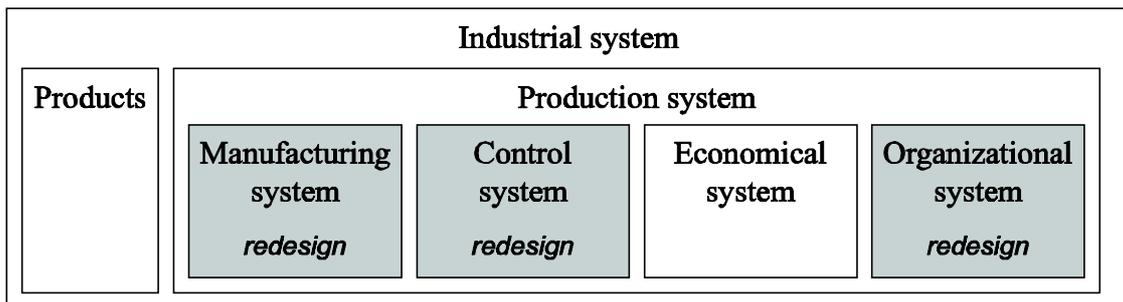
In the design of operations phase, the resource design is refined. The influence of scheduling related variables on the behavior of the production system is taken into account. In order to support this refinement, an operational design tool has been designed. With this tool, manufacturing and control scenarios are evaluated. The tool consists of two scheduling procedures: one for packaging and pasteurizing equipment and one for preparation equipment. Both scheduling problems have been decomposed into several smaller problems, which have been solved with local search algorithms. These scheduling procedures have been linked with graphical and tabular user-interfaces, which support the definition of scheduling problem instances and the analysis of results. The scheduling procedure for the preparation equipment has been tested extensively in industrial practice and has performed very well.

The design method presented in this thesis has been used to support the redesign of Riedel’s production system. Although the design method has been developed for supporting the design process of a new production system, it has shown to be of great help in structuring Riedel’s redesign project. The architecture is being simplified extensively and cost efficiency is increased. Apart from modifications in the manufacturing system and in the control system, the new architecture has triggered large organizational changes. Riedel’s scheduling process has been improved by implementing the PRSP module for scheduling preparation equipment. The PPSP module for scheduling pasteurizing and packaging equipment is being tested, first impressions are that schedules of good quality are generated in about 10 minutes.

The operational design tool can be improved by adding iteration between the PRSP and the PPSP module. Applying the proposed method for designing a new fruit juice blending and packaging plant will have to prove the quality of the method; Riedel's experiences are promising.

Epilogue

The design method presented in this thesis is used to redesign Riedel's blending and packaging plant. The scope of the redesign project is broader than the scope of this thesis, as it also involves implementing a shop-floor control system and restructuring the organizational system (see figure below). This epilogue describes briefly Riedel's redesign project with respect to the manufacturing system, the control system and the organizational system.



Scope of the redesign project.

Manufacturing

The redesign project has affected the architecture and the resources present in the manufacturing system.

Architecture. Riedel has decided to implement a Group Technology (GT) architecture in the Preparation Department. As discussed in Chapter 4, Riedel distinguishes the product groups Appelsientje (orange juice); Goudappeltje (apple juice); Carbonated drinks; Taksi; Rest, in-line dilution; and Rest, in-tank dilution. In 1998 the first experiments with GT were carried out on Appelsientje. Two receiver tanks

and one blender were installed as dedicated equipment to process Appelsientje. The blender was placed under the two tanks, which has resulted in a compact installation. In order to be able to empty one tank while filling the other, a number of double seated valves were installed. Since then, this dedicated installation has produced over hundred million liters of high quality orange juice, with relatively low operator effort.

Resources. The GT architecture extensively simplifies the production process. Because of these simplifications, the costs of automating the production process have been decreased substantially. Furthermore, GT has enabled the use of specialized equipment, as a result, processing times have been reduced substantially. For example, stirring a batch of orange juice concentrate and essences used to take four hours, nowadays, with the use of specialized mixing gear, it takes only one hour. For every group the production process has been analysed. First, the routings of the products in a group were depicted in a flow chart. Then, all activities involved with the production of a batch were listed and structured. The workload and processing time of each activity or group of activities have been determined. The activities have been classified as production, cleaning or reclaim activities. The latter two categories have been considered as non-value-added activities, or waste, and therefore, should be minimized. With Activity Based Costing techniques the cost structure of the production process has been analysed. The results of these analyses were discussed with operators, technologists and production management. On the basis of these discussions, 'Ist' and 'Soll' situations were sketched, and a large number of process improvement projects have been started. Finally, for each group, detailed resource designs were made in cooperation with suppliers.

Control

The shop floor control system (SCS) of Riedel's Preparation Department consists of a large amount of stand-alone control systems. Together with the control system of the pasteurizers, these stand-alone systems are being integrated into one modern SCADA/PLC shop floor control system. As a result, operators will be able to monitor and control the majority of their production processes from the control room. The GT architecture has a large influence on the SCS design. This starts with the coding of manufacturing equipment and ends with the operators screens. Implementing this new shop floor control system is planned in phases. The first phase is concerned with the groups Appelsientje and Goudappeltje, the second phase is concerned with the Carbonated drinks, and so on. In each phase a part of the old control sys-

tem is migrated to the new SCS. Interfacing the old and the new SCS during the implementation is considered to be a critical task.

Organization

The Preparation Department and the Pasteurizing Department are being integrated into one Processing Department. The preparation and pasteurizing organizations have been functionally organized and strongly hierarchical. In the current organization, for example, an operator is responsible for functional units, such as a group of blenders, or the discharging of trucks. Authorization for actions is limited to direct control activities only. Hierarchy and functional boundaries are typical characteristics of a so called vertical organization. The new processing organization will have key elements of a horizontal organization. Employees will be organized around product groups, and there will be less organizational layers, i.e., the hierarchy is flat. Responsibility and authorization will be extended and will comprise the entire process ranging from the discharging of raw materials, to the pasteurizing of the juice and delivery to the filling line. Prime condition for this transition is that the SCS and the MES will provide the necessary tools, such as a central monitoring of the production process and automatic batch control. Former pasteurizing and preparation operators are trained to become all-round processing operators. When the project is implemented successfully, processing personnel will be reduced with one third. Furthermore, Riedel expects that these changes will lead to a more rewarding work environment and that, as a consequence, the quality of the products will be improved.

Conclusion

Riedel is convinced that the redesign project described above will contribute to the profitability and competitiveness of the company. The design method presented in this thesis has formed the basis for this redesign project. Besides the manufacturing and control system, it has directly influenced the organizational system as well. The core activities of the optimization project are: modifying the architecture, implementing a shopfloor control system, enhancing cost efficiency of the manufacturing process, and restructuring the organization.

Bibliography

- [Aar97] E.H.L. Aarts and J.K. Lenstra, editors. *Local Search in Combinatorial Optimization*. John Wiley and Sons, Chichester, 1997.
- [And97] E.J. Anderson, C.A. Glass, and C.N. Potts. Machine scheduling. In E. Aarts and J.K. Lenstra, editors, *Local Search in Combinatorial Search*. John Wiley and Sons, 1997.
- [Are96] N.W.A. Arends. *A Systems Engineering Specification Formalism*. PhD thesis, Eindhoven University of Technology, The Netherlands, 1996.
- [Bak74] K.R. Baker, editor. *Introduction to Sequencing and Scheduling*. Wiley, 1974.
- [Bak90] K.R. Baker and G.D. Scudder. Sequencing with earliness and tardiness penalties. *Operations Research*, 38:22–36, 1990.
- [Bee98] D.A. van Beek and J.E. Rooda. Languages and applications in hybrid modelling, positioning of the chi language. In *Proceedings of 9th Symposium on Information Control in Manufacturing*, volume 2, pages 77–82, Nancy, 1998.
- [Boo83] G. Boothroyd and P. Dewhurst. *Design for Assembly. A designer's handbook*. University of Massachusetts, 1983.
- [Bra93] L.E.M.W. Brandts. *Design of Industrial Systems*. PhD thesis, Eindhoven University of Technology, The Netherlands, 1993.
- [Bro93] M. Brown, R.L. Kilmer, and K. Bedigian. Overview and trends in the fruit juice processing technology. In S. Nagy, C.S. Chen, and P.E. Shaw, editors, *Fruit Juice Processing Technology*. Agscience, 1993.
- [Bro96] J. Brown, J. Harhen, and Shivnan J., editors. *Production Management Systems*. Addison-Wesley, Reading, Massachusetts, 1996.

- [Buf87] E.S. Buffa and R.K. Sarin. *Modern Production/Operations Management*. Wiley, Chichester, 1987.
- [Bur71] J.L. Burbidge. Production flow analysis. *The Production Engineer*, 4:139–152, 1971.
- [Bur96] J.L. Burbidge. *Period Batch Control*. Oxford University Press, 1996.
- [Cam01] E.J.J. van Campen. *Design of a multi-process multi-product IC wafer fab*. PhD thesis, Eindhoven University of Technology, The Netherlands, 2001. to be published.
- [Cha98] R.B. Chase, N.J. Aquilano, and F.R. Jacobs. *Production and Operations Management, Manufacturing and Services*. Irwin/McGraw-Hill, 1998.
- [Che93] C.S. Chen. Physiochemical principles for the concentration and freezing of fruit juices. In S. Nagy, C.S. Chen, and P.E. Shaw, editors, *Fruit Juice Processing Technology*. Agscience, 1993.
- [Chu92] C. Chu and M.-C. Portmann. Some new efficient methods for solving the $n/1/r_i/\sum t_i$ scheduling problem. *European Journal of Operational Research*, 58:404–413, 1992.
- [Dow95] J.W. Downes. Equipment for extraction and processing of soft and pome fruit juices. In P.R. Ashurst, editor, *Production and Packaging of Non-Carbonated Fruit Juices and Beverages*. Blackie Academic and Professional, 1995.
- [Du90] J. Du and J.Y.-T. Leung. Minimizing total tardiness on one machine is NP-hard. *Mathematics of Operations Research*, 15:483–495, 1990.
- [Fre82] S. French. *Sequencing and Scheduling, An Introduction to the Mathematics of the Job-Shop*. Ellis Horwood Limited, Chichester, 1982.
- [Fre95] L.-B. Fredsted. Processing systems for fruit juice and related products. In P.R. Ashurst, editor, *Production and Packaging of Non-Carbonated Fruit Juices and Beverages*. Blackie Academic and Professional, 1995.
- [Fry87] T.D. Fry, R.D. Armstrong, and J.H. Blackstone. Minimizing weighted absolute deviation in single machine scheduling. *IIE transactions*, 19(4):445–450, 1987.
- [Fry90] T.D. Fry, R.D. Armstrong, and L.D. Rosen. Single machine scheduling to minimize mean absolute lateness: A heuristic solution. *Computers Operations Research*, 17(1):105–112, 1990.

- [Gar88] M.R. Garey, R. Tarjan, and Wilfong G. Processor scheduling with symmetric earliness and tardiness penalties. *Mathematics of Operations Research*, 13:330–348, 1988.
- [Glo89] F. Glover. Tabu search - part I. *ORSA Journal on Computing*, 1:190–206, 1989.
- [Glo90] F. Glover. Tabu search - part II. *ORSA Journal on Computing*, 2:4–32, 1990.
- [Glo93] F. Glover, E. Taillard, and D. De Werra. A user's guide to tabu search. *Annals of Operations Research*, 41:3–28, 1993.
- [Gol84] E.M. Goldratt and J. Cox. *The goal*. Harper Business, North River Press, 1984.
- [Ham93] M. Hammer and J. Champy. *Re-engineering the corporation: a manifest for business revolution*. Harper Business, New York, 1993.
- [Hau89] R. Haupt. A survey of priority rule-based scheduling. *OR Spektrum*, 11:3–16, 1989.
- [Hen95] C.M. Hendrix and J.B. Redd. Chemistry and technology of citrus juices and by-products. In P.R. Ashurst, editor, *Production and Packaging of Non-Carbonated Fruit Juices and Beverages*. Blackie Academic and Professional, 1995.
- [Her97] A. Hertz, E. Taillard, and D. De Werra. Tabu search. In E. Aarts and J.K. Lenstra, editors, *Local Search in Combinatorial Search*. John Wiley and Sons, 1997.
- [Hij98] N.P. Hijdra. Schedulingsalgoritmen voor de bereidingsafdeling van Riedel Drankenindustrie (in dutch). Master's thesis, Eindhoven University of Technology, 1998. SE 420180.
- [Joh95] R.B. Johnston. Making manufacturing practices tacit: a case study of computer-aided production management and lean production. *Journal of the Operational Research Society*, 46:1174–1183, 1995.
- [Kin80] J.R. King. Machine component grouping in production flow analysis: an approach using rank order clustering. *International Journal of Production Research*, 20:117–133, 1980.

- [Kin95] J.R. King. Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm. In C. Moodie, R. Uzsoy, and Y. Yih, editors, *Manufacturing Cells, A Systems Engineering View*. Taylor and Francis Ltd., 1995.
- [Kou94] C.P. Koulamas. The total tardiness problem: Review and extensions. *Operations Research*, 42(6):1025–1041, 1994.
- [Kra96] L.J. Krajewski and L.P. Ritzman. *Operations Management, Strategy and Analysis*. Addison Wesley, Reading, Massachusetts, 5 edition, 1996.
- [Kum92] V. Kumar. Algorithms for constraint-satisfaction problems: A survey. *AI Magazine*, 13(1):32–44, 1992.
- [Lag91] M. Laguna and J.L. González Velarde. A search heuristic for just-in-time scheduling in parallel machines. *Journal of Intelligent Manufacturing*, 2:253–260, 1991.
- [Mit66] S.P. Mitrofanov. *Scientific Principles of Group Technology*. National Lending Library for Science and Technology, 1966.
- [Mon74] U. Montanari. Networks of constraints, fundamental properties and applications to picture processing. *Information Science*, 7:95–132, 1974.
- [Moo95] C. Moodie, R. Uzsoy, R. Yih, and Y. Yih. *Manufacturing Cells, a Systems Engineering View*. Taylor and Francis Ltd., 1995.
- [Nic92] K. Nichols. Design for quality and reliability. *Journal of Engineering Design*, 3(2):139–148, 1992.
- [Nui94] W.P.M. Nuijten. *Time and Resource Constrained Scheduling: A Constraint Satisfaction Approach*. PhD thesis, Eindhoven University of Technology, The Netherlands, 1994.
- [Pan77] S.S. Panwalker and W. Iskander. A survey of scheduling rules. *Operations Research*, 25:45–61, 1977.
- [Pet99] R. Peters. Scheduling Algorithms for the Packaging Department of Riedel Drankenindustrie. Master’s thesis, Eindhoven University of Technology, 1999. SE 420211.
- [Rin75] A.H.G. Rinnooy Kan, B.J. Lageweg, and J.K. Lenstra. Minimizing total costs in one-machine scheduling. *Operations Research*, 23:908–927, 1975.

- [Roo97] J.E. Rooda. The modelling of industrial systems. Lecture notes, Eindhoven University of Technology, The Netherlands, 1997.
- [Roy98] R. Roy. Scheduling and control, performance measures and discrete event simulation. *Journal of the Operational Research Society*, 49:151–156, 1998.
- [Sch86] R.J. Schonberger. *World Class Manufacturing: The lessons of simplicity applied*. Free Press, 1986.
- [Sha95] N. Shah, K. Kuriyan, L. Liberis, C.C. Pantelides, L.G. Papageorgiou, and P. Riminucci. User interfaces for mathematical programming based multi-purpose plant optimisation systems. *Computers and Chemical Engineering*, 19:S765–S772, 1995.
- [Shi81] S. Shingo. *Study of "Toyota" Production System: From Industrial Engineering Viewpoint*. Japan Management Association, Tokyo, 1981.
- [Sit94] L.U. de Sitter. *Synergetisch produceren*. Van Gorcum, Assen, 1994.
- [Sun84] P. Sundararaghavan and M. Ahmed. Minimizing the sum of absolute lateness in single machine and multimachine scheduling. *Naval Research Logistics Quarterly*, 31:325–333, 1984.
- [Vol92] T.E. Vollmann, W.L. Berry, and D.C. Whybark. *Manufacturing Planning and Control Systems*. Business One Irwin, third edition, 1992.
- [Wie95] H.P. Wiendahl. *Load-oriented manufacturing control*. Springer-Verlag, 1995.
- [Wom90] J.P. Womack, D.T. Jones, and D. Roos. *The machine that changed the world: how Japan's secret weapon in the global auto wars will revolutionize Western industry: the story of lean production*. Harper Perennial, 1990.

Samenvatting

De consumentenmarkt voor vruchtensappen en -dranken heeft de laatste vijf jaren te maken gehad met steeds hevigere concurrentie. De industrie moet betere kwaliteit en meer variëteit leveren tegen steeds scherper wordende prijzen. Vanuit het oogpunt van productie moet een hoge leverbetrouwbaarheid worden gecombineerd met korte doorlooptijden. Dit samen met de sterke stijging van de productvariëteit, heeft gezorgd voor nieuwe prestatienormen voor fabrieken in deze industrie.

De literatuur beschrijft veel ontwerpmethoden die geschikt zijn voor een specifieke subset van ontwerpproblemen. Er is echter geen universeel geaccepteerde methode voor het ontwerpen van een hele fabriek. Verder negeert de literatuur het feit dat een fabriek evolueert gedurende haar levenscyclus en dat daardoor het optimaliseren en herontwerpen essentiële taken zijn in het ontwerpproces. Tenslotte behandelt de literatuur het ontwerpen van het fabricagesysteem en het besturingssysteem separaat, ondanks dat ze een grote mate van interactie vertonen. Integratie van het ontwerpen van deze twee subsystemen in één methode, kan de prestatie van het industriële systeem verhogen.

Het doel van dit proefontwerp is ontwikkelen van een methode die het ontwerpproces van een fabriek voor het bereiden en verpakken van vruchtensappen ondersteunt. Deze methode bestaat uit een structuur en een aantal gereedschappen. De structuur geeft aan welke ontwerpbeslissingen wanneer moeten worden genomen, de verzameling met gereedschappen ondersteunt vervolgens het nemen van deze ontwerpbeslissingen. De voorgestelde methode omvat zowel het ontwerpen van het fabricagesysteem als het ontwerpen van het besturingssysteem, voor zover dit het fabricagesysteem beïnvloedt. Verder ondersteunt de methode zowel het initiële ontwerpproces als het herontwerpproces.

De ontwerpmethode bestaat uit vier fasen: doeldefinitie, ontwerp van de architectuur, ontwerp van de productiemiddelen en ontwerp van de operatie. Eerst worden de doelen en de randvoorwaarden gedefinieerd. Vervolgens wordt de architectuur van het fabricagesysteem en het besturingssysteem ontworpen. Dan wordt een eerste orde benadering bepaald voor de benodigde productiemiddelen. Tenslotte wordt het ontwerp verder gedetailleerd in het ontwerp van de operatie.

De ontwerpmethode is gebruikt voor het ondersteunen van een herontwerpproject in de fabriek van Riedel Drankenindustrie. De productie is opgedeeld in drie stappen: bereiden, pasteuriseren en verpakken. Voor het bereiden van sappen wordt een groepentechnologie-architectuur toegepast, voor het pasteuriseren en verpakken wordt een lijnarchitectuur toegepast. In het ontwerp van de productiemiddelen worden beide architecturen gekwantificeerd. Om dit proces te ondersteunen zijn twee ontwerpgereddschappen ontwikkeld. Het resultaat van deze ontwerpfase is een lijst met productiemiddelen en een layout van de productievloer. In het ontwerp van de operatie wordt het ontwerp van de productiemiddelen verfijnd en wordt het besturingssysteem ontworpen. Om het ontwerp in deze ontwerpfase te ondersteunen zijn twee schedulingalgoritmes ontwikkeld en geïntegreerd in een ontwerpgereddschap. Dit gereedschap ondersteunt evaluatie van scenario's en de constructie van gedetailleerde productieschedules.

Curriculum Vitae

Jeroen Fey was born on april 3, 1970 in Maastricht, The Netherlands. He studied Mechanical Engineering at Eindhoven University of Technology, were he graduated in 1994 on the subject of scheduling in a stochastic and dynamic job shop. His Master's thesis was written in the Systems Engineering group, under the supervision of prof.dr.ir. J.E. Rooda.

After his graduation he entered the post-masters programme Computational Mechanics at the Stan Ackermans Institute. In the second year of this programme, Jeroen designed a control system for Riedel's fruit juice blending and packaging plant, which was implemented in a simulation model. In december 1996 he was appointed as industrial engineer at the Riedel juice plant in Ede, The Netherlands. At the same time, under supervision of prof.dr.ir. J.E. Rooda, he started to work on this thesis.