

Decision support for clinical laboratory capacity planning

Citation for published version (APA):

van Merode, G. G. (1994). *Decision support for clinical laboratory capacity planning*. [Doctoral Thesis, Maastricht University]. Rijksuniversiteit Limburg. <https://doi.org/10.26481/dis.19940317gm>

Document status and date:

Published: 01/01/1994

DOI:

[10.26481/dis.19940317gm](https://doi.org/10.26481/dis.19940317gm)

Document Version:

Publisher's PDF, also known as Version of record

Please check the document version of this publication:

- A submitted manuscript is the 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.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy

If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

**Decision support for
clinical laboratory capacity planning**

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Merode, Godefridus Gerardus van

Decision support for clinical laboratory capacity planning

/ Godefridus Gerardus van Merode. - [S.l. : s.n.]. - Ill.

Ook verschenen in handelseditie. - 1994. - Proefschrift

Maastricht. - Met lit. opg.

ISBN 90-9006935-6

Trefw.: operationele research / management ; laboratoria.

© 1994, G.G. van Merode

Production: Datawyse Maastricht | Universitaire Pers Maastricht

Cover design: RL design - Maastricht

Acknowledgement

The prototypes mentioned in this study and this publication were partially funded by the EC under the OpenLabs (AIM 2028) program. I would like to thank both the AIM-office and the OpenLabs consortium.

Decision support for clinical laboratory capacity planning

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Rijksuniversiteit Limburg te Maastricht,
op gezag van de Rector Magnificus, Prof. dr. H. Philipsen,
volgens het besluit van het College van Dekanen,
in het openbaar te verdedigen
op donderdag, 17 maart 1994 om 14.00 uur

door

Godefridus Gerardus van Merode

Promotores: prof. dr. ir. A. Hasman
prof. dr. J.A.M. Maarse
prof. dr. ir. drs. O.J. Vrieze

Co-promotor: dr. H.M.J. Goldschmidt

Beoordelingscommissie: prof. dr. ir. J.L.G. Dietz (voorzitter)
prof. dr. A.R. Bakker (Rijksuniversiteit Leiden /
Stichting BAZIS)
prof. dr. P.J. Brombacher
prof. dr. ir. A. Kolen
prof. dr. R.M. Leidl

1 INTRODUCTION	1
1.1 Case description	2
1.2 Problem definition	4
1.3 A decision support system	4
1.4 Methods	5
1.5 Data collection	5
1.6 Overview of the thesis	6
2 GENERAL ASPECTS OF THE DSS	9
2.1 Introduction	10
2.2 The laboratory structure	10
2.3 An example of a what-if analysis	13
2.4 Planning problems	15
2.5 Basic concepts of a Decision Support System	17
2.6 Performance criteria	20
2.7 Functions of the Decision Support System	24
3 PLANNING AND CONTROL IN THE LABORATORY	27
3.1 Introduction	28
3.2 Production control in the laboratory	28
3.3 Production control applied in the laboratory	30
3.4 Detailed control techniques	31
3.5 Aggregate control: workload control in departments	35
3.6 Integrating detailed control with aggregate control	36
3.7 Conclusion: decisions to be supported	38
4 MODELLING THE OBJECT SYSTEM	41
4.1 Introduction	42
4.2 Description of the laboratory production flow	42
4.3 The transactions in the laboratory	44
4.4 The complete communication model	46
4.5 The structure of staff control rules	47
4.6 The structure of the DSS	49
4.7 Conclusions	52
5 THE SIMULATION MODULE	53
5.1 Introduction	54
5.2 The simulation approach	54
5.3 Simulation in job shops	55
5.4 Control rules	63
5.5 Calibration and validation	68

5.6 Optimization with simulation	71
5.7 Simulation and the DSS	88
6 THE OPTIMIZATION MODEL	89
6.1 Introduction	90
6.2 The deterministic approach	90
6.3 The extended model	97
6.4 Solving the model	104
6.5 Example: Instances for Tilburg laboratory	112
6.6 The optimization model and the DSS	114
7 DISCUSSION AND CONCLUSIONS	115
7.1 Observations concerning the DSS	116
7.2 The hierarchical concept of the DSS	117
7.3 Skills level and transfer time	118
7.4 Workload control	118
7.5 Performance criteria for the laboratory	119
7.6 Influence of the case-mix on the performance	120
7.7 Future research	120
7.8 Conclusions	121
GLOSSARY	123
REFERENCES	131
APPENDICES	139
APPENDIX A: list of symbols	140
APPENDIX B: DATA OBJECTS (CHAPTER 4, 5 AND 6) AND DATA FILES	145
APPENDIX C: CONTROL.D	147
APPENDIX D: SHOPS.D	149
APPENDIX E: WORKST.D	156
APPENDIX F: PROCESS.D	168
APPENDIX G: HEURISTIC FOR THE OPTIMIZATION PROBLEM	173
SUMMARY	177
In Dutch:	
SAMENVATTING	179
NAWOORD	181
CURRICULUM VITAE	183

1

INTRODUCTION

1.1 Case description

Clinical laboratories in hospitals have to be effective and efficient. An effective laboratory provides accurate information (results of one or more tests on samples placed in a defined context) in a form and within a time-span that is requested by physicians and hospital departments. Efficient means that laboratory services are delivered at least costs.

Hospital laboratories are always facing uncertainty. Production orders (requests to perform tests and to communicate the results) are received at predictable and unpredictable time intervals. The volume and types of tests as well as the required throughput times are also often uncertain, indicating that at the beginning of a certain planning period, e.g. a day (for staff assignment) or a year (the budgeting period) the demand parameters are up to a certain extent unknown. Laboratories need a flexible production system to cope with these uncertainties. When necessary, the production system (instruments, logistics as well as personnel) should be quickly adjusted to the actual demand.

An often used but costly manner to realize such a flexibility is to allow for slack capacity (Galbraith, 1974). However, slack capacity (or redundant capacity) is not the only method available to cope with uncertainty. The effectiveness and efficiency of the laboratory can be improved by the way of queuing and processing samples, by the assignment of staff to machines, by the extent of cross-training of staff or by the way of clustering workstations/machines into shops (departments/sections). Various interdependency relationships exist between these factors. The effectiveness and efficiency of a laboratory may be improved by quantitatively modelling the impact of these factors. Such a quantitative analysis should take into account that the possibilities to control each factor are constrained. For instance, each factor has its own reaction time: while some decisions have an immediate impact, the impact of other decisions takes a longer period of time. Furthermore, there is much variation in the decision-making period: some decisions require a (very) short period of time, whereas other decisions require a longer decision-making process.

Decisions regarding the number and types of machines, staff and the organization of the laboratory typically involve long time horizons (often more than two years). This introduces a difficulty, namely the uncertainty of future demand. Information is needed on the sensitivity of effectiveness and efficiency of laboratory production to changes in the design parameters of the job shop (a department or section of the laboratory). In addition, information is required on the sensitivity of effectiveness and efficiency of laboratory production to changes in future demand. Knowledge about this sensitivity is of great importance in supporting decision-making, especially when future demand is highly unpredictable.

Future demand (volume, types of tests and required throughput times) is often difficult to predict, because the relationship between the case mix and the required laboratory production capacity is indirect and technology and protocols are changing. This is shown in figure 1.1. This figure shows the relationships between the case-mix as input and the required capacity as output, and between the available capacity for production and the demand pattern.

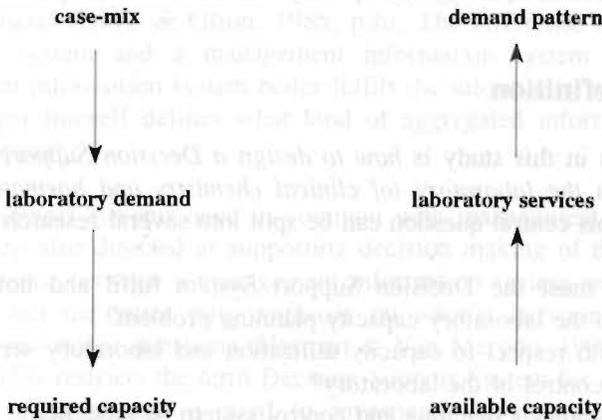


Figure 1.1 The laboratory from a hierarchical perspective.

The relations in figure 1.1 should be interpreted as follows. The case-mix of the hospital determines the demand for laboratory services. The required capacity can be derived from this demand. One can also reason the other way around. A change in the available capacity or the production process causes a change in the laboratory services which, subsequently, may result into a changing demand pattern. The decision of a laboratory to provide IVF services is a typical example. Figure 1.1 indicates that a sensitivity analysis for studying the relationship between the required capacity on the one hand and the case mix with certain service requirements on the other hand, can be performed in two directions. This is best illustrated by the following two *what if* - questions:

- If the case-mix changes, which changes are required in the capacity of the laboratory?
- What are the consequences for the demand pattern if the degree of service, the required time for delivering the test results, or the available capacity for production is changed?

Note that the problem of so-called *interdoctor variance* is left out of consideration. Interdoctor variance refers to the fact that physicians order different packages of hospital services for the same classes of patients. Although the interdoctor variance may be appreciable this is of little importance since the demand is given (thus implicitly taking into account the interdoctor variation).

In this study the efficiency of the laboratory will only concern capacity (machines and staff), not materials such as reagents. This choice is based on the fact that the volume of laboratory capacity is the main cost factor of the hospital laboratory. *Capacity* is defined in this study as available equipment and human resources. *Capacity utilization* is that part of the total volume of capacity that is actually used during a certain planning period (e.g. a day, a week or a year). The *required capacity* is the volume of capacity needed to realize the demanded production.

1.2 Problem definition

The central question in this study is *how to design a Decision Support System for capacity planning in the laboratory (of clinical chemistry and haematology) in a (Dutch) hospital?* This central question can be split into several research questions:

1. Which functions must the Decision Support System fulfil and how do these functions relate to the laboratory capacity planning problem?
2. How do goals with respect to capacity utilization and laboratory services affect the planning and control of the laboratory?
3. How can the laboratory's planning and control system be modelled?
4. What are adequate performance criteria for the laboratory?
5. Which factors affect the performance of the laboratory?
6. How do changes in the case-mix influence the performance of a given laboratory?
7. What is the optimal organization (i.e. the capacity structure) of the laboratory?

1.3 A decision support system

This study aims at the development of a decision support system (DSS). Such a system is a kind of information system. An information system is a system (an integrated whole of equipment, software, procedures, humans and data) which delivers information to someone who wants to make a decision. For decision-making in an organization in the literature often three types of information systems are distinguished (see e.g. Hasman & Van Merode, 1990, p. 377):

- transaction processing systems;
- management information systems;
- decision support systems.

In a *transaction processing system* data about the transformation system of the organization are stored and through knowledge of the context processed to information. An example is the registration of the arrival of samples in a laboratory. The resulting datafile can be processed for demand or workload

statistics. Another well-known example concerns the storing and processing of data on revenues and expenses.

A *management information system* is 'an integrated user-machine system for providing information to support operations, management, and decision-making functions in an organization. The system utilizes computer hardware and software; manual procedures; models for analysis, planning and control and decision making; and a database' (Davis & Olson, 1985, p.6). The difference between a transaction processing system and a management information system is only gradual. A management information system better fulfils the information needs of the manager. The manager himself defines what kind of aggregated information is required as decision support.

Decision support systems have in common with management information systems that they are also directed at supporting decision making of the management. The main difference between a management information system and a decision support system is that the latter puts emphasis on *what-if* questions, while the former focuses upon *what-is* questions (Hasman & Van Merode, 1990, p. 378). Van Hee (1985, p. 978) restricts the term Decision Support System for applications in which 'the information system is used to investigate developments of organizational (production) processes under the constraints given by the user of the system'. According to Vellekoop (1989, p.7) 'the aim of a decision support system is to improve the quality of decisions to be made by integrating OR techniques, human ability and computer technology in terms of effectiveness and efficiency'.

1.4 Methods

The development of our decision support system for laboratory capacity planning rests upon an evaluation of the laboratory performance under varying conditions. The approach is based on scenario-analysis. Two methods will be used for the development of scenarios: discrete-event simulation and integer programming. A discrete-event simulation model underlies the evaluation of the laboratory performance. Details of all methods will be discussed at several places in this study.

1.5 Data collection

The development of a decision support system requires data which are needed for the description of the object system and for calibrating purposes. In this study the clinical laboratory of a general hospital will be treated as the object system. Calibration is a technique for estimating system parameters in such a way that the results maximally resemble the object system's output (Kleijnen and Van

Groenendaal, 1992). This technique and how it is implemented in this study will be discussed in chapter 5.

The data used for the development of our decision support system have been collected in the clinical laboratory of the St. Elisabeth Ziekenhuis at Tilburg. This is a general hospital with 649 beds and \pm 110 specialists in 1991 and an acute care department. Special tasks of the hospital are IVF and haemodialysis. The clinical chemistry and haematology laboratory provides not only services for the hospital itself, but also for customers in the region (general practitioners, pharmaceutical industry). Clinical chemistry and haematology are fully integrated. Non-routine services are provided to other hospitals in the region and highly specialized services even to customers all over the southern part of the Netherlands.

Data on workstations, staff, demands for tests and data concerning the performance of the laboratory were collected in 1991 during a period of six months. The data were not continuously collected but during a number of arbitrarily selected time intervals (weeks). No data were collected in atypical periods (for example in holidays). This implies that our data on demands for tests as well as on the effectiveness and efficiency of the laboratory are a sample from the total demand for tests and the total performance. The representativeness of the sample has been checked by comparing the data on demand and performance with data on annual demand. The sensitivity of the performance on the laboratory data as obtained by the decision support system has been analyzed. This analysis will also be presented in chapter 5.

1.6 Overview of the thesis

Chapter 2 presents an overview of general aspects of the decision support system. After a brief analysis of the general structure of a laboratory, attention is paid to the strategic, tactical and operational planning problems in a laboratory setting. Two other main topics concern the general structure of a decision support system and performance criteria for the evaluation of scenarios.

The decision support system to be developed in this study should support capacity decisions in the laboratory. The development of such a system requires an analysis of the production planning and control system of the laboratory. This system will be discussed in chapter 3. In this discussion a basic distinction will be made between various forms of detailed and aggregate control techniques. We shall also give a brief overview of the literature on the effects of these techniques upon the performance criteria.

Chapter 4 offers a presentation of the object model that underlies our decision support model. Particular attention will be paid here to the communication flows in

the laboratory and to the transition as well as the action space for the laboratory (these terms are explained in chapter 2).

Chapter 5 discusses the design of a simulation model on the basis of the information model. With this simulation model variables and parameters determining the performance of laboratory job shops are studied. Simulation experiments are used to answer the questions on the planning and control of job shops, the influence of the configuration (available staff and equipment) and workload characteristics on the performance.

Chapter 6 discusses the problem of how to determine the optimal structure of the laboratory. In contrast to chapter 5 the laboratory as a whole will be studied. Because decisions on the number of staff and machines often require information on future demand which is normally not available, it is studied how the structure of the laboratory can be made less sensitive to changes in the demand (the 'solidness' of the structure). Another topic of analysis relates to the question how the clustering of staff and equipment into shops influences the efficiency of the laboratory and how this clustering influences the sensitivity on demand changes. This problem is dealt with by first defining a deterministic problem (the future demand is exactly predictable) and then a stochastic problem (the future demand is not exactly predictable). The definition of the stochastic problem can be considered as an elaboration of the deterministic definition.

Chapter 7 presents and discusses the main conclusions of this study.

2

GENERAL ASPECTS OF THE DSS

2.1 Introduction

This chapter discusses some general aspects of the Decision Support System (DSS) that will be developed in this study. This DSS concerns capacity planning problems in a clinical laboratory of a general hospital.

In section 2.2 we shall start with a brief analysis of the general structure of the laboratory. This analysis provides important information for the development of an analytical model that underlies the DSS. Attention will be paid to the elements that constitute the laboratory structure as well as to the concept of the laboratory throughput time that both play a major role in this study.

Section 2.3 shows how the DSS can be used for simulating the performance of a bloodgas department (a section of the clinical laboratory) under varying conditions. Laboratory management faces various types of planning problems. A DSS is intended to help laboratory management in handling planning problems.

In section 2.4 the well-known distinction between strategic, tactical and operational planning problems will be discussed as well as the application of these concepts to a laboratory setting.

Section 2.5 is devoted to a discussion of the basic concepts of a DSS. Again, special attention will be paid to the application of these concepts to practical laboratory planning problems.

The issue of performance criteria will be discussed in section 2.6. Performance criteria play a major role in the judgment of alternative scenarios that can be generated by a DSS.

This chapter ends in section 2.7 with a brief overview of the functions of a DSS. This overview may serve as a summary of this chapter.

2.2 The laboratory structure

A DSS is based upon an analytical model of the so-called object system. In this study the clinical laboratory is selected as the *object system*. The aspects of the object system we are interested in are: planning and control of samples (tests), staff and equipment.

2.3 An example of a what-if analysis

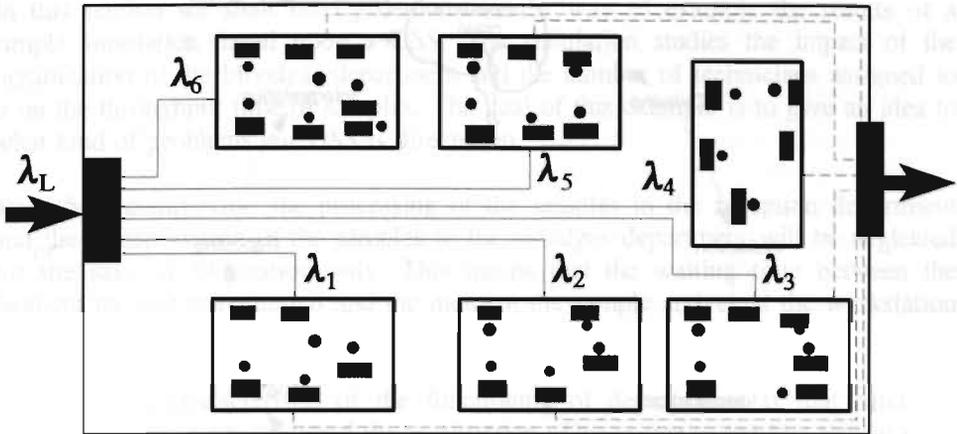


Figure 2.1 Graphical representation of the laboratory.

Figure 2.1 presents a general model of a laboratory. Indicated are workstations (the black boxes), technicians (the black dots), departments (the boxes containing the workstations and technicians, the departments are also called shops or job shops), the reception (the black box at the left side) and the quality department (the black box at the right side). The reception can also be considered as a department comprising staff and machines, but these elements are not displayed in figure 2.1. The quality department validates in addition to the job shops the test results to be reported. The left big arrow indicates the inflow of samples into the laboratory. This inflow will be expressed here as the number of samples arriving during a certain time period (e.g. one hour) at the reception. The notation for this inflow is λ_L where L stands for the *laboratory*. The lines between the reception and the departments represent the transport of samples from the reception to the departments for production. The symbol for this flow of samples is λ_i and indicates the number of samples arriving at shop i during a certain time period (e.g. one hour). The dotted lines between the departments represent the flow of information from the various shops to the *quality controller* who is responsible for a check of the test results. The right big arrow in figure 2.1 symbolizes the communication of the test results to the physicians who ordered the tests. Sometimes another senior controller adds some interpretative and advisory comments to the test results.

The total time between the arrival of the sample at the laboratory reception and the communication of the test results is called the *laboratory throughput time*. This concept is different from the total throughput time of a test request which not only comprises the laboratory throughput time but also the time for ordering tests, collecting samples and interpreting the test results. The total throughput time is represented by the well-known brain-to-brain information loop of Lundberg (1981).

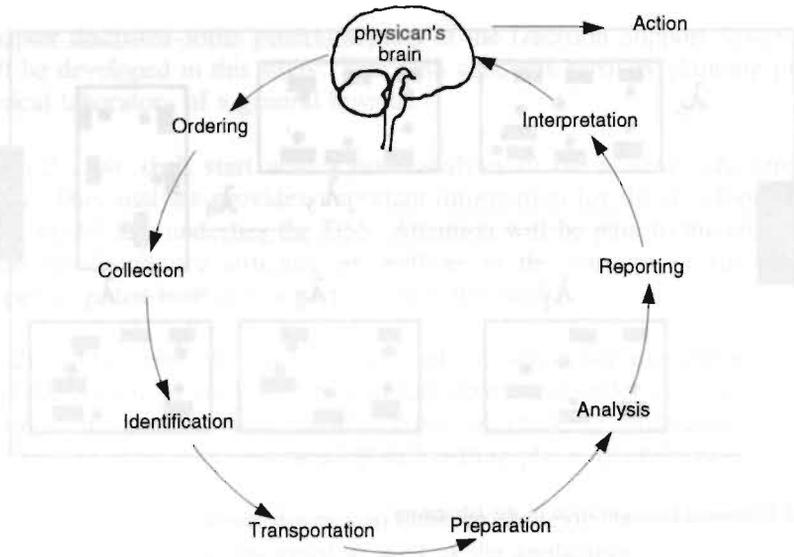


Figure 2.2 'Brain-to-brain' information loop (Lundberg, 1981).

In this study we restrict ourselves to the laboratory throughput time. An important element of the laboratory throughput time is the *shop throughput time*. The shop throughput time is the sum of the waiting time (the time the sample is waiting before being processed) and processing time. Or in symbols: $F = W_q + W_p$, where F denotes the shop flow (or throughput) time, W_q the waiting time and W_p the processing time. Often it is convenient to express the processing time as $W_p = 1/\mu_p$ where the symbol μ_p denotes the processing capacity in terms of the number of samples which can be processed during one time unit.

With regard to the organizational structure of the laboratory three levels can be distinguished. The lowest level is the *workstation* which is a machine or some other equipment to perform certain tests on samples. A *configuration* is a set of workstations. This set may contain a number of workstations of the same type or a number of workstations of a different type. A *department* is a part of the laboratory with the capability to plan its operational activities independently from other departments. The department has a certain configuration and has technicians assigned to it for a certain period (e.g. one shift). The *laboratory structure* is in this study defined as the clustering of departments at one site.

2.3 An example of a what-if analysis

In this section we shall briefly demonstrate by way of example the results of a simple simulation based upon a DSS. The simulation studies the impact of the organization of the bloodgas department and the number of technicians assigned to it on the throughput time of samples. The goal of this example is to give an idea to what kind of problems our DSS is directed to.

To reduce complexity, the processing of the samples in the reception department and the transportation of the samples to the bloodgas department will be neglected for the sake of illustration only. This means that the waiting time between the moment the test is requested and the moment the sample arrives at the workstation is set to zero.

An important characteristic of the functioning of departments is that often no technician is available to attend the workstation. This is due to the fact that in most departments the number of workstations exceeds the number of technicians. Consequently, technicians have to switch between workstations in order to achieve the requested throughput times (*due time*). From the perspective of a particular workstation such a switching behaviour implies that the technician takes intermittent vacations which means that the technician responsible for operating the workstation attends to the station in a cyclical manner. Obviously, such a vacation influences the throughput time of the samples.

Let us consider the example of a department consisting of two workstations, one for manual techniques and one for bloodgases. Let us further assume that requests for the bloodgas station always have priority to requests for the manual techniques station. The waiting time for a specific bloodgas sample is caused by the fact that other requests for bloodgas analyses arrived at an earlier point in time and/or by the fact that no technician attends the bloodgas station at the arrival of the request (the analysis at the manual techniques station has first to be decently interrupted).

In the Tilburg hospital laboratory we have measured the variables and estimated the parameters for the bloodgas station. The results are given in table 2.1.

Theoretical production rate per hour	$\mu_p = 20$
Actual maximal production rate per hour (mean)	$\mu_a = 5,38$
Number of samples per hour (mean)	$\lambda_r = 1,57$
Number of samples at peak times per hour (mean)	$\lambda_h = 2,19$
Vacation transition rate	$\gamma = 600$

Table 2.1 Parameters of bloodgas station.

The results show that the theoretical processing capacity of the station (μ_p) far exceeds the actual maximal production (μ_a). Another conclusion is that the mean inflow of bloodgas samples (λ_r) is low compared to the capacity. The same conclusion applies for the inflow of samples at peak times (λ_h). Finally, the table contains a vacation transition rate γ which is defined as the number of times a technician can switch between workstations within a fixed period of time (here one hour). The vacation rate has been set at an arbitrarily high level, because in our example the technician can immediately switch from the manual techniques station to the bloodgas station in the case of a bloodgas request. μ_p was set so high by the management of the laboratory to assure that there is backup when equipment goes down and to prevent that equipment is a bottleneck.

The actual maximal production (μ_a) and the vacation transition rate γ can both be manipulated for simulation purposes. Given λ , μ and γ and their probability distribution functions, the expected behaviour of the system can be calculated. For example, we may be interested in the waiting time, the throughput time, the utilization degree of the technicians or in the degree of satisfying the desired service levels.

A simple simulation based upon a Markov model shows that in a situation where only one technician is active, the required throughput time for super rush demand (defined as a test request with a required throughput time of less than 5 minutes) can be no longer realized if demand increases by 17.5 percent or more during peak times and by 64.0 percent or more at other times.

These results are important in dealing with the problem how to achieve an optimal utilization of technicians. Waiting times for blood gas analyses occur in two different situations. In the first situation no technician capacity is available at the time of arrival of a super rush request because other bloodgas analyses are being processed at that time. Here there is no other alternative than waiting. In the second situation technician capacity is being used at the manual techniques station at the moment a super rush request arrives. Under such circumstances re-

assignment of technicians - which presupposes central labour control within the job shop - may help to expedite the processing of the super rush demand. Re-assignment of technicians (according to rules) reduces waiting times and improves the utilization of technician capacity.

An important determinant of the waiting time in the second situation is the so-called *inter-reassignment time* (IRT). This time depends on the type of test which is produced at the moment the super-rush request arrives. Some tests can immediately be interrupted which means that the re-assignment time is near to zero. For other tests however it may take some minutes before the technician is available for re-assignment. It may also cost some time for the technician to adapt him- or herself to the new workstation. Our simulations show that longer inter-reassignment times, which are assumed to be exponentially distributed, only slightly affect mean throughput times, but substantially increase the variance of throughput times ($\sigma_{\bar{t}}$). To compensate for this effect (in order to realize the requested throughput times) the department needs more technicians (approximately 2 technicians at an IRT = 2.5 minutes, 3 technicians at an IRT = 10 minutes).

2.4 Planning problems

The goal of our DSS is to help laboratory management at various levels in solving planning problems. Laboratory management faces various types of planning problems. These problems are mentioned in table 2.2. The difference between the terms sequencing and scheduling will be clarified in chapter 3.

No	Planning problem	Management Function	Time span
1.1	scheduling orders	operational	< 1 day
1.2	sequencing orders	operational	< 1 day
2.1	assigning staff to workstations	operational	< 1 day
3.1	deciding on needed workstations	tactical	1-5 years
3.2	combining workstations to departments	tactical	1-5 years
3.3	fixing staff capacity for departments	tactical	< 1 year
3.4	staff control	tactical	< 1 year
4.1	types of analyses offered	strategic	1-5 years
4.2	volumes offered (capacity)	strategic	1-5 years
4.3	degree of services (throughput time)	strategic	1-5 years

Table 2.2 Time spans and management function of planning decisions.

Following Anthony (Anthony, 1965/1988, Anthony & Young, 1988) planning problems 1 and 2 can be considered *operational* planning problems. Planning problem 3 is an example of a *tactical* planning problem, whereas planning problem 4 is of a *strategic* nature. *Strategic planning* is the process of deciding on the goals of the organization and on the broad strategies that are followed to attain them ('what' and 'how' questions). This planning process is often irregular and non-systematic (Anthony & Young, 1988). *Operational planning* refers to what Anthony & Young call task control, that is the process of assuring that the tasks are carried out effectively and efficiently. *Tactical planning* is situated between strategic planning and operational planning. Anthony and Young also use here the term *management control*. In their view management control is directed at 'the implementation of strategies and the attainment of goals'. It has to assure 'that the organization implements its strategies effectively and efficiently' (Anthony & Young, 1988).

The framework provided by Anthony & Young shows that strategic, tactical and operational planning relate to different management functions and that these functions will often be fulfilled by different persons in the organization. In fact, each planning problem corresponds with a hierarchical level in the laboratory organization. Problems at the workstation level can be considered as operational problems whereas problems at the department level are primarily tactical problems. Decisions about the laboratory structure are of a strategic nature. The three types of planning problems also have different time horizons. For each of the planning problems the approximated time span is indicated in table 2.2.

Laboratory management has to solve operational, tactical and strategic planning problems. It is evident that operational problems occur more frequently than tactical or strategic problems, but they have less impact. In fact, it will even be quite rare that operational, tactical and strategic planning problems have to be solved simultaneously. This implies a certain simplification of the problem-solving process.

Moreover, one should realize that decisions about strategic problems act as constraints with respect to the solution of tactical or operational problems or that decisions on tactical problems limit the range of alternatives of laboratory management with respect to operational problems. For instance, the laboratory equipment is fixed in daily - operational - practice. Thus, the hierarchy in planning problems implies a simplification of the problem-solving process. This simplification is essential in the design of a decision support system. In the next section we shall discuss this hierarchy of planning problems further.

Simon (1971) has spelt out, however, that a hierarchical conception of planning problems also has a serious drawback. It may hide possible dependency relationships between the rules for making strategic, tactical and operational decisions. Rules for operational decision-making, for example, may affect the solution of a tactical problem. Rational problem-solving not only implies 'forward' problem solving (from the strategic to the operational level), but also 'backward' problem solving (from the operational to the strategic level). Strategic or tactical

solutions which are not feasible at the operational level or do not take into account operational problems are obviously doomed to fail and are thus irrational. Rational decision-making requires an optimal mix of forward and backward reasoning in order to deal with the dependency relationships between strategic, tactical and operational planning problems. The implication of dependency relationships is that the DSS should allow for both ways of problem solving (see chapters 5 and 6).

It should be noted that pure backward problem solving may be equally worse as pure forward reasoning. Suppose for instance, that a shortage of capacity (a tactical problem) is caused by an inefficient way of job scheduling (an operational problem). The range of tactical solutions depends upon the extent to which the way of job scheduling is taken for granted. In the extreme case extension of capacity may be considered an effective solution for the tactical problem of a shortage of capacity.

2.5 Basic concepts of a Decision Support System

This section discusses some basic concepts of a Decision Support System (DSS). In our view a DSS uses a planning and control model of the production system. Following the definition of a DSS by Van Hee (1985) the planning and control model can be defined by a state space, an action space, a transition space, and a goal.

The *state space* can be defined as the set of all states the object system (production system) can adopt. The definition of the state space depends upon the problem which is analyzed. The total number of possible states of an object system varies according to the hierarchical level within the laboratory. This number increases as one moves down from the strategic to the operational level.

The *action space* can be defined as the set of actions which causes a transition in the state of the object system, that is a change from one state to another one.

The *transition space* includes all possible changes (transitions) from one state to another state. The way transitions take place depends on physical laws (e.g. the technical characteristics of the production process) and/or on the planning and control system (e.g. staff assignment rules). This implies that certain transitions cannot occur or are not allowed to occur. The design of a DSS may include rules which prohibit certain transactions.

Finally, a *goal* can be defined as a specific state that has to be realized. A goal always belongs to the state space. It is assumed that actions are available to achieve the goal. A *plan* will be defined here as a set a selected actions to obtain a goal.

The concepts of the state space, action space, transition space and goal are illustrated in figure 2.3 which presents a simple Markov model of a bloodgas

station. This Markov model is an adapted version of a network developed by Ibe & Trivedi (1991).¹

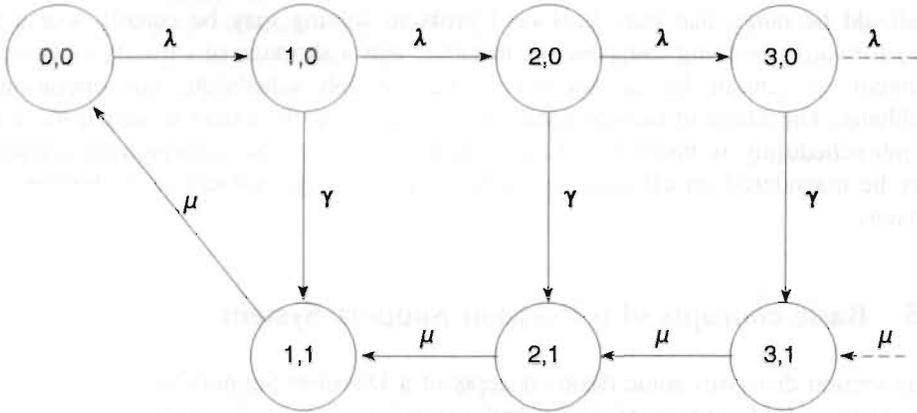


Figure 2.3 Markov model for bloodgas workstation as a vacation queuing system. The state of the system (the bloodgas station) is indicated by a circle. In each circle the first number indicates the number of samples and the second number the presence (1) or absence (0) of a technician.

A transition from one state to another state is represented by an arrow. Transitions are caused by an action. There are three types of transitions: the arrival of a sample (test request), indicated in figure 2.3 by λ ; the arrival of a technician, indicated by γ ; and the processing of the sample, indicated by μ . Transitions can be *timed* and *immediate*. While timed transitions require a certain period of time, immediate transitions do not. In figure 2.3 there are no immediate transitions, because a technician for processing the sample is absent at the moment the sample arrives (the second number is a 0). If all samples at the bloodgas workstation are processed, the technician leaves this workstation. For timed transitions the reaction time can be calculated by taking the inverse of the transition rate given by λ (arrival rate of test requests), μ (processing rate), and γ (vacation rate) (see section 2.3) respectively. Another elaboration of the basic concepts of a DSS is presented in table 2.3.

¹ A Markov-model approach was used in the beginning of the study. Later discrete event simulation was used. The results were the same.

variable	Strategic	Tactical	Operational
job queues	-	-	C
job schedules	-	-	C
staff at workstations	-	-	C
staff in the laboratory	-	C	N
equipment	-	C	N
processes	C	N	N
demand	C	N	N

Table 2.3 Status of state variables at various planning levels. C means controllable, N means uncontrollable, and - indicates that the variable is not relevant.

Let us consider a tactical planning problem. According to table 2.3 the state space corresponding to this problem contains the state of the staff and equipment of the laboratory, the production process as well as the demand for laboratory services. The action space for a tactical problem, however, is more limited. The decision-maker can only consider actions with respect to staff and equipment in the laboratory (indicated by a C); he cannot manipulate the production process and the demand for laboratory services (indicated by a N). The same restriction applies for the goal(s) that can be realized: no goals can be formulated with respect to the production process as well as the demand for laboratory services. Table 2.3 also shows that the state space for operational planning has more states than the state space at the tactical level. But again, only a restricted subset of these states can be controlled at the operational level. This is due to decisions at higher hierarchical levels. Another observation is, that decisions at the tactical level affect both the state and action space for decisions at the operational level. While the staff and equipment in the laboratory can be manipulated at the tactical level, they cannot at the operational level (C's have become N's).

The decision support system to be developed in this study is primarily intended to help laboratory management in solving planning problems. The DSS must be capable to evaluate alternative scenarios.

There are two basic strategies for scenario management. The first strategy is to manipulate the state of the object system (here the clinical laboratory). The second strategy aims at a manipulation of the environment of the system. Of course, the first and second strategy can also be combined into a mixed strategy. We shall only discuss here certain aspects of the first and second strategy.

The essence of the first strategy is that the planner considers a number of alternative changes in the object system itself, for instance changes in the size of the staff, the number and type of workstations, or in the rules for assignment of staff to workstations. These changes are the result of actions. Scenario studies are

performed to investigate the effects of these changes on the performance of the object system (the concept of performance will be discussed below).

The essence of the second strategy is that the planner considers a set of changes in the environment of the object system. These changes may be achieved in several ways. One option is to assume changes in the volume or type of laboratory tests and to analyze the effects of these changes on the performance of the object system. Another option is to manipulate the required service levels to be satisfied and to analyze to what extent the object system is capable to accomplish these levels. It should be noted, that the object system only initially remains unchanged in the second strategy. If scenarios suggest that a system cannot satisfy the required service levels under the condition of changes in the volume and types of test, changes in the object system have to be considered in order to satisfy the required service levels. Consequently, manipulations of the object system must be considered in order to satisfy changes in the required service level. In other words, the second strategy may eventually result into changes of the object system.

In this study both strategies as well as combinations of them will be analyzed (see chapters 5 and 6).

2.6 Performance criteria

In the previous section we have seen that performance criteria play an essential role in laboratory simulation. In this section some aspects of performance criteria will be discussed.

A basic criterion in this study is *effectiveness*. Effectiveness is defined as the extent to which the due-times are realized. In this study due-times - that is the standard to be satisfied - are considered exogenous (see Janse & Kateman (1983) for a discussion about the determination of due-times). Due-times strongly depend upon the service-policy of the laboratory and the hospital as well as the urgency of the test requests.

An individual job (test request) is performed effectively if:

$$A \leq d_i; \quad \text{eff}_i = 1 \quad (2.1)$$

where A is the actual time at which the test result is available, d is the due-time and the subscript i denotes the individual job.

Similarly, an individual job is not performed effectively if:

$$A > d_i; \quad \text{eff}_i = 0 \quad (2.2)$$

For a number of processed jobs during a period E the total effectiveness of the laboratory can be defined as follows:

$$Eff_E = \sum_{i \in S_E} eff_i \quad (2.3)$$

where i is a production order and S_E is the set of all production orders during period E. The value of Eff_E may vary between 0 and $|S_E|$. The job shop or laboratory is fully effective if:

$$\frac{Eff_E}{|S_E|} = 1 \quad (2.4)$$

Performance criteria may be formulated with respect to individual jobs, to shops (or departments) and with respect to the capacity in terms of human resources and equipment).

Job related criteria

The *job related* performance criteria are: mean flow time (\bar{F}), variance of flow time (σ_f^2), mean lateness (\bar{L}), variance of lateness (σ_l^2) and proportion of jobs late (P). The mean flow time (\bar{F}) is the mean time that jobs spend in the production system. The flow time of a job i is the sum of the time a job has to wait in the shop (W_i) and the time a job is actually processed (P_i). Thus $F_i = W_i + P_i$ or $F_i = C_i - r_i$, where C_i is the completion time of order i and r_i is the release time of this order. Observe that C_i and r_i are time points at the time axis, while w_i and P_i are time periods. In chapter 3, the application of these performance criteria in planning problems will be discussed.

Shop related criteria

The *shop related* characteristics are the idle time of the capacity available at the shop and the total labour transfer between workstations (T). The idle time measures the degree of under-utilization of shop capacity and total labour transfer measures the frequency of labour switching between different machines or workstations during a given period of time. A high labour transfer (for instance made possible by training and education) indicates a high flexibility, which generally improves the performance of e.g. the mean flow time. But it also implies high costs because during transfer no work can be done and supervising as well as salary costs increase. Whether a particular transfer rate in a certain department should be considered positive for the performance should be checked by a sensitivity analysis.

Capacity costs

Performance criteria are also defined with respect to *capacity costs*. In this study we are only interested in the cost of labour, both with regard to the volume (number of workers) and the level of training. The costs of labour will be measured by the number of technicians and the level of training of these technicians. The ratio between the input of a shop (capacity costs) and its output (performance of a shop) is called the *efficiency* of the shop.

Other distinctions

In addition to the distinction between job related criteria, shop related criteria and capacity costs two other distinctions are important: the distinction between short-term and long-term criteria and the distinction between detailed and aggregate criteria.

The distinction between short-term and long-term criteria is related to our earlier discussion on the variety of time horizon in planning problems. Whereas some problems have a very short horizon (for example one hour), others have a much longer horizon. *Short-term* performance criteria refer to problems with a short time horizon. For instance, the due-times for certain types of tests can be set at 10 minutes (during surgery or delivery). It is important to note that the extent of uncertainty about the future state of the object system is relatively low in the case of short-time horizons, because many elements of the object system and its environment are fixed.

Long-term performance criteria are related to problems with a longer time horizon (tactical and strategic problems). These criteria may refer to the size of the staff or staff utilization. Both types of criteria will be used in this study.

The distinction between detailed and aggregate criteria strongly correlates with the distinction between short-term and long-term criteria. Most detailed criteria are also short-term and most aggregate criteria are also long-term. But they are not the same. One may apply aggregate criteria for short-term problems. This is for instance the case if staff is allocated to workstations without taking into account the specialized skills of the staff. Similarly, detailed criteria may be used in long-term problems. An example is the criterion that two years from now a specific capacity or processing technique should be available in the laboratory.

It is evident that a laboratory (shop) performance is dependent upon a set of internal and external conditions. A laboratory that is effective at time t_1 may for instance become ineffective at time t_2 due to an increase in demand or to shortage of staff. The objective of scenario-analyses in this context is to get information about the performance of the shop under varying conditions. The external environment may change as well as the internal environment. Scenario-analysis may reveal that the shop-performance depends upon the specific combination of external or internal conditions. Several types of results will be discussed here.

A first possible result is that a specific shop configuration dominates on all performance criteria in all scenarios. In this situation the choice is obvious. Choice becomes more difficult, if a dominant shop configuration is absent. Two characteristic situations can at least be distinguished: more than one criterion or

	c_1	c_2
w_1	$p_1 > p_2$	$p_1 < p_2$
w_2	$p_1 < p_2$	$p_1 > p_2$

Table 2.4 Example of non-dominant configurations for two scenario's.

more than one scenario. Assume we have two criteria c_1 and c_2 for evaluating configurations p_1 and p_2 in scenario's w_1 and w_2 . Assume that the performance of the configurations is as given in the table 2.4.

In this table the two characteristic situations are presented. In the first situation a dominant configuration is absent, because according to criterion c_1 configuration p_1 is to be preferred and according to criterion c_2 configuration p_2 . The strategy of a decision maker would be to define an evaluation function that assigns weights to the evaluation criteria (see for example Verbeek, 1991).

The second characteristic situation is that no configuration performs well in all scenarios. In scenario w_1 configuration p_1 has to be preferred according to criterion c_1 , but in scenario w_2 this is configuration p_2 . Here the decision maker has to choose a strategy to evaluate various scenarios. An approach would be to weight scenarios. An example is the expected value approach. In this approach for each scenario the probability of becoming true is multiplied by the costs or profits of each configuration. For each configuration the total costs and profits are calculated.

Also alternative functions for evaluating configurations under different scenarios may be formulated, for example by taking into account the decision makers appetite for risk. This type of risk may be called *tactical risk* which should be distinguished from *operational risk*. Operational risk is represented by performance criteria mean lateness (\bar{L}), variance of lateness (σ_L^2) and proportion of jobs late (P). These criteria only measure the risk of violating the effectiveness of a given configuration. They indicate to what degree orders are not produced according the goals of the operational planner. Tactical risk refers to the situation of not choosing a configuration which would have had a better performance under certain demand characteristics (the actual scenario). Tactical risk can evidently influence the strategic performance.

Our decision support system will evaluate scenarios in two respects. First, it proposes only configurations which are able to process the required workload. Second, with respect to the problem of clustering workstation into departments it tries to minimize risk if no trade off between risk and costs exists.

The choice of the configuration will then depend on the attitude of the decision maker to risk. We will assume that the decision maker's appetite for risk is given. Under this assumption the *portfolio-model* approach (see e.g. Harrington, 1987) can be used. In that approach a rational decision maker will choose that plan or configuration that provides the highest return for a given level of risk or that one that offers the least risk for a given level of return.

If there is a trade-off between risk and cost due to uncertainty in demand we propose that the user uses the *plan and scenario management* function to make several combinations of scenarios (demand characteristics) and plans (configurations and number of technicians). With the help of this plan and scenario management function the user is able to define optimal combinations in an interactive way. A multi-criteria analysis could help him here. Also game theory (especially cooperative game theory) could be often useful here since when planning the processing of orders there is some freedom with regard to the allocation of capacity over time. Such analysis methods will not be built into our DSS.

2.7 Functions of the Decision Support System

According to Verbeek (1991) and Eiben (1989) a DSS has the following basic functions:

- data manipulation
- representation of the plan
- generation of a plan
- manipulation of a plan
- evaluation of a plan
- selection of a plan
- plan and scenario management

On the basis of the earlier described analysis of the decision processes in the laboratory we arrive at the following configuration of the DSS.

In our DSS a decision-maker is able to specify his problem by defining demand characteristics, the type of workstations, skill level of technicians and organizational characteristics (*data manipulation*).

The function *representation of the plan* provides the decision maker a specification of workstations, departments, operational rules and the number of available technicians.

The function *generation of a plan* yields a plan for achieving one or more goals. A goal can be conceived as a desired state. The function also checks the feasibility of the goal given a starting state. In our DSS a simulation and an optimization module will be developed for plan generation. The simulation module provides a plan that contains a certain combination of operational rules and a configuration of workstations; in the optimization module the plan includes a clustering of workstations into departments.

The function *manipulation of the plan* simply means that the DSS offers the option to define alternative plans. In this study various kinds of plan manipulation will be discussed.

The *evaluation* function assesses the quality of each generated plan. For assessing plans a set of criteria has to be defined (see section 2.6). Our DSS produces a set of performance parameters for each configuration and each scenario.

The *selection* function of the DSS results into a proposal for decision-making. In this study proposals will be formulated for the shop level as well as the laboratory level.

The *plan and scenario management* function of the DSS allows the decision-maker to specify any configuration and any scenario.

A final remark should be made about the output of our system: ready-to-use solutions are not to be expected. The DSS is directed to giving decision makers insight into capacity planning problems and possible solutions on the basis of which the decision maker can decide. Such insight gets more and more valuable in a time with increasing budgetary pressures and changing client - producer relationships.

3

PLANNING AND CONTROL IN THE LABORATORY

3.1 Introduction

In chapter 2 we discussed general aspects of the decision support system to be developed. We also dealt with the relationship between the decision support system and the planning and control system of the laboratory. Compared to industrial production systems the laboratory has to fulfil other requirements. This is due to the following characteristics of laboratory job shops that industrial job shops usually do not have:

- the actual demand for services for the operational planning period is unpredictable;
- a part of this demand is ordered through rush requests;
- external due-times are given and may not be violated;
- technicians can work on different workstations simultaneously.

These characteristics lead to the conclusion that a separate investigation of the success of decision support systems in a clinical laboratory environment is worthwhile. In this chapter a more detailed theoretical analysis of the planning and control system of the laboratory will be presented.

The first step consists of a further elaboration of the production control system of the laboratory. Section 3.2 introduces new concepts and strategies for structuring the production system. The application of these concepts and strategies is discussed in section 3.3. In section 3.2 an important distinction is made between detailed and aggregate control techniques. Detailed control techniques are discussed in section 3.4 and the aggregate control techniques in section 3.5. The integration of both control techniques is discussed in section 3.6. Section 3.7 presents some conclusions. These conclusions particularly focus upon the implications of the theoretical analysis for the construction of the decision support system.

3.2 Production control in the laboratory

Our Decision Support System should support capacity decisions in the laboratory. These decisions are strongly interrelated with other types of decisions, such as scheduling and sequencing of samples or staff assignment. In order to prevent suboptimal decision-making, the Decision Support System has to take into account the interrelation between capacity decisions with these other types of decisions. More precisely, the structure of the production control system should be defined.

Production control (or the production control system) is defined by Van Rijn (1986, p.41) as 'all activities which concern the adjustment in time and place of materials and capacities to each other, so that goals with regard to the product and to the production process are fulfilled'. Bertrand and Wijngaard (1986) consider complexity reduction for production management as one of the main goals of a production control system.

There are two strategies for complexity reduction, decomposition and aggregation. *Decomposition* means that the production system is divided into subsystems

(laboratory management, laboratory departments). Each subsystem performs one or more decision functions for which local goal variables have to be defined (Bertrand & Wortmann, 1981; Simon, 1981).

Example: In the laboratory efficient utilization of technicians and meeting demand are the two most important goals. Meeting demand, that is the realization of due-times of test-requests, is a goal at the department level: the haematology section is responsible for realizing due-times of haematology requests. Effective utilization of technicians, however, is the responsibility of the management of the laboratory: technicians should be assigned to departments in such a way that they are efficiently employed.

In this example vertical decomposition is applied for the assignment of staff and horizontal decomposition for the realization of the due-times (a department is only responsible for the realization of the due-time in its own part of the production process, given its staff assignment). In the case of *horizontal* decomposition the production process is split up into a number of pooled, sequential or reciprocal subprocesses (Mintzberg, 1979). *Vertical* decomposition is connected with the concept of hierarchy. Because horizontal decomposition asks for coordination between the various subprocesses, horizontal decomposition presupposes vertical decomposition (unless one assumes a pure market model).

With respect to horizontal decomposition the term *decouple points* is often used. Decouple points indicate the border of control and are used to decouple subsystems (for example departments). Decouple points are stocks and queues. The major drawback of decouple points are the incurred costs.

The second strategy for complexity reduction is *aggregation* (Hax & Candea, 1984, Bertrand & Wijngaard, 1986, Van Rijn, 1986). Aggregation means that different products or capacities are considered identical from a certain perspective. The advantage of aggregation is that the information concerning the actual or the expected state is more simple.

Example: In the laboratory technicians are assigned to departments twice a day (for the morning and the afternoon). Technicians are assigned to departments on the basis of the expected workload expressed in minutes processing time for each department. At the departments test requests are assigned to workstations and technicians are asked to process tests requests using information (such as due-time, type of test) considering each individual test-request.

In this example aggregation reduces the complexity of assigning technicians: the expected workload expressed in process time is the only information needed. At the department's level aggregation is not used: individual characteristics of test requests are important and detailed control is needed.

3.3 Production control applied in the laboratory

The laboratory production process is divided (horizontally decomposed) into four different phases: receiving and registering test requests (phase 1), the actual processing of specimen (phase 2), the quality control on the specimen (phase 3) and the communication of results to the requesters (phase 4). These production phases are performed by different departments. Phase 1 is realized by a reception department, phase 2 by analysis departments, phase 3 by a quality controller. Phase 4 is in principle a completely automated process. In some cases a senior technician or the clinical chemist adds his advice to the test results.

The following decouple points for test request/specimen (sample buffers) result from this organization: reception buffer, department buffer, workstation buffer, quality control buffer and a buffer before results are communicated.

Variations in this horizontal decomposition are dependent on the type of test request, the time and the laboratory's department. Part of the processing of specimen may for instance take place in the first phase. It is also possible that the actual production process has more than four phases because of re-scheduling of specimen.

Vertical decomposition is needed because the activities of the horizontally decomposed departments have to be coordinated. Vertical decomposition may also be necessary within departments. This is the case when a department is subdivided into several workstations where one or more technicians simultaneously process samples. A sample may pass several production phases within the department and, dependent on the assignment of the production activities within the department to workstations, a sample may pass several workstations. In the laboratory under study a sample usually passed only one workstation and sometimes (often in case of rescheduling) two or three workstations. With regard to vertical decomposition we can discern the following layers in the laboratory: the laboratory management, the shop managers, and the workstations.

The applicability of aggregate control depends on the possibilities for treating test requests or capacity as identical from a certain planning perspective. In the case of assignment of staff to departments, technicians can be considered identical if they are cross-trained or nearly cross-trained. Test requests can be considered identical with respect to their workload expressed in needed process-time. If unique characteristics of staff or test requests are important, detailed control should be applied. For example, if certain scarce skills are indispensable for performing specific tests, detailed control is necessary.

For staff assignment as well as for test request assignment, the real question is not whether aggregate or detailed control should be applied, but rather the degree to which it is possible to apply aggregate control (and as a consequence the degree to which detailed control is necessary). In the laboratory under study it appeared that the technicians are sufficiently cross-trained to consider them as identical.

Uncertainty in demand is due to variations in demand. Variations in the aggregate workload of a certain department are dependent on the variations in the workload of the various department's processes. If the variations in the workload for individual processes are negatively correlated, the aggregate workload for a certain department can be more accurately estimated than in a situation where they are positively correlated. In other words, the accuracy of the forecast of the aggregated workload of departments is dependent upon the covariation between the demands for processes within each department. Successful aggregate control presupposes such an assignment of production processes to departments that the variations in aggregate workload are smaller than the sum of variations of the workload for each individual process. How this can be realized will be discussed in chapter 6.

3.4 Detailed control techniques

3.4.1 Introduction

If the workload of the department, the workstations and the number and types of technicians are given, the actual performance of a department is determined by rules that plan and control the operations. As discussed in chapter 2, the performance indicators are mean flow time (\bar{F}), variance of mean flow time (σ_f^2), lateness (\bar{L}), variance of lateness (σ_l^2), proportion of jobs late (P) and labour transfer (T).

The types of rules for planning and controlling operations that can be found in the literature are:

- a. scheduling and sequencing rules for processing specimen at the workstation;
- b. rules for assigning staff to workstations;
- c. rules to regulate staff reassignment when staff at workstations becomes available.

In the simulation model these types of rules will be implemented. It depends on the type and size of the department which rules or combinations of them perform best. In our experiments with these rules we will consider them as *experimental factors* in simulation experiments. The shop configuration and workload will be called *scenario factors* in this study. First, we will discuss each experimental factor.

3.4.2 Scheduling and sequencing of specimen

There is an important difference between scheduling and sequencing. Sequencing means that the planner determines some order in which the samples will be processed. There are many different orders (sequences) conceivable (see below). The essential point is that sequencing does not require exact information about arrival times of test requests and due-times. Scheduling, on the other hand, is more specific than sequencing, because it also includes time-tabling. It requires exact information about the arrival time and external due-time of the test requests

(French, 1982). This information is often not available in a laboratory setting because at any moment rush orders may arrive.

In the literature various types of sequencing rules have been analyzed. Weeks & Fryer (1976) discuss the effects of the First Come, First Served rule (FIFO), the Shortest Processing Time rule (SPT) and the Least Slack per Operation rule (LSO) in a dual-constrained job shop (two scarce resources e.g. labour and equipment are considered). The FIFO-rule stipulates that jobs are processed in the order of their arrival at the shop or the workstation. The SPT-rule gives priority to orders with the shortest processing time. The LSO-rule calculates the priority of orders as follows. First for each order the due-time tolerance is calculated by dividing the time span between the moment of the assignment decision and the due-time of the sample by the sum of times of the operations still to be done for this order. Next the due-time tolerance is divided by the number of remaining operations. The outcome of this procedure determines the priority (lower numbers have higher priority). Weeks & Fryer conclude that the Shortest Processing Time rule (SPT) performs best in terms of mean flow time, mean lateness, proportions of jobs late and total labour transfers. The FIFO-rule performs best when the variance of flow-time has to be minimal. The LSO rule reduces the variance of lateness maximally.

Although Weeks & Fryer studied their hypothetical shop with a fixed utilization rate (90%), Huang et al. (1984) came to the same conclusion on the superiority of the SPT-rule for various shop utilization rates (70%, 85% and 99%). It appeared that the superiority of the SPT-rule is less pronounced under the condition of low utilization: the performance of the SPT-rule increased rapidly with higher utilization rates. Because the mean flow time increases under higher utilization rates (higher workload), the performance of the SPT-rule even becomes much better under this condition when compared to the alternative sequencing rules. This is an important fact to account for in scenario analyses.

	\bar{F}	σ_f^2	\bar{L}	σ_L^2	P	T
FIFO	+/-	+	+/-	+/-	+/-	+/-
SPT	++	+/-	+	+/-	+	+
LSO	+/-	+/-	+/-	+	+/-	+/-

Table 3.1 Performance of sequencing and scheduling rules.
 Meaning of symbols: ++ very positive effect; + positive effect; +/- no effect; - negative effect; -- very negative effect.

Sequencing rules for orders can be combined with *due-time assignment rules*. This type of rules has not received much attention in the literature. The probable reason for this is the fact that in many theoretical studies no distinction is made between the internal and external due-time. According to Bertrand et al. (1990) internal due-time assignment can be an important tool for workload control. Its utility is especially useful for the stabilization of the workload. For a certain proportion of the orders the internal due-time is set before the external due-time and for another

proportion after the external due-time. In the last situation it is often assumed that customers have no objection to late delivery.

Weeks & Fryer (1976) have compared tight and loose due-time assignment rules. In contrast to loose due-time assignment rules, tight due-time assignment rules allow for little freedom in postponing the actual processing of orders once released. Tight rules perform well with respect to mean flow time and variance of flow time and bad for mean lateness and percentage of jobs late. The opposite is true for loose due-time rules. Especially the effect on the mean flow time is very large.

	\bar{F}	σ_f^2	\bar{L}	σ_L^2	P	T
Tight	++	+	-	+	-	+/-
Loose	--	-	+	-	+	+/-

Table 3.2 Due-date assignment and performance.

3.4.3 Staff control and assignment

For staff control and assignment two sets of rules are important. The first set relates to labour assignments. The rules belonging to this set indicate who is in control of labour assignment and what kind of assignments and reassignments are allowed. More specifically, they handle the problem of how technicians are to be assigned to job shops or to workstations. The second set of rules only deals with the assignment of personnel to separate workstations within job shops. This second set of rules has been left out of consideration in this study, because all technicians were cross-trained.

The relation between *staff control* (centralized and decentralized) and shop performance is discussed by Russell *et al* (1991), by Fryer (1975) and by Weeks & Fryer (1976). All results indicate that centralized control (on the shop level) is far superior to decentralized control. The idea behind centralized in contrast to decentralized staff control is that once staff is assigned, it can be reassigned during the production period. If no reassignment is allowed, control over staff during the production period is in fact absent. Then in each production period the equipment/labour ratio at the workstation remains constant.

In the above mentioned literature, only one rule for centralized staff control is studied: when a technician completes a job he/she is immediately available for reassignment. Also only one rule for decentralized staff control is found: a technician can only be reassigned when there is no queue with jobs at his present workstation. As might be expected, the total labour transfer under the condition of central staff control is high compared with decentralized staff control. For all other performance parameters centralized staff control is better than decentralized staff control. But, if the advantage in performance of centralized staff control is small,

decentralized control may be preferred if total labour transfer is an important performance measure. The higher the total labour transfer, the more intense managerial organisation of the job shop is required.

	\bar{F}	σ_f	\bar{L}	σ_l	P	T
CSC	+	+	+	+	+	—
DCS	-	-	-	-	-	++

Table 3.3 Merits of centralized (CSC) and decentralized staff control (DSC).

Staff assignment appears to have a significant effect on the performance of the shop. According to Russell *et al* (1991), Huang *et al* (1984), Fryer (1975) and Weeks & Fryer (1976) the rule according to which technicians are assigned to workstations or machines with the longest queue (LNQ) and/or the rule according to which technicians are assigned to the workstation with the largest processing time enqueued (LPT), perform best with respect to the flow-time and total labour transfer. Another favourite labour assignment rule, namely the rule which assigns labour to the workstation with the job in queue with the longest waiting time (LWT), gives the best performance with respect to the variance of flow-time. One should however be careful in drawing conclusions because the statistical significance of the effects appears to be very weak in all studies quoted.

	\bar{F}	σ_f	\bar{L}	σ_l	P	T
LNQ	+	-	+/-	+/-	+/-	-
LPT	+	-	+/-	+/-	+/-	-
LWT	+/-	+	+/-	+/-	+/-	+/-

Table 3.4 Staff assignment rules.

3.4.4 Interaction between experimental factors

In the literature two interaction effects between factors are considered important, namely the interaction between sequencing rules and labour assignment rules and the interaction between due-time assignment rules and sequencing rules.

Concerning the interaction between sequencing rules and due-time assignment Weeks & Fryer (1976) found that the variance of flow-time, variance of lateness and the proportion of jobs late depends on the tightness of the assigned due-times. The ranking according to the optimality of the rules does not change, however. The SPT-rule remains optimal especially with regard to the proportion of jobs late.

When loose due-times are assigned, the SPT and the least slack rule perform almost equally, and the least slack rule performs then even better than the SPT-rule

in terms of variance of lateness. The superiority of the SPT-rule is apparent when the interaction with labour assignment is studied (Huang *et al*, 1984; Russel *et al*, 1991): the SPT-rule performs best under all labour assignment rules. The best shop performance (in terms of mean flow time) is realized if SPT is used in combination with the LNQ rule.

	\bar{F}	σ_f^2	\bar{L}	σ_L^2	P	T
SPT/ LOOSE	+/-	+/-	+/-	+/-	+/-	+/-
SPT/ TIGHT	+/-	+	+/-	+	+	+/-
LSO/ LOOSE	+/-	+/-	+/-	+	+/-	+/-
LSO/ TIGHT	+/-	+/-	+/-	+/-	+/-	+/-
SPT/ LNQ	+	+/-	+/-	+/-	+/-	+/-

Table 3.5 Interaction between factors (rules).

3.4.5 Interaction between experimental factors and shop size

The interaction between shop size and rules is discussed by Fryer (1975). He studied the interaction between shop size, sequencing rules, labour assignment and staff control. It turned out that only interaction effects between shop size and staff control are significant. Other interaction effects are not present, not significant or were not studied. The shop size only affects the degree of influence of the staff control rules, but not the quality order between them. From this the conclusion can be drawn that the shop size is not an important variable with regard to the job-related performance measure. However, the shop size may have an important influence on shop related performance measures as idle time of capacity (especially of technicians). This effect may be very significant when labour (through staff control) is flexible.

3.5 Aggregate control: workload control in departments

Internal due-time assignment can be used for the control of the workload of a department or a workstation. The usefulness of workload control can be increased by making a distinction between the arrival time of a test request at the department and the release of that order. The release time of an order is defined as the time of approval that an order can be processed. The distinction between the shop level and the workstation level is important in this respect. It is decided at the shop level which specimen are released and thus allowed to be *queued* at the workstations.

Specimen which have not been released are *stocked* at the shop level. The difference between stocked specimen and queued specimen is that specimen which are stocked are released by a decision, whereas specimen in a queue are processed as soon as capacity becomes available. Workload control is a technique that uses information about actual workload, forecasted demand and available capacity to release and process orders. It considers the job shop as a black box. Forecast information is used to adapt the release rate of stocked specimen to the expected arrival rate of rush orders. When this arrival rate is expected to be high, the release rate is low, and visa versa. This type of workload control has the advantage that a constant and high level of capacity (labour) utilization can be realized even if rush orders are allowed.

How can we apply this concept of workload control? The first step is to determine how much capacity for a certain period (e.g. the next two hours) should be reserved for test requests with a given due-time. The next step is to determine the maximum allowed utilization degree of technicians. The difference between the maximum utilization degree and the capacity needed for test requests with due-times determines the release rate from the workload stock.

Example: For the next two hours at the bloodgas department the rate of arriving requests for bloodgas analyses with a required throughput time of 5 minutes is 2.17 specimen per hour. With the exception of requests for blood gas analyses, all orders are stocked in the workload stock. With three technicians available and with a maximum allowed utilization degree of 90%, 77.9 % or 140.3 minutes workload ('mean aggregate workload') other than bloodgas analyses can be processed maximally per hour at this department. If Inter-reassignment times (see chapter 2) and/or transfer times are taken into account the released workload will be substantially less.

In this example workload control is used as an aggregate control technique combined with detailed control. Detailed control is applied to test requests for blood gas analyses, aggregate control is applied to all other test requests. Detailed control is more accurate than workload control. For the specimen of a certain department a mean workload per test request is calculated. In workload control this mean workload is used to calculate the release rate. This implies that the actual released workload in a certain period may differ from the number of test requests multiplied with the mean workload per order and that capacity is not used in the most effective way.

3.6 Integrating detailed control with aggregate control

3.6.1 Introduction

If no planning freedom exists, it is not possible to control the workload since the due-times of each sample have to be taken into account. Therefore in this situation only detailed control rules can be used. This will require flexibility of staff. Moreover, slack capacity is required to realize the external due-times. When there

is planning freedom, production smoothing could be applied. As workload control is an aggregate control technique, it should however be applied carefully.

We have already discussed the possibilities for the aggregation of the workload. The topic how and why to implement workload control is not yet discussed. First we point out why to apply workload control. Next attention will be paid to how to apply workload control.

3.6.2 Combinations of workload control and detailed rules

The most obvious goal of workload control is to stabilize the workload for the job shop at a certain level. Stabilization of workload is important for efficient technician utilization. Suppose that at a certain job shop for a certain staff assignment period only one technician is needed who spends only 20 percent of his time on rush test requests. In this situation it would be very efficient to have a stock of samples available to increase the utilization degree of technicians up to for example 90 percent. This application of workload control stresses the importance of balancing the workload between different staff assignment periods but also between job shops (as the released workload is strongly dependent on the available technicians at the departments).

The second goal of workload control is to assure that due-times are met and that the processing capacity is utilized as efficient as possible. To discuss this goal we first discuss two typical situations.

The first situation is a job shop where rush test requests will arrive. In this situation the planning freedom is limited and slack capacity has to be available for the realization of the due-times. Here it would be useful to introduce a rule which assigns staff to the workstation which has a job in queue with the highest priority (HPJ).

The second situation is a job shop where only orders with low priorities are delivered. As there is much planning freedom, especially the SPT-rule should be applied to realize high efficiency rates.

Most laboratory job shops are positioned somewhere between both situations. There will be moderate planning freedom. Rush orders require the application of the HPJ-rule and lower priority orders allow for the application of the SPT-rule.

Due to the uncertainty about the arrival of rush test requests, the decision to process other orders than rush test requests depends on the situation on the shops floor. If technicians are (or are expected to be) busy with rush orders no lower priority orders will be released. Thus the HPJ-rule is applied. If however technician capacity is available and no rush orders are expected, then the lower priority orders should be processed. This argues for the application of the SPT-rule.

In order to combine the SPT-rule with the HPJ-rule it is necessary to have samples stocked which can be processed according to the SPT-rule. Here detailed workload control is applied.

3.6.3 Implementation of operational control in departments

The planning of workload is important for two reasons. First for the assignment of technicians to job shops, and second for efficient utilization of capacity at the job shops once capacity is assigned. The assignment of technicians to job shops is discussed in chapter 6. The implementation of workload control at the shop level is detailed: to apply the SPT-rule only jobs for specific processes are released.

Remains the problem which process time the SPT-rule should take into account. Two characteristic situations can be distinguished. One is the situation where the process time of the workstation and the technicians are identical or the technician time is longer than the workstation time. Here there is no problem. The SPT-rule selects samples to be processed at a certain workstation and staff is assigned to that workstation. In the other situation staff has to attend to the workstation only a fraction of the time of the total process time of the workstation. A typical case occurs where the technicians have only to attend the workstation for the preparatory activities. The SPT-rule has only to take into account the process time of technicians. The maximum workload that can be processed in this situation will however also depend on the process-times of workstations. If a job shop has only to perform processes for which the technician time is a small fraction of the workstation processing time, the SPT-rule creates free-technician time by first assigning technicians to workstations with small process-times for technicians (compared to the workstation process time), the SPT-rule takes then advantage of the free-time of technicians by assigning them if possible also to other workstations. Characteristic of the SPT-rule in our implementation is that more free-time (where technicians can work) is created compared to other rules.

Our DSS will use aggregate information for clustering the workstations into departments. Workload control will be applied in a detailed manner to guarantee the effective (applying the HPJ-rule) and efficient (applying the SPT-rule) functioning of the job shop.

3.7 Conclusion: decisions to be supported

Laboratory job shops have to fulfil other requirements compared to industrial job shops (see section 3.1). As a result planning rules performing well in industrial job shops are not guaranteed to be useful in the laboratory. An example is the SPT-rule. If used it should be applied in combination with the HPJ-rule. These specific requirements have impact on the functions of the DSS.

The functions a DSS has to fulfil can be derived from the decisions that have to be supported. These decisions were discussed in the preceding sections in this chapter.

In this section we will discuss the functions of the DSS following the structure of our previous discussion.

An important strategic decision concerns the processes the laboratory should perform. Such a decision can be based upon an estimation of the demand for laboratory services. After these decisions have been made, the number of staff and the number of workstations have to be determined. Because the way workstations are clustered into departments influences the number of technicians, support is needed for this type of decision-making. The module that provides this support will be described in chapter 6.

Another category of decisions which should be supported relates to the way samples are stocked and released and how technicians are assigned to workstations. The DSS should contain a module for testing rules for these decisions. Because the module in our DSS conducts simulation experiments, it will be called the *Experimenter*.

The utilization of capacity depends on the ability to apply rules. Planning freedom and uncertainty about planning freedom are important characteristics of demand. When planning freedom increases, the possibility to smooth the workload (by workload control) improves. Increased uncertainty in planning freedom leads to an increase in slack capacity and, as a consequence, to a lower utilization degree of capacity.

When decisions have to be made on the configuration of the laboratory and the rules to be used, information about the expected demand is needed. Since the actual workload may differ from the forecasted workload, the decision maker should be informed about the performance of the laboratory in alternative scenarios. For a given department therefore, several scenarios will have to be analyzed. The module that realizes this function is called the *Scenario-analyzer*. It calculates the effect of a workload mix that is different from the expected mix.

4

MODELLING THE OBJECT SYSTEM

4.1 Introduction

This chapter offers a presentation of the informational aspects of the Tilburg laboratory and the structure of our decision support system. It will serve as a description of the clinical laboratory and it will indicate the relevant data. Furthermore we will indicate how these data relate to the functions of the DSS.

An introduction to our way of describing information aspects will be given in section 4.2. We particularly focus upon the transaction types in the laboratory (section 4.3) and upon the dataflow (section 4.4). The relation between the organization of the laboratory and the operational rules as stored in databanks will be discussed in section 4.5. Our DSS is directed to feed these banks. Finally the structure of the decision support system is analyzed in section 4.6. It deals with the functions of the various modules and the data flows between the modules of the DSS.

4.2 Description of the laboratory production flow

In this study the production system of a laboratory is selected as the object system. An important characteristic of a production system is its dynamics: the object system is constantly changing as time evolves. In a production situation only certain states can occur or are allowed to occur. The set of states that can and are allowed to occur is called the state space of the system (see our discussion in chapter 2). The definition of this state space serves as the Information Model of the system (Dietz, 1992). Bratko (1990, p. 258) defines a state space as problem situations and possible moves (or actions) forming a directed graph. Moves or actions transform problem situations into other situations.

In describing the laboratory production flow we abstract as much as possible from the organization in the laboratory. An organizational aspect is for example the fact that a shop manager decides to which workstation a certain technician will be assigned. These organizational aspects can be considered as rules about which the DSS should give advice.

An information model of an object system should not be identified with the representation of the object system in a DSS. The representation of the object system in a DSS often reflects implementation issues with respect to the required functions which the DSS has to fulfil. Consideration of implementation issues in the design of the information model has a serious drawback. It may damage the value of the information model when only a small change in the functionalities of the DSS is required. We take implementation issues into consideration in chapter 5 and 6 where the programs themselves are discussed. The specification of the object system in an information model should be independent of a certain implementation in a program.

A method that fulfils our requirements is DEMO, *Dynamic Essential Modelling* (Dietz, 1992). The central idea behind this method is that a distinction should be

made between essential actions on the one hand and informational and documental actions on the other hand. In DEMO an action is *essential* if it can only be performed by a human being, a *subject*. An action is *informational* if it considers the deduction or derivation of information. It is *documental* if it concerns the storage, the transmission and the reproduction of information (Dietz, 1993). In DEMO the involvement of a subject is denoted by an *actor*. An actor is defined by the set of actions and the set of communications it is able to perform. However, we abstracted from the question which subjects are involved in the action. Therefore an actor resembles the concept of *function*. Thus an *object system* is modelled as a system of acting and communicating actors. Thus actors act and communicate with respect to a particular *object world*.

In the following we discuss the transaction types of which the laboratory production process consists, the information that is created by each transaction type and the information needed by each actor for carrying out transactions (figure 4.1). A *transaction* is a sequence of three steps: carrying on an actagenic conversation, executing the action, and carrying on a factagenic conversation. In the *actagenic* conversation, initiated by actor A, the plan or the agreement for the execution of the action by B is achieved. This conversation ends successfully if B commits himself to execute the action. Therefore the result is called an *agendum* for B. In the *factagenic* conversation, the results are stated by B, and accepted by A. After successful completion of this conversation, the fact of the transaction being performed is created, along with a number of information items. The information items created are called the *facta*, and the information items needed while carrying through the transaction are called the *data*. The data for an actor may be facta of itself or of other actors in the system, or they may be externally provided. We consider the following data as external to the production process: the due-times belonging to test requests and rules according to which staff is assigned to processes.

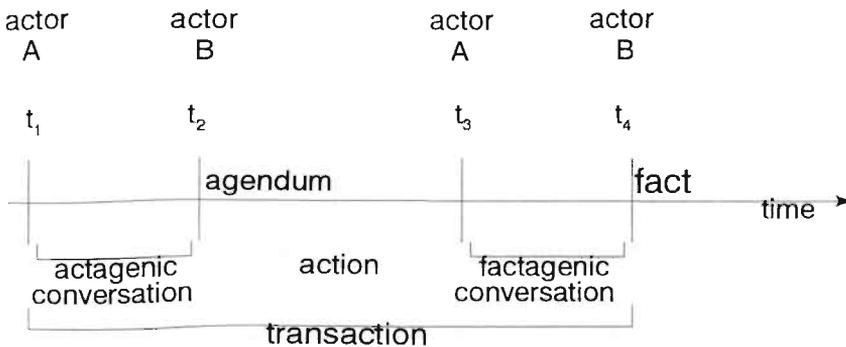


Figure 4.1 Transaction pattern (Dietz, 1993).

4.3 The transactions in the laboratory

A model of the actors in a system and their communicative relationships, is called a *communication model*. Such a model can according to Dietz (1993) conveniently be represented in a *communication diagram*, which is a network of actors, banks and channels. A *bank* is a conceptual store for facts concerning the object world. A *channel* is a conceptual store for the agenda, i.e. agreed future actions. Two kinds of influencing between actors are distinguished, called interaction and interstriction. There is *interaction* between two actors if they are both involved in carrying out the same kind of transactions. One of them is the *initiator* and the other one is the *executor*. There is *interstriction* between two actors if one of them uses information items created by the other as data.

The interaction between the laboratory and the environment (clients) is represented in the following communication diagram (see figure 4.2).

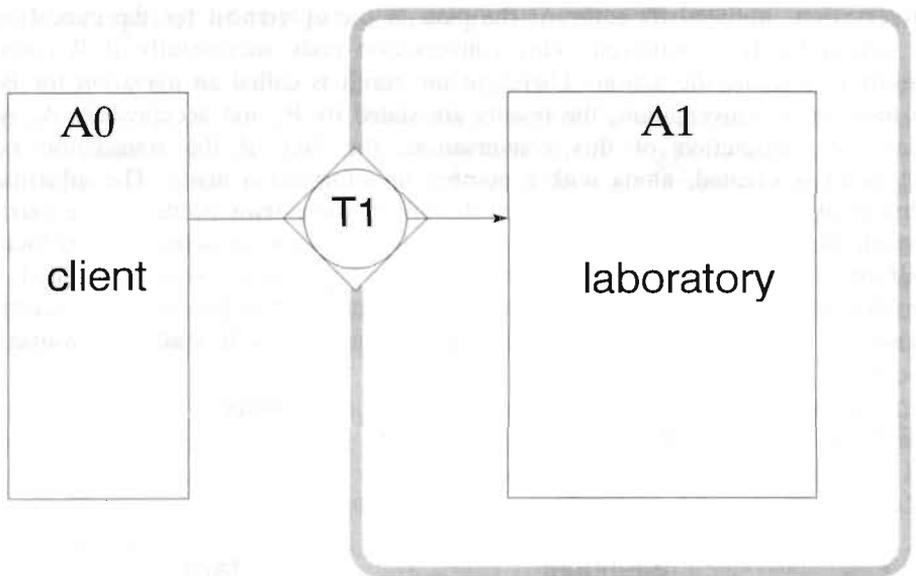


Figure 4.2 The kernel and the environment of the laboratory system.

In communication diagrams the rectangular boxes represent *actors*, the diamond boxes represent *banks*, circles represent *channels*. Plain lines represent *generate links*, which connect the actor who is the executor of a transaction type to the transaction channel. Plain lines with a arrow head represent *execute links*, which connect the executor of a transaction type with the transaction channel. The grey 'roundangle' represents the system boundary.

All transaction types indicated in the communication diagram can be listed in a table (*transaction table*):

<i>transaction type</i>	<i>initiator</i>	<i>executor</i>
T1 testing	A0 client	A1 laboratory

The communication diagram indicates that the laboratory is working for clients. These clients belong to the environment of the laboratory. Clients communicate requests for tests to the laboratory. The laboratory commits to the requests and thus commits to the agenda proposed by the clients.

In figure 4.3 the communication diagram of the entire laboratory is given. The *reception* commits to the requests on behalf of the laboratory as a response to requests for tests by clients, and accepts the required service levels proposed by clients. The clients indicate their service level (for example 'rush'). The exact maximum throughput time belonging to that service level requested by a particular client is given in an external information bank. In the laboratory studied for each request the client can indicate if it is a rush request or not. For the actual required maximum throughput time a distinction is made in 5 categories (15 minutes, 90 minutes, 4 hours, a day and 10 days). Requests belonging to the first two categories are considered to be rush requests. Rush requests are immediately transferred to actor A1.2 *process_samples* by transaction type T2 *rush_order_processing*. The order is then immediately conditioned to be processed and it is given high priority in further processing. If the request is normal (not rush), the processing of the sample is controlled by transaction type T4, *process_controlling*. Then it is actor A1.4 *process_control* who orders by T5, *sample_processing* to process samples (A1.2). A similar situation can be found for the primary authorization. The primary authorization is a check on the results of a particular test by a senior technician. Rush orders are immediately offered to A1.3 *primary_authorization* after performing a particular test. This is reflected through T3 *rush_order_authorized*. For normal orders the proceeding of the primary authorization is controlled by T4 *process_controlling* by actor A1.4 *process_control*. The order for primary authorization is transferred through T6 *normal_order_authorizing*. All results for test requests are checked by actor A1.5 *quality_controller* through transaction type T7 *quality_control*. For the operational work in the laboratory staff is needed. Actor A1.6 *staff_control* conducts through T8 *staff_controlling* this task.

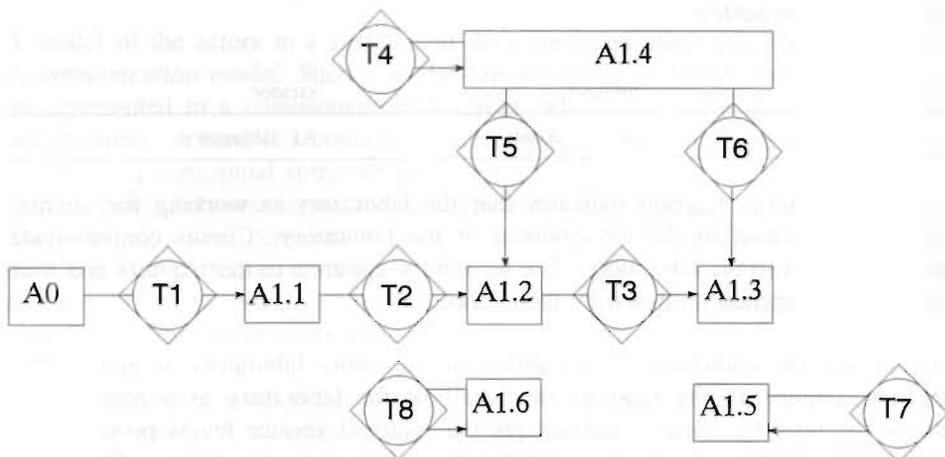


Figure 4.3 Communication diagram with all internal actors.

<i>transaction type</i>	<i>initiator</i>	<i>executor</i>
T1 testing	A0 client	A1.1 reception
T2 rush-order-processing	A1.1 reception	A1.2 process_samples
T3 rush-order-authorizing	A1.2 process_samples	A1.3 primary_authorization
T4 process_controlling	A1.4 process_control	A1.4 process_control
T5 sample_processing	A1.4 process_control	A1.2 process_samples
T6 normal_order_authorizing	A1.4 process_control	A1.3 primary_authorization
T7 quality_controlling	A1.5 quality_control	A1.4 quality_control
T8 staff_controlling	A1.6 staff_control	A1.6 staff_control

4.4 The complete communication model

In figure 4.4 the complete structure of the laboratory process is given. It comprises both the interaction and interstriction between actors. Internal banks containing data needed by actors are T1 *testing*, T2 *rush_order_processing* and T5 *sample_processing*. T1 contains the due-times of orders. These due-times are needed by actors A1.4 *process_control* and A1.6 *staff_control*. Next to these data, actor A1.6 also needs access to the data in T2 and T5 in order to know the actual workload. T2 *rush_order_processing* contains the data needed for actor A1.5

quality_control to do a quality check on the results. For normal orders this information is taken from bank T5.

There is also information needed from external banks ($E_1..E_6$). E_1 contains the rules according to which staff is assigned to processes. As advising on the optimal rules belongs to the core of our DSS, this bank is more extensively discussed in the next section. E_2 contains the quality control rules. It contains quality criteria, description of procedures and how often quality control is performed. E_3 contains the primary authorization rules. They indicate how the results of tests should be checked and how often. E_4 contains the process control data. They also indicate which workstations can be used and whether it is allowed to interrupt the processing of a certain sample. They stipulate when maintenance and calibration should take place. In case of batch processing E_4 prescribes batch sizes. E_5 contains the rules according to which samples should be divided into rush and normal requests. In our laboratory all orders with a maximum required throughput time of 90 minutes are considered to be rush. E_6 contains data describing how test results should be obtained (the test procedures).

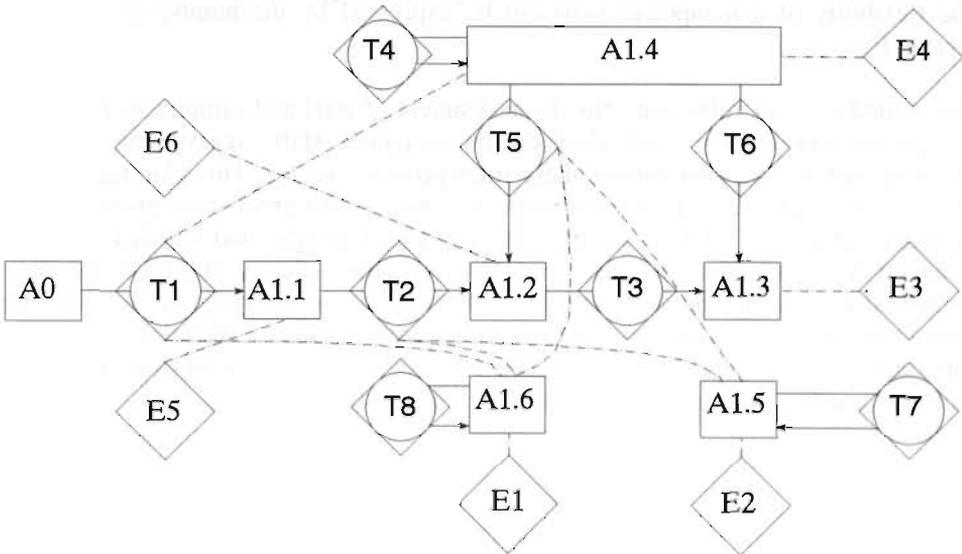


Figure 4.4 Complete communication diagram of the laboratory.

4.5 The structure of staff control rules

In chapter 3 we discussed a planning and control framework. We have seen that a horizontal and a vertical decomposition of the laboratory process is needed. It is evident that the manner of decomposing influences the functioning of the laboratory. Especially the decomposition of the laboratory into jobshops is important here. An important characteristic of a job shop is that technicians belonging to it can be assigned to processes performed in that specific job shop, but not to other job shops. An exception to this is when it is allowed that between

certain periods staff may be re-allocated between job shops. In each period staff and samples can be assigned to processes according to a specific combination of rules concerning workload control, job assignment, staff assignment rules and staff re-assignment rules. Specific instances of these rules have been discussed in chapter 3.

The structure of the laboratory has as a consequence that for the rules contained in bank E_1 two classes of rules are distinguished. First there are rules which stipulate which technicians are allocated to job shops. In the following this set of rules are indicated by us as *manpower_plan*. The *manpower_plan* is operationalized as follows. A day is divided into E planning periods e . The length of such a period is for example 2 hours. The basis of the manpower plan is that the number of technicians needed in a certain period is a function of the workload that has to be processed by a particular job shop. A manpower plan can be derived from statistical (historical) data. This manpower plan determines how many technicians should be assigned at the beginning of a planning period to job shops. The manpower plan does not allow reassignment of staff during the planning period. The flexibility of a manpower plan can be expressed by the number of planning periods E .

The second class of rules concerns the assignment of staff and samples to processes (*assignment rules*). For each decision on assigning staff, reassigning staff or assigning samples to workstations these rules provide criteria. This plan has also to take into account that it is often possible to split up the production process for a particular sample or batch of samples. It should then be specified whether staff has to attend the production, as sometimes during certain phases of processing only equipment is needed. The division into production phases also indicates whether the production processes can be interrupted by withdrawing a technician. This is important, because sometimes technicians have to be withdrawn from a process in favour of another process (in the case of arriving rush samples).

4.6 The structure of the DSS

The structure of the decision support system can be described by a Data Flow Diagram. This diagram is given in figure 4.5.

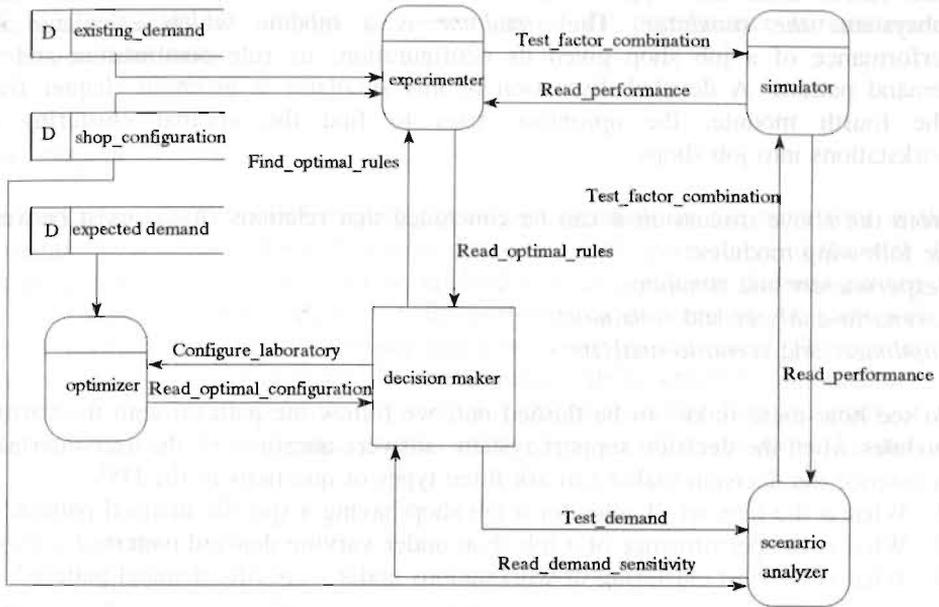


Figure 4.5 Data Flow Diagram of the decision support system.

The decision support system consists of four subsystems or modules:

- the *experimenter*;
- the *scenario-analyzer*;
- the *simulator*;
- the *optimizer*.

The decision maker is located in the centre of the diagram to emphasize that the decision maker is actively involved in planning with the aid of the system. To facilitate the interaction between the decision maker and the DSS, our system is equipped with an user interface. The *experimenter* and the *scenario-analyzer* fulfil functions at the shop level. The *experimenter* conducts experiments to test combinations of scheduling rules at the shop. It is also possible (if desired) to use the *experimenter* to perform single runs (only one rule is tested). This is useful to acquaint the decision-maker with a certain problem. The *scenario-analyzer* determines the performance characteristics of a shop configuration and a given set of operational rules under varying workload conditions. The *experimenter* and the *scenario-analyzer* differ especially from each other from a functional point of view. The *experimenter* is directed to test rules regarding performance characteristics. The *scenario-analyzer* assesses rules on a given configuration under various demand patterns (workload characteristics). It can do this for a given rule but also for a set of rules. In the last case the *scenario-analyzer* compares the quality of

rules taking the demand characteristics into account. It is evident that from a more technical point of view the difference between the *experimenter* and the *scenario-analyzer* is marginal. In the next chapter, where the functions of the *experimenter* are discussed one can read instead of *experimenter* also *scenario-analyzer*, but not vice versa. Both the *experimenter* and the *scenario-analyzer* utilize the third subsystem, *the simulator*. The *simulator* is a module which calculates the performance of a job shop given its configuration, its rule combination and its demand pattern. A detailed discussion of this simulator is given in chapter five. The fourth module, the *optimizer*, tries to find the optimal clustering of workstations into job shops.

From the above discussion it can be concluded that relations (*links*) exist between the following modules:

- *experimenter* and *simulator*;
- *scenario-analyzer* and *simulator*;
- *optimizer* and *scenario-analyzer*.

To see how these links can be fleshed out, we follow the path through the various modules when the decision support system answers questions of the user-interface. In essence the decision maker can ask three types of questions to the DSS:

1. What is the best set of rules for a job shop having a specific demand pattern?
2. What is the performance of a job shop under varying demand patterns?
3. What is the best clustering of workstations under a specific demand pattern?

The DSS has procedures available to deal with each of these questions. In essence these procedures regulate the communication between the already described modules of the DSS. The first question is taken care of by a procedure called *Find_optimal_rules*. The second question is dealt with by a procedure called *Test demand*. The last question about the best clustering of workstations is answered by the procedure called *Configure_laboratory*. We will describe each of these procedures.

Find_optimal_rules

This procedure enables the decision maker to get information about the performance of a set of rules implemented in a certain job shop with a certain configuration (workstations and technicians). In this module the demand pattern (demand and service levels of analyses) is given. The main function of the *experimenter* in this procedure is:

- to translate user defined rules in a format suitable for input to the simulation module;
- to control the use of the *simulator*;
- to translate the simulation results into performance measures.

The translation function of the *experimenter* is needed since the rules to be tested should have a format that conforms to the format requested by the *simulator*. In our case (which is described in more detail in the next chapter) rules are

quantified. By the procedure *Test_factor_combination* these quantifications together with the characteristics of the job shop configuration and the demand pattern are communicated to the *simulator*. Control of the *simulator* is needed because of the experimental approach. The *experimenter* selects rules to be used in a simulation on the basis of a certain experimental design. Finally the *experimenter* has to select the effective rules and to calculate the sensitivity of the performance parameters of these successful rules.

Test_demand

The procedure *Test_demand* uses the *simulator* essentially in the same way as the procedure *Find_optimal* does. The main difference is the perspective of the user. In this procedure the set of rules to be applied is given, only the demand pattern is variable. What is tested, therefore, is the performance of a certain job shop with a certain configuration under demand patterns given by the user. The scenario-analyzer uses information from the experimenter about optimal combinations of rules for a certain configuration. Given this information the scenario-analyzer is able to indicate the performance of the configuration for different demand patterns.

Configure_laboratory

This procedure aims at optimizing the structure of the laboratory. The input from the user consists of information on demand and possible variations and uncertainty in demand, the span of control of shop managers and the minimum size of shops defined in terms of workload. The output of this procedure is a clustering of workstations and the number of technicians needed to meet demand. Interaction with other modules is restricted to information about the performance of shops with a certain configuration under various demand conditions. In table 4.1 the data coupling is defined between the various modules.

Name	Source	Target	Data objects
Find_Optimal rules	Decision maker	Experimenter	shop_configuration; workload_mix
Test_demand	Decision maker	Scenario-analyzer	shop_configuration; rules
Configure_laboratory	Decision maker	Optimizer	workload_mix_laboratory; process_lab
read_optimal_rules	Experimenter	Decision_maker	optimal_rules; workload_mix shop_configuration; performance
Read_demand_sensitivity	Scenario-Analyzer	Decision maker	rules; performance_sensitivity
Read_optimal_configuration	Optimizer	decision maker	clustering_of_workstations; performance
Test_factor_combination	Experimenter Scenario_analyzer	Simulator	rules; workload_mix experiment_nr; replication_nr
Read_performance	Simulator	Experimenter Scenario_analyzer	rules; experiment_nr; replication_nr; performance

Table 4.1 Data coupling between DSS interfaces.

4.7 Conclusions

In this chapter we developed an information model of the laboratory. The functioning of the laboratory is described at three levels: the laboratory, the job shop and the workstation. The DSS uses this information model as a reference. The implementation of this information model in a DSS is described in the following two chapters. We also described the structure of the decision support system. Here we distinguished four subsystems: an *experimenter*, a *scenario-analyzer*, a *simulator* and an *optimizer*. This is a functional distinction. In fact only two 'engines' are built: a simulator and optimizer. The functions of the experimenter and the scenario-analyzer can best be qualified as tools which give the simulation engine the functions of a DSS as described in section 2.7. For the optimizer these tools are integrated in the engine.

5

THE SIMULATION MODULE

The work described in this chapter has been developed in co-operation with Jean Derks, Department of Mathematics, University of Limburg.

5.1 Introduction

The simulation module has two functions. It is used to determine the performance of rules for laboratory job shops. As such it is used by both the experimenter and the scenario-analyzer. Before we discuss the structure of the simulation module we discuss the background of the simulation approach (section 5.2). Then we discuss why discrete event simulation has been chosen as a method to determine the performance of laboratory job shops. In the same section 5.3 we also discuss the system design. Section 5.4 describes how the control rules are implemented. In section 5.5 we discuss calibration and validation issues. How the simulation model can be used for optimization purposes at the job shop level is discussed in section 5.6. In this section the functionalities of the scenario-analyzer and the experimenter are demonstrated with aid of examples. Finally in section 5.7 consequences for the DSS are discussed.

5.2 The simulation approach

The planning and control of the activities in a laboratory can in principle be formulated as a mathematical optimization problem. In case the data of the job shops are deterministic, i.e. contain no stochastic elements, solution methods can be found in the field of combinatorial optimization. An extensive list of literature on sequencing and scheduling algorithms is available. However, for many practical problems, the number of feasible solutions is so large that it is impossible to find the best solution. For such cases many heuristics have been developed giving a solution approximating the best solution. Some of these heuristics make use of criteria as described in chapter 3.

In two cases heuristics are less suitable to apply to the job shop problem. The first case concerns the situation where the dimension of the laboratory activities is so large that even the application of heuristics does not guarantee a solution that is close enough to the best solution. The second case refers to the situation where the data, partially or totally, are stochastic in nature. For stochastic problems only very simple job shops can be solved analytically. The above considerations have led us to conclude that the best mathematical support for the development of a structural tool for analysing arbitrary laboratories is simulation. With simulation both complex deterministic as well as stochastic job shops can be studied.

Mathematical simulation refers to simulating a practical collection of processes with the help of symbolic (mathematical) tools. Discrete event simulation "concerns the modelling of a system as it evolves over time by a representation in which the state variables change instantaneously at separate points in time" (Law & Kelton, 1991). A casual simulation starts with the subdivision of the problem into components whereby a component represents an actor as defined in chapter 4. Examples are technicians, workstations, clients, and so on. Each actor is involved in one or more processes. Here each process describes a certain basic activity of a job shop, for example processing samples.

When an actor is working on a process the following consequences for the state of the laboratory are implied:

- a certain sample is processed. This fact has to be registered.
- the actor is busy for a certain period of time. In case the completion time is only stochastically given, an artificial chance experiment is carried out in order to infer an actual completion time from a given distribution function.
- when a sample has been processed, additional test activities may be necessary. This is for instance the case when a test result implies further testing. This has as a consequence for the administration of the simulation that the waiting queue for the workstation has to be extended.
- the performance of an activity of an actor usually asks for additional resources like labour capacity. In the simulation an activity can only be performed if all the required conditions are fulfilled. Limits to the stock of available resources have to be built into the simulation module.

5.3 Simulation in job shops

5.3.1 Introduction

For programming the dynamic behaviour of the shop system several mathematical modelling methods which are discussed in the literature were considered. For specific problems (especially at the workstation level) an analytical approach (queuing theory) has been applied (Müster, 1978; Janse & Kateman, 1981). The applicability of queuing theory is however very restricted: it is very difficult to use priority rules to control queues and the stochastic behaviour of the system is restricted to exponential distributions. It has, however been shown (VandeGinste, 1980) that if it is possible to model the laboratory as a network of queues (Jackson Network) of samples, queuing theory can be applied (figure 5.1). However, if at least one of the following conditions is not satisfied, it becomes necessary to use simulation techniques to estimate the system's performance:

- no correlation exists between the number of samples in the various queues;
- the sample stream towards a queue is independent of the number of samples in the queue;
- the input and the output sample-flows are mutually independent and not correlated (VandeGinste, 1980, pp. 143-144).

In this study the third condition is not satisfied because the queues of samples of workstations may influence the input flow of samples to these workstations. This is especially the case if workload control is applied. Furthermore, workstations will sometimes have various input flows of samples that are negatively correlated (violation of the first condition). The stochastic process describing the system should be stationary to apply queuing theory.

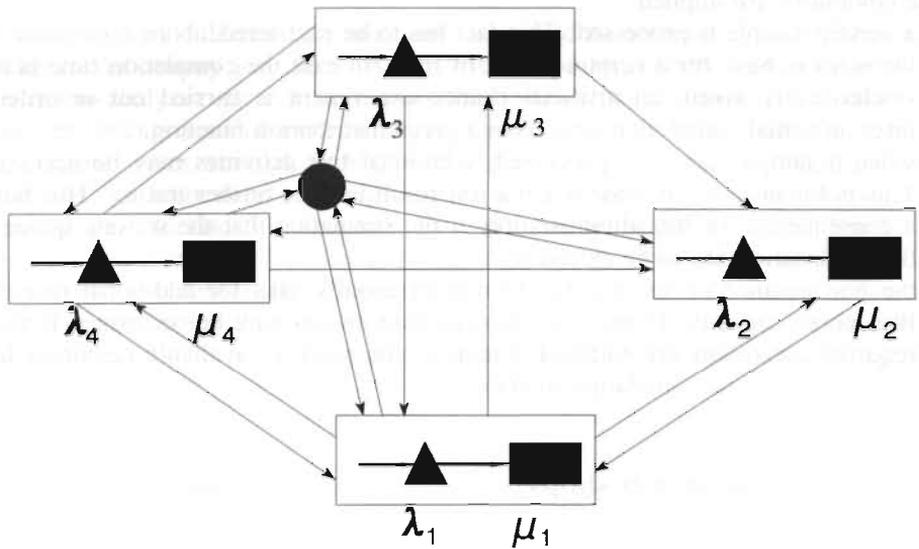


Figure 5.1 The Laboratory as a network of queues (VandeGinste, 1980). λ_i denotes sample flows towards workstations, μ_i denotes processing capacity of workstations for samples. The central dot in the picture indicates the total in- and outflow of the laboratory (or job shops). ▲ indicates the queue belonging to a workstation (symbol ■).

Overall laboratory modelling by simulation has been attempted by Connelly (1989). His work concerned the modelling of some aspects of a clinical haematology laboratory using discrete event simulation. The purpose of this simulation was to study the effects of a growing workload on service times and staff time and to study the effects of a highly automated analyzer on the laboratory's service and staff cost. The approach of Connelly is limited in three respects. First, it was only applicable to the haematology laboratory. Second, the planning and control system is not flexible with regard to changes in the configuration and the workload. Third, priority handling is not explicitly dealt with. Simulation on the workstation level has been applied by Winkel (1984). Although Winkel studies some organizational issues (integration of functions of two or more laboratories), he restricts his study to a model at the workstation level.

In order to be able to simulate different sections of the clinical laboratory, we wanted to design a simulation program that can be applied to different laboratory sections both with respect to configuration and priority handling. Moreover we required that the simulation module should find a good set of planning and control rules if a change in configuration or workload mix takes place.

Our simulation program uses the simulation programming language Simscript which is successfully used over the last 20 years. Simscript supports the organisation of the simulation. It is necessary to specify the collection of actors and

the collection of activities together with their characteristics. These characteristics may be quite extensive, comprising arrival times at the job shop, stochastic processing time, required additional resources, preferred due times, etcetera. Simscript produces an event list, taking into account all the requirements. This event list consists of the moments on a timescale at which the state of the system changes e.g. the set up of a process, the completion of a process, the end of a working shift, etcetera. We rewrote some procedures for event handling to make our simulation program general enough in order to meet our requirements. Internally the simulation proceeds along a time axis and each time an event is encountered the consequences of that event are elaborated, resulting for instance in future events.

In this way we are able to simulate what is actually going on in a job shop. However one important restriction of a simulation approach should be mentioned: during one simulation run the rule is fixed. All the conceivable planning rules are to be tried one by one to this situation. Next the quality of the planning rules can be inspected by evaluating the performance of each rule using specified performance criteria. In specifying and trying out the planning rules, a kind of learning occurs. Normally it is clear in which direction the planning rule should be adapted in order to find better values of the performance parameters. After a number of steps a good, perhaps nearly best, planning rule can be found. An important characteristic of our simulation module is the way in which the states of the shop are represented. The shop is seen as a set of workstations in a certain phase. These phases are represented in figure 5.2. The workstation can be in one

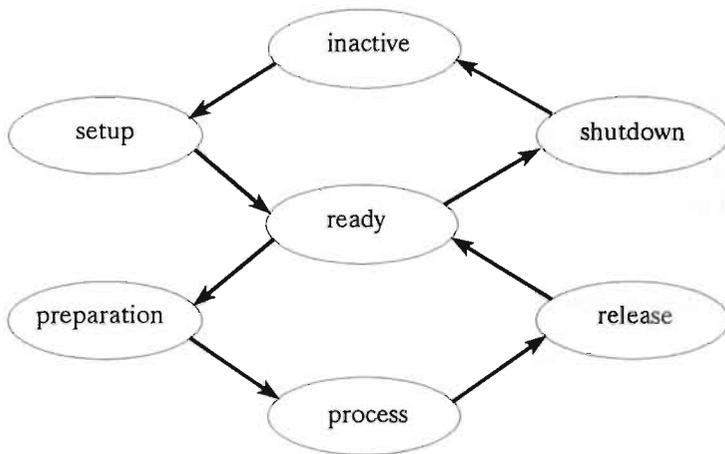


Figure 5.2 Phases of the workstation.

- of 7 phases:
- the ready phase;
 - the shutdown phase;
 - the inactive phase;

- the setup phase;
- the preparation phase;
- the process phase;
- the release phase.

In the *ready phase* the workstation is not operating but it is available for processing when needed. The calibration actions are dealt with in this phase. In the *shut down phase* the workstation is prepared to become inactive. In the *inactive phase* the workstation is not available. It can only become active again after it has passed the *setup phase*. In the *preparation phase* the samples are being prepared for the workstation. In the *process phase* the workstation is actually processing the samples. In the *release phase* the samples have been processed and the test results are being prepared for communication to the requesters. The state of the workstation is a specific combination of (a) the phase of the workstation, (b) the presence or absence of a technician, (c) the activity status of the phase which is either waiting, working, being interrupted or suspended, and (d) the queue of samples belonging to that workstation. In fact the workstation process consists of two cycles. The first cycle includes the opening and closing of the workstation and the second one the production process itself. The first cycle is mainly determined by the characteristics of the workstation, whereas the second one is mainly determined by the sample flow.

In the workstation technicians are not represented separately. They are attached to the workstation. This means that if a technician is assigned to a workstation, this station is in an active state (that is neither in the inactive phase nor suspended or interrupted). It is however possible that the workstation is active but that no technicians are present because some active workstations do not need constant attendance. It is thus possible that a technician can handle more than one workstation at a time.

The queue is mainly determined by the workload mix (which is exogenous), the planning and control rules (which are exogenous for the shop), the presence of an operator and the activity status of the workstation.

5.3.2 System design

In the simulation experiments conducted by the DSS, four groups of variables are considered (figure 5.3).

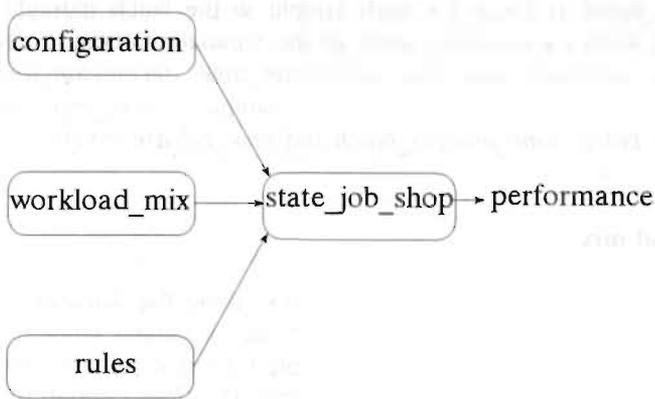


Figure 5.3 Variables in simulation experiments.

5.3.2.1 The dependent variables

The dependent variables comprise the *state_job_shop*. In the simulation program after each request for allocating resources the *state_job_shop* is evaluated. This generates performance measures. All variables in data object *state_job_shop* can be used for making decisions.

5.3.2.2 Independent variables

The independent variables comprise the *configuration* of the job shop, the *workload_mix* and the *rules* for the functioning of the job shop.

The configuration

For each workstation of the job shop a number of variables are communicated to the simulation module (the complete lay-out of the input file is given in appendix 2). Important variables here are:

- workstation_name*: name of the workstation;
- time_set_up*: time to set up the workstation;
- time_prepare_batch*: time for preparing the batch;
- time_process_batch*: time to process the batch;
- time_release_batch*: time to release the batch;
- time_shut_down*: time to shut down the workstation;
- proportionality*: processing time dependency on actual batch size;
- batch_size*: maximum number of samples in the processing batch.

All times are expressed in minutes. The variable *proportionality* indicates whether *time_process_batch* is given for each sample in the batch individually or for the entire batch. Relevant variables used in the simulation module derived from the configuration are *cycle_time* and *technician_time*. *technician_time* is the time needed for a technician to process a sample. *cycle_time* is the sum of *time_prepare_batch*, *time_process_batch* and *time_release_batch*.

The workload mix

The workload mix is a set of parameters describing the demand for services of a certain department. This set consists of the set of sample types together with the actual volume of each sample type. A sample type is defined by the set of tests to be performed on a sample, which characterizes the flow through the shop, and the arrival pattern through the day. The pattern is determined by the parameters $\lambda_{i,t}$ indicating the number of samples of sample type *i* with priority class *s* arriving in time period from *t* to *t+1* (1 hour), *t*=0,1,..23.

The rules to be tested

In chapter three a planning and control model of the laboratory was discussed. With this planning and control model it is possible to make decisions relating to the functioning of the laboratory job shop. For these decisions it is necessary to have information available on the state of the laboratory job shop (*state_job_shop*).

The rules according to which decisions are taken can be described by the way they use the variables in *state_job_shop*. The rules that can be formulated with *state_job_shop* can be very specific or very general. A specific rule is for example the rule that "a certain workstation always gets a high priority for assigning samples". This is important in case the constant costs for this workstation are very high and the marginal costs for processing samples are very low. In such a

- V(1,w): number of samples in the shop queue for workstation *w*
- V(2,w): number of samples enqueued at workstation *w*
- V(3,w): number of rush samples to be expected for workstation *w* during the present staff assignment period *e*
- V(4,w): indicates whether there is enough workload at the workstation for processing
- V(5,w): mean waiting time of samples in the shop queue for workstation *w*
- V(6,w): mean waiting time of samples in workstation *w* queue
- V(7,w): smallest due time in workstation *w* queue (expressed as 1/(smallest due time))
- V(8,w): number of samples with due-time in specified priority time range in workstation *w* queue
- V(9,w): longest waiting time of samples in workstation *w* queue
- V(10,w): indicates whether the next phase of the workstation needs the presence of a technician

Table 5.1 Variables in *state_job_shop* in relation with the rules to be tested.

situation a high weight is assigned to that workstation w . For that purpose $V(4,w)$ could be used. General rules as described in chapter 3 can also be described using $state_job_shop$. In table 5.3 we indicate the rules and the variable(s) of $state_job_shop$ they use.

Rule	Variable of $state_job_shop$
FIFO (\approx LWT)	$V(9, \cdot)$ or $(V(5, \cdot)$ and/or $V(6, \cdot))$
SPT	$V(4, \cdot)$
LSO	$V(4, \cdot)$
LNQ	$V(1, \cdot), V(2, \cdot), V(3, \cdot)$
HPJ	$V(7, \cdot)$ or $V(8, \cdot)$
LPT	$V(1, \cdot), V(2, \cdot), V(3, \cdot)$

Table 5.2 General rules and their relation to $V(i,w)$.

Sometimes there is some freedom how to apply a certain general rule. For example, in case of the LPT-rule and the LNQ-rule one can choose to take into consideration not only the workstation queues, but also the shop queues and even the expected number of samples for the assignment period. With respect to the FIFO-rule and the LWT-rule one can choose to take into account only $V(9, \cdot)$ but also $V(5, \cdot)$ and/or $V(6, \cdot)$. For the HPJ-rule the option exists to take into account the degree of priority $V(7, \cdot)$ or the number of samples within a certain priority range, variable $V(8, \cdot)$. It is also possible to look at both of them. The SPT-rule and the LSO-rule can be implemented by $V(4, \cdot)$. $V(4,w)$ gets the value 1 if there is a certain minimum of samples in the batch, or if there is a rush sample in the workstation queue or there is a sample in the queue with a due-time in the next hour.

In order to describe the rules (which $V(i,w)$ should be used?) we introduce quantifiers to the variables. This is implemented by assigning a weight θ_i to each $V(i, \cdot)$. If $V(i, \cdot)$ is not taken into account, then θ_i is equal to 0, otherwise it is 1. To weigh workstations w in each rule we introduce the parameter b_{iw} which is the multiplication factor for $V(i,w)$ when θ_i is one. The weight for each workstation w is now calculated as follows:

$$\forall w \in W(shop): weight_w = \sum_{i=1}^{10} \theta_i \cdot b_{iw} V(i,w) \tag{5.1}$$

So parameter b_{iw} has not only the function to weigh different workstations but also to weigh different $V(i, \cdot)$ in rules which take more than one $V(i, \cdot)$. Note that b_{iw} is rule-dependent. In table 5.3 we present the b_{iw} for general rules we use.

value of b_{iw}	rules
$b_{iw} = 1$	FIFO (LWT), LNQ
$b_{iw} = \text{cycle_time}_w$	SPT, LPT, HPJ

Table 5.3 Values of b_{iw} for general rules.

The reason why Cycle_time_w is used to weigh workstations is that when the SPT-rule is used workstations with the shortest cycle time have priority. When HPJ and LPT are used the process time is necessary to determine the planning freedom ((due-time - actual moment)/cycle_time).

If different $V(i, \cdot)$ are used in one rule also a common scale unit is needed in order to weigh them. This is for example the case in an application of the HPJ-rule where both $V(7, w)$ and $V(8, w)$ are taken into account. How much weight should be added to $V(7, w)$ and how much to $V(8, w)$? If one $V(i, \cdot)$ should have another impact on the total weight of the workstation than another balancing factor c_i could be introduced. The weight function (5.1) becomes then:

$$\forall w \in W(\text{shop}): \text{weight}_w = \sum_{i=1}^{10} \theta_i \cdot c_i \cdot b_{iw} V(i, w) \quad (5.2)$$

The value of c_i is determined by the planner and is subjective. As this value can be put in b_{iw} we will not explicitly use it. In the following analysis we will stick to weight function 5.1 and assume that the balancing weights are equal to one.

Suppose that in (5.1) we take $b_{iw} = 1, \forall i, w$, then we could experiment with rules through simulation experiments only by manipulating θ_i . As depending on the circumstances the functioning of shops can be described by two b_{iw} values (1, cycle_time_w) in case of $V(4, \cdot)$ we introduce a new element $V(11, \cdot)$ equal to $V(4, \cdot)$ that can be assigned the second b_{iw} value ($b_{11w} = 1$). A complete list of symbols is given in appendix A.

5.3.2.2 Performance criteria

For each simulation run the workload_mix, the configuration, the control rules and the number of technicians are given. To obtain performance characteristics of the shops, the simulation module produces for each simulated day output comprising the actual number of incoming samples, the number of samples not processed in time (samples late), the actual throughput time for each service class, the utilization degree of each workstation and the utilization degree of technicians. Each instance is simulated for 100 days. The performance characteristics are aggregated over this period by calculating the mean and the standard deviation per day for each performance measure.

The performance criteria are given below.

<code>mn_out</code>	the mean number of processed and authorised samples per day during the simulation;
<code>sd_out</code>	the standard deviation around <code>mn_out</code> ;
<code>mn_left</code>	number of samples not processed at the end of the day;
<code>sd_left</code>	standard deviation around <code>mn_left</code> ;
<code>mn_late</code>	number of samples not authorised before due-time;
<code>sd_late</code>	standard deviation around <code>mn_late</code> ;
<code>mn_transfer</code>	number of times that technicians are transferred between workstations;
<code>sd_transfer</code>	standard deviation around <code>mn_transfer</code> ;
<code>mn_working</code>	time that technicians have worked at the workstation (percentage of total available time);
<code>sd_working</code>	standard deviation around <code>mn_working</code> ;
<code>util_w</code>	utilization degree of workstation w ;
<code>operator_w</code>	utilization degree of technicians at workstation w ;
<code>throughput_time_p</code>	mean throughput time for each priority class p .

5.4 Control rules

In this section we describe how our quantification method will be applied to workload control and assignment rules. Both workload control and assignment rules are described in chapter 3. Workload control is an aggregate technique controlling the release of samples to workstation queues. It is thus a manner of controlling the arrival of samples at workstations queues. Assignment rules control the assignment of staff to workstations. As sample types are processed by specific workstations staff assignment is important for the throughput of samples. All queues are ordered on the due-time.

First we discuss workload control, then we deal with staff assignment.

5.4.1 Workload control

The workload released is dependent on: the available capacity (*capacity*), the expected number of samples arriving with a due-time within two hours (*expected_rush_samples*) and the workload released in previous periods and not yet processed (*previous_released_workload*, $\Sigma_w V_{3w}$).

The variable *expected_rush_samples* (equal to $\Sigma_w V_{3w}$) is calculated from the mean fraction of rush requests (*rush_demand_fraction_w*) of the expected demand for workstation w . A rush request is a request arriving in this assignment period with a due-time within the actual assignment period.

First the value of O_{jc} is read from the input file (the value is given by the user or the optimization module). O_{jc} is the number of technicians at a certain job shop j in technician assignment period e . The time needed for a technician t_w to process a sample should also be provided by the user (and can thus be found on the input

file). If one does not care about an overestimation of the value of t_w , one could consider calculating t_w by:

$$t_w = \frac{\text{cycle_time}_w}{\text{batch_size}_w} \quad (5.3)$$

The workload W to be released can be calculated as follows:

$$W = O_{je} \cdot W_0 \quad (5.4)$$

where W_0 equals the total time a technician is available for working.

Next we determine the workload for rush orders:

$$W_u = \sum_w t_w \cdot V(3,w) \quad (5.5)$$

As W_u is released immediately after the arrival of the concerned samples at the shop, for non-rush samples only $W - W_u$ is available for release.

The next step is that we consider the workload due to samples from the last technician assignment period:

$$W_q = \sum_w t_w \cdot V(2,w) \quad (5.6)$$

In case $W - W_u - W_q$ is positive, samples are released from the shop stock until this amount is non-positive. If needed, this release is repeated during the technician assignment period. Each time a sample is released, the total technician time needed to process that sample is subtracted from $W - W_u - W_q$. This amount is only dependent on the sample type and can be calculated on the basis of workload data, which is done at the start of the simulation.

Note that the shop queue size (sum of $V(1,w)$) is unimportant for determining the actual amount of workload to be released in the next period, except when the workload in the shop queue is less than shop's capacity (*capacity*).

Example on workload control

A department has three technicians and four workstations available. At each workstation only one type of analysis can be performed. Different analyses have to be performed on different workstations. The cycle-times (cycle_time_w) of each analysis on workstation w , $w=1, \dots, 4$, are:

cycle_time_1 : 10 minutes;

cycle_time_2 : 15 minutes;

cycle_time_3 : 8 minutes;

cycle_time_4 : 3 minutes.

Every technician assignment period (suppose 2 hours) samples are released from the shop_queue $V(1, \cdot)$. Consider table 5.5 describing the situation at the beginning of an assignment period.

$V(i, w)$	$V(i, 1)$	$V(i, 2)$	$V(i, 3)$	$V(i, 4)$
$V(1, w)$	5	10	20	0
$V(2, w)$	1	2	1	1
$V(3, w)$	4	0	0	30

Table 5.4 Example of situation at begin of assignment period.

If we apply the procedure *Workload_Control* to the relevant variables of *state_job_shop*, the instances will be:

Call: *Workload_Control*

input: 360

output: *workload_to_be_released*

Execution:

(1) *previous_released_workload* = $(1*10) + (2*15) + (1*8) + (1*3) = 51$

(2) *expected_rush_samples* = $(4*10) + (0*15) + (0*8) + (30*3) = 130$

(3) *workload_to_be_released* = $360 - 130 - 51 = 179$

(4) if $179 > ((5*10) + (10*15) + (20*8) + (0*3))$ then: False.

Return: *workload_to_be_released*: 179

The effect of this call of *Workload_Control* is that a workload of 179 minutes is released to the workstations. In this call *available_capacity* has a value of 360 minutes. The value is obtained by multiplying the number of technicians by their time available to work for a period of two hours. It should be noticed that, due to human factors, in practice the actual utilization degree of technicians cannot be expected to exceed 85-90 percent.

5.4.2 Staff assignment

In our approach an order for a specific process implies the use of a certain workstation. Therefore no decision rule is formulated for assigning samples to workstations: they just go there. The only remaining decision rule concerns the assignment of staff to workstations (variable: *staff_assignment_rule*). The sample assignment is implemented by releasing samples from the shop_queue in a certain sequence (e.g. FIFO). The staff assignment rule is implemented by evaluating the states of the workstations and their queues. Each assignment decision is made after evaluating the *state_job_shop*.

The following describes the assignment procedure.

Procedure Assign

input: *staff_assignment_rule*; *no. of technicians*

output: *technician, workstation*

(1) evaluate $V(i, \cdot)$, $\forall i$; $\theta_i \neq 0$

(2) $\forall w \in W(\text{Shop}): \text{weight}_w = \sum_{i=1}^n \theta_i \cdot b_{iw} V(i, w)$

(3) \forall technician do

(4) for $\max\{\text{weight}_w \mid w = 1, 2, \dots, W\}$ $\text{weight}_w = -\infty$

(5) loop (2)

In this procedure technicians are assigned to the '- ∞ '-workstations.

Example on staff assignment

The department from the example on workload control faces the problem of how to assign staff to workstations. Here we give an example by comparing the rule that assigns staff to workstations with the longest mean waiting time (LWT-rule) with the HPJ-rule for assigning staff to workstations. We implement the LWT-rule here by using both the mean waiting time of samples in the workstation queue and the mean waiting time in the job shop queue. Thus for the LWT-rule $V(5, w)$ and $V(6, w)$ are relevant. For the HPJ-rule the workstation queue with the sample with the nearest due-time has priority. It is also possible to assign a weight to queues with due-times within a certain specified time range. This time range is defined here as 2 times the cycle_time_w . For the HPJ-rule $V(7, w)$ and $V(8, w)$ are evaluated.

Process	V(5,w)	V(6,w)	V(7,w)	V(8,w)
1	150	50	1/10	3
2	240	80	1/35	0
3	220	70	1/40	0
4	0	0	1/5	1

Table 5.5 Example of *state_job_shop*.

In table 5.5 the *state_job_shop* is given. We first apply the LWT rule by assigning equal weights to V(5,w) and V(6,w), by giving θ_5 and θ_6 the value 1. The others are valued 0.

The procedure *Assign* is executed as follows.

```

Call: Assign
input:  $\theta_5=1$ ;  $\theta_6=1$ ; no of technicians: 2;
output: workstation(s)
Execution:
{comment:  $b_{iw}$  and V(i,w) for  $\theta_i=0$  are not displayed;
only steps 2 en 4 are displayed}
(2.1) w=1
(1.1)  $b_{51}=1$ ;  $b_{61}=1$ ;
(1.2)  $weight_1=(150*1)+(50*1)=200$ 
(2.2) w=2
(1.1)  $b_{52}=1$ ;  $b_{62}=1$ ;
(1.2)  $weight_2=(240*1)+(80*1)=320$ 
(2.3) w=3
(1.1)  $b_{53}=1$ ;  $b_{63}=1$ ;
(1.2)  $weight_3=(220*1)+(70*1)=290$ 
(2.4) w=4
(1.1)  $b_{54}=1$ ;  $b_{64}=1$ ;
(1.2)  $weight_4=(0*1)+(0*1)=0$ 
(4)  $\max\{200, 320, 290, 0\}$ :  $weight_2=-\infty$ ;
(4)  $\max\{200, -\infty, 290, 0\}$ :  $weight_3=-\infty$ ;
Return: 2,3

```

Compare this to the execution of the procedure *Assign* with the HPJ-rule which we implemented by giving θ_7 and θ_8 the value of 1 and by setting the other θ_i to zero.

Call: Assign
input: $\theta_7=1$; $\theta_8=1$
output: workstation(s)
Execution:
{comment: b_{iw} and V_{iw} for $\theta_i=0$ are not displayed only steps 2 en 4 are displayed}

(2.1) $w=1$
 (1.1) $b_{71}=10$; $b_{81}=10$;
 (1.2) $weight_1=(10*1/10)+(10*3)=31$;

(2.2) $w=2$
 (1.1) $b_{72}=15$; $b_{82}=15$;
 (1.2) $weight_2=(15*1/35)+(15*0)=0.43$;

(2.3) $w=3$
 (1.1) $b_{73}=8$; $b_{83}=8$;
 (1.2) $weight_3=(8*1/40)+(8*0)=0.2$;

(2.4) $w=4$
 (1.1) $b_{74}=3$; $b_{84}=3$;
 (1.2) $weight_4=(3*1/5)+(3*1)=1.6$

(4) $\max\{31, 0.43, 0.20, 1.6\}$: $weight_1=-\infty$;
 (4) $\max\{-\infty, 0.43, 0.20, 1.6\}$: $weight_4=-\infty$;

Return: 1,4

In this example (especially in the application of the HPJ-rule) we observe that it is sometimes important to use b_{iw} to balance weights when we use rules in which more than one θ_i is non-zero. In the last example we see that if for one workstation $V(8,w)$ is non-zero, practically all $V(7,w)$ could be neglected. If the planner does not want this, he may correct b_{7w} with a certain multiplication factor (c_i).

5.5 Calibration and validation

Calibration refers to estimation of the parameters of the model.
 Validation refers to the evaluation whether a certain model (in our case the simulation model) reflects the situation it is supposed to represent. Hence the question is how close the model simulates reality. Validation is often a difficult task since there are no similarity measures available, which measure the similarity between the model and reality. Usually this problem is overcome by comparing the results of the simulation model with well-known results. When they are close enough the model is accepted and else the model has to be adapted. In fact validating in this way approaches the process of calibration. We applied our simulation model to a number of departments of the St. Elisabeth Hospital at Tilburg, using data gathered from that hospital. The data were collected in several ways. We first captured data from administrative resources. By this method we were able to determine the volume of samples processed on a yearly basis. Clearly this was not sufficient for our purposes, but these data provided us with a global mean.

Continuous recording of data concerned with the processing of samples was a second method we used. At all job shops time data about samples were registered. This was done in the months September and October 1991. For each sample the exact time was registered at the following events:

1. the arrival of the sample at the reception;
2. the beginning of the processing of the sample at the reception;
3. the departure of the sample from the reception;
4. the arrival of the sample at the job shop;
5. the beginning of the processing of the sample at the workstation;
6. the end of the processing of the sample at the workstation;
7. the time of primary authorization;
8. the time of quality control (second authorization of the results).

These registrations were very informative with respect to the actual number of samples arriving per hour for each process, not however for determining the exact statistical distribution of the interarrival times. This is due to the fact that when registrations are made queues are generated. These queues prevent an accurate estimation of the arrival time. The net effect of this is that the measured distributions of interarrival times become too smooth.

As a third method of gathering data we registered the rush requests coming from certain medical departments. These departments were the Obstetrics Department and the Intensive Care Unit. Determination of the arrival rate of samples by this method is only useful for rush requests with very short required throughput times (between 10 and 30 minutes) as the time between the collection of the sample and the arrival are constant and very short.

A fourth method of gathering data was by interviewing technicians. Interviews were held in two ways. In one series of interviews technicians were asked to estimate the number of samples arriving each hour. In a second series we ran the simulation program. In such a presentation only the flow of arriving samples, the number of rush samples and the actual workload were shown, the intermediate results of the simulation run were not presented. Technicians were individually asked whether the output was reasonable. If that was not the case the input data were corrected and another simulation run was performed. If finally the output was approved the input data were compared to the mean number of samples arriving per hour according to the continuous registrations. This method appeared to be successful for determining mean number of samples arriving/processed per hour, not however for determining the statistical distributions of the arrival patterns.

From the measured distribution of samples arriving each hour the mean values were determined. An exponential distribution was assumed to represent the arrival times. For rush samples this is a very reasonable and common approach as only very few samples arrive at each workstation each hour (the mean number of super rush samples per hour at each workstation was always less than 3). Moreover these samples arrive independently from each other. For other samples (not rush requests) the statistical distribution of their arrival pattern does not have significant influence on the shop performance. The volume of these other types of requests are

often high and there is also planning freedom for these samples. The choice was thus to use an empirical distribution for the means per hour and an exponential distribution to represent the arrival times within each hour.

Calibration and validation, although different, are closely related to each other. For calibration purposes data from registrations and interviews can be used. Validation of the simulation can then be conducted for the existing situation (for which the data has been collected). Two situations can be distinguished where validation is a problem. First when simulations are needed to calibrate the model and second when situations are simulated for a scenario-analysis.

A first problem was encountered when calibrating the discrete chemistry department. It appeared impossible to estimate the exact parameter values using either registrations or interviews. Our strategy was to assume extreme values belonging for the problematic parameters and to use these in simulations. The responsible technicians were then asked which simulations appeared to be correct. Several technicians were independently involved in this process. After a few iterations satisfactory values could be obtained. Since in the calibration phase we did not use the *experimenter* we could be sure that technicians did not choose the parameter values of the model that would result in an output they would like to see. An other example of a situation where it is difficult to estimate the correct parameter values is when labour is more flexible than can be modelled by us. Simulating the existing situation (the nominal scenario) of the departments of batch chemistry and haematology resulted in the conclusion that the percentage of idle time of technicians was sometimes high. It appeared that technicians conducted for example administrative activities or attended other job shops during that time. For calibration purposes one can in such cases try to simulate the situation with less technicians available. But the results then showed a very low output of processed samples. Discussion of results with technicians is very important here and a model can only be considered to be validated after such discussions.

The second situation where it is difficult to validate the model, is in a scenario-analysis. We should state here that in principal it is impossible to validate the results if one does not have the opportunity to perform real experiments. Since experiments usually cannot be performed, two other strategies are important. The first strategy is to rely on face validity: the results are discussed with experienced technicians and with the laboratory management. A second strategy is to test whether the results of a scenario-analysis are very sensitive to some model parameters. If this is the case, one should interpret the results with care. The characteristics of the calibrated departments are given in the appendix 3 and 4.

5.6 Optimization with simulation

5.6.1 Introduction

As described in chapter 4 the simulation module can be called in two ways: by the *experimenter* and by the *scenario_analyzer*. When called by the *experimenter*, the simulation module is called at least several times (as many times as there are experimental factor combinations). When called by the *scenario_analyzer* the difference with the situation in which the simulation module is called by the *experimenter* is that sensitivity figures are provided with regard to changes in the workload or the workload-mix. Both the *experimenter* and the *scenario_analyzer* allow single run simulation. In a single run simulation only one rule is tested. We will discuss each way of simulation with the aid of an example (the haematology department).

To test the performance of a configuration some aspects have to be taken into account. The first aspect is the set of rules (θ_i) that are optimal for that workstation. The second aspect is the workload (*workload_mix*) it can handle. The first aspect is tested by the procedure *Find_optimal_rules*, the second aspect is handled by the procedure *Test_demand*. Here we discuss the procedure *Find_optimal_rules*. We will first discuss the input variables, then the performance and finally the approach taken in the module *Experimenter* (to which *Find_optimal_rules* communicates).

The department's configuration

The variables concerning the department's configuration are used in the procedure *Test_factor_combination*. These variables are constant in the procedure *Find_optimal_rules*. These variables have already been described in section 5.3.2.2.

The workload mix

Important are the *workload_mix* variables. These variables are used as parameters, meaning that their value is fixed when calling *Find_optimal_rules*. The maximum mean sum of λ_{i_s} per day (*mn_in*) for each set of rules is used as a performance measure. The proportion of all processes i having a certain value λ_{i_s} remains the same during experimentation. If the simulation run is in a steady-state *mn_in* will be (almost) equal to *mn_out*. In the following we will use *mn_out* instead of λ_{i_s} as performance parameter. For finding the maximum possible sum of *mn_out* a binary search procedure is used. For each instance for which a maximum *mn_out* exists a further analysis is conducted to determine the influence of the rules on other criteria. This last analysis and the binary search procedure will be discussed together with an example.

An interesting question a decision-maker may pose is how the performance of a department changes when the workload-mix changes. This question is answered by changing *workload mix*, experimenting with the θ_i 's to find a set of good rules for the new situation after which the difference in performance between the old and new demand conditions is calculated. This analysis method is also discussed in the next section.

5.6.2 Single runs

The data required by the simulation module are described in detail in the appendices. Table 5.6 gives the workload characteristics of the available workstations of the haematology department which we will use here as an example.

No	workstation	actual volume	distribution in priority classes				
			1	2	3	4	5
1	Bulk haematological analyzer	400	0	10	60	30	0
2	Small size haematological analyzer	20	0	0	30	70	0
3	Eye blood cell differential	40	100	0	0	0	0

Table 5.6 Available workstations and workload characteristics.

The appendices B, C, D and E describe the input files and specify the technical characteristics.

When using the simulation module for single runs the results should be stable in order to be interpretable. Output based on a single run of a simulation of a single day is often not reliable. Therefore the planner has to indicate how many days have to be simulated in a single run and he can choose between different starting values for the random streams. In the next table the output is given for 6 single runs of ten days using different random streams. The input parameters were obtained from the nominal scenario (the existing situation). In this chapter the combination of this configuration, the nominal workload mix and 3 technicians is indicated as *hem_std.3te*.

variable	run 1	run 2	run 3	run 4	run 5	run 6	avg
mn_in	428.58	428.71	425.10	428.61	424.10	427.29	427.06
sd_in	22.49	17.13	18.38	18.33	20.31	15.73	18.69
mn_out	421.65	424.13	417.90	421.06	416.61	423.71	420.84
sd_out	80.86	70.60	75.24	84.21	53.41	70.37	72.18
mn_left	86.42	85.48	81.61	85.84	82.55	87.29	84.87
sd_left	14.40	13.36	15.08	14.63	10.89	13.59	13.70
transfer	45.94	46.29	46.19	46.87	46.03	47.35	46.45
sd_transfer	5.25	7.30	8.49	6.75	6.72	7.78	7.04
mn_working	54.61	54.00	53.17	55.35	53.83	55.56	54.42
sd_working	5.51	5.84	5.63	5.33	6.28	6.20	5.79
mn_late	0.00	0.00	0.00	0.03	0.00	0.00	0.03
sd_late	0.00	0.00	0.00	0.18	0.00	0.00	0.25
util1	30.82	30.67	31.10	30.81	30.84	30.49	30.79
util2	25.95	25.75	25.82	25.64	25.80	25.85	25.80
util3	30.87	30.40	30.16	30.53	30.35	30.61	30.49
util4	12.38	13.18	12.93	13.00	13.00	13.06	12.93
utop1	36.96	36.82	36.65	36.73	36.65	36.81	36.77
utop2	13.86	13.86	13.79	13.75	13.79	13.84	13.82
utop3	25.62	24.91	24.95	26.20	25.23	26.23	25.52
utop4	12.51	12.96	12.76	12.74	12.71	12.83	12.75
thr1	0.37	0.36	0.38	0.37	0.36	0.51	0.39
thr2	1.22	1.36	1.24	1.32	1.39	1.28	1.30
thr3	1.29	1.33	1.31	1.45	1.28	1.33	1.33
thr4	2.72	2.68	2.64	2.64	2.61	2.78	2.68
thr5	6.33	6.50	6.39	6.53	6.21	6.53	6.41

Table 5.7 Results of single runs for *hem_std.3te*.

As can be seen the results from the different runs vary only slightly. This is due to the fact that the number of samples arriving daily is fluctuating around 425, whereas the machine capacity is around 550 per day. Moreover the workload mix in the existing situation allows enough planning freedom to smooth the workload. An important variable with which to check the validity of the simulation is *mn_left*. Together with *mn_late* it indicates whether the shop is capable to realize the required production. If *mn_left* or *mn_late* increases as a function of the number of

simulated days, the results are not stable. The simulation module monitors the number of samples late for each run, and it is possible to let the simulation module stop if the number of samples late exceeds a certain user defined percentage. The advantage of this procedure is that the number of complete but not usable runs decreases. In setting this percentage we should take into account that this percentage should not be too high (because in that case it does not check on lateness). On the other hand it should not be too low either, because as we are simulating, there is always a chance that samples be late on a certain simulated day. One should however keep in mind that when the percentage of samples allowed to be late increases, also the number of days to be simulated in each run should be increased. The reason is that if a shop is not capable to realize the required production an increase in the number of *samples late* may be expected. The alert that samples are late is given when the due-time is exceeded. Therefore only after some time the critical percentage of samples late is exceeded. This effect is further strengthened as samples with a due-time within a certain specified time interval can have priority. Therefore the number of samples left should be measured against the number of simulated days. Figure 5.4 shows the relation between the number of samples left and the number of simulated days in a stable situation. In figure 5.5 the relation between the number of processed and authorized samples and the number of simulated days is given for the same situation. The parameters for determining the results are the same as for run 1 in the above table: for samples left and the number of jobs late the variance is low and not dependent on the number of simulated days. Both figures show that the results are stable.

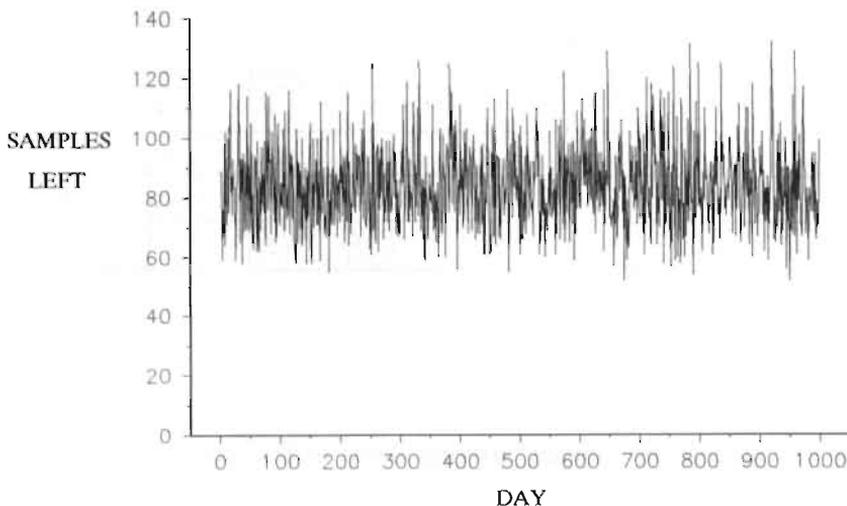


Figure 5.4 Samples left during simulation run.

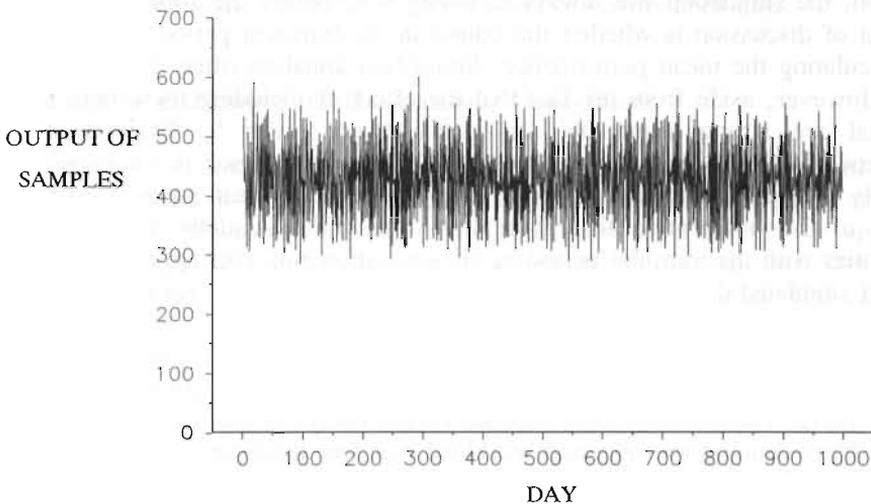


Figure 5.5 Processed and authorized samples during simulation run.

5.6.3 Experimenting with rules

Here the *simulator* is called by the *experimenter* which tries to find for each rule the highest demand for a certain workload mix (service degree) which the department is able to process. The *experimenter* checks the performance of the department by counting the number of samples late. The highest workload for each rule is found by a binary search procedure. For each rule half of the technically feasible (or some other arbitrary high) workload is simulated. After each simulated day the *experimenter* checks whether the number of samples late does not exceed the allowed maximum percentage of samples late. If the simulation for that day is successful, the *simulator* goes on to the next day. If each day (for the total number of days the *experimenter* wants to simulate) is successfully simulated, these data are averaged and stored. Then a new workload is taken which is between the workload of the last successful run and the nearest higher unsuccessful run. If a run does not succeed a workload is taken which is between the last run and the highest successful run. For each rule this process continues until it is not possible to find a higher workload. The user can indicate how accurately this highest workload has to be determined. He can choose a value between 2 and 10% of the starting value (maximum workload indicated by the user) for the binary search procedure. This value determines the number of runs for each instance. If this value is low (e.g. 2%) then an accurate estimation of the workload is obtained, but then 6 runs are necessary, whereas if this value is 10% only 4 runs are required.

The *experimenter* and the *scenario_analyzer* both perform only single runs for each instance. To assure that stable results are obtained each run should last long enough. In those situations where the run was successful according to the above criterion, the simulation was always in steady state before the 10th simulated day. A point of discussion is whether the output in the transient period should be used for calculating the mean performance. Simulation analysts often do not use these data. However, aside from the fact that the effect of including these data is often marginal, strong arguments for inclusion are given by Kleijnen & Van Groenendaal (1992). First the exact end of the transient period is not known. And secondly including the transient period often reduces the mean squared error. This is due to the observation that in reality sometimes situations occur that have similarities with the transient period in the simulation run. Our approach is that we used all simulated days with the exception of the first one (day 'zero').

Once the simulation has reached a steady state we can be confident that the calculated performance measure *mn_out* ('samples processed each day') has the correct value. The accuracy only depends on the choice of the user of the accuracy level for determining the maximal workload (see our discussion above). For other performance measures this is not true and for them we have to estimate confidence intervals. As laboratory job shops have to be characterized by a non-terminating model, the confidence interval should be influenced by the number of days the simulation is in a steady-state situation. Non-terminating models apply when we study a job shop where the situation of the previous day influences the performance of the present day. Queues of samples are typical in this respect. A queue length on day t influences the queue length on day $t+1$.

In determining the confidence interval we split the total simulation run into a number of subruns (or batches) where each subrun is of size k days. If k is large enough the observed values of variables (for example transfer rate) of each subrun will be normal random variables with the same mean and variance (Law & Kelton, 1991). This means that we can use the t statistic to determine the number of subruns given a certain confidence interval $(1-\alpha)$. The problem is of course the choice of subrun size k . Subrun size k should be high enough so that the means of the subruns are uncorrelated with each other. As queues will often correlate between two subsequent days, the number of samples left at the end of each day (left) was used to check the correlation of the subrun means. For example for the *hem_std.3te* configuration only subruns below 6 days showed significant correlations to estimate the exact parameter values ($\alpha=0.05$) between the samples left. All subrun sizes with k greater or equal to 6 are allowed. As the first subrun contains the transient period we can generate between 1 and 15 subruns from a simulation run of 100 days. Suppose we want to test (with confidence $p=0.95$) that the mean of the transfer rate, 46, determined on the basis of a simulation run in the *hem_std.3te* scenario, is not significantly different from 48 transfers. If we take $k=10$ we have 9 subruns available to determine the confidence interval. With a standard deviation calculated on the 9 means of transfer rate, 2.2, we obtain a confidence interval of 46 ± 1.69 by applying the t statistic. This implies that the 46 transfers is significantly different ($\alpha=0.05$) from 48 transfers.

In all our examples the simulation length was set to 100 days. As already discussed the significance of output is highly dependent on the necessary size of k . The maximum necessary value of k we found was 15. It is of course possible for the user to extend the simulation run to a higher number of simulated days.

Experimenting with rules: the Haematology Department

From the earlier definition of the Haematology Department several alternatives for configuring (type of workstations and number of technicians) the department could be evaluated. We first analyzed the existing situation (*hem_std.3te*). It appeared that all rules realized a maximum output around the mean ($mn_out=495$). The minimum value of mn_out was 471 samples, the maximum was 496 samples.

variable	HEM_STD.3TE			HEM_STD.2TE		
	mean	best	worst	mean	best	worst
rule		$\{\theta_2, \theta_7\}$	$\{\theta_4, \theta_7\}$		$\{\theta_2, \theta_7\}$	$\{\theta_3, \theta_6\}$
mn_out	495.32	496.33	470.96	307.82	424.59	228.36
sd_out	59.75	60.89	60.76	63.93	64.87	58.36
mn_left	118.92	120.60	98.95	182.57	224.49	53.53
sd_left	26.08	26.26	16.16	60.27	56.75	11.02
transfer	49.60	48.92	48.09	52.09	49.23	52.27
sd_transfer	7.18	6.85	6.74	3.68	3.20	3.68
mn_working	61.87	59.59	59.70	60.38	67.63	48.68
sd_working	5.27	4.61	5.84	4.83	4.27	7.52
mn_late	0.03	0.00	0.03	1.45	0.28	1.68
sd_late	0.19	0.00	0.17	1.52	0.67	2.43
util1	30.66	30.58	30.61	43.02	32.58	43.22
util2	24.51	25.53	23.45	17.62	13.99	22.14
util3	32.07	30.07	32.34	27.00	36.64	28.89
util4	12.75	13.83	13.59	12.36	16.79	5.76
utop1	37.44	37.46	37.37	30.88	32.54	26.45
utop2	14.00	14.00	14.01	10.70	12.13	9.87
utop3	29.57	28.96	28.45	17.84	23.89	13.86
utop4	13.69	13.93	13.61	7.61	10.16	4.88
thr1	0.44	0.36	0.37	3.22	0.77	4.84
thr2	1.32	1.34	1.42	1.45	1.57	1.46
thr3	1.35	1.40	1.37	1.43	1.57	1.54
thr4	2.95	2.90	2.92	3.94	4.26	3.00
thr5	9.58	9.78	7.16	20.30	19.83	7.54

Table 5.8 Experiment with decrease in staff for haematology department from 3 to 2 technicians.

Therefore it can be concluded that in the *hem_std.3te* configuration rules do not have a significant influence on the performance of the shop with regard to the *mn_out*. Therefore we decided to test alternative scenario's. Here we give as an example the scenario where for the actual workload mix a reduction in staff is considered (from 3 to 2 technicians, *hem_std.2te*). From the results (table 5.8) one can immediately conclude that a change in the number of staff influences the performance of the shop. The mean output reduced for example from 495 to 308.

The effect of rules on mn_out for this department are given in table 5.9. The diagonal of the table shows the main effects of each θ_i separately, the other cells show effects were two θ_i are used in rules.

θ_i	1	2	3	4	5	6	7	8	9	10	11
1	0	-	0	0	0	0	316	261	0	-	274
2		-	251	0	-	320	425	389	-	-	0
3			0	239	0	228	0	274	0	229	251
4				0	0	-	-	318	0	275	0
5					0	0	316	261	0	-	274
6						0	0	0	0	0	367
7							316	316	274	-	317
8								261	402	261	273
9									0	0	355
10										-	321
11											274

Table 5.9 Effects of rules on mean output of *hem_std.2te* (2 technicians). '0' indicates an infeasible rule. '-' indicates an effect which cannot be discerned from the mean ($mn_out=298$) by the method used.

From this table some conclusions can be drawn with regard to effects on mn_out . The rule that combines θ_2 and θ_7 is superior. The main effect of θ_7 is significantly better than the mean, but not much. The combination θ_2 and θ_8 performs also very good. The main effect of θ_8 is however lower than the mean. This is true for most main effects: they are not significantly different from the mean (θ_2, θ_{10}), less than mean (θ_8, θ_{11}) or not feasible (all other θ_i). Other rules which have high positive effects are: $\{\theta_2, \theta_6\}$, $\{\theta_8, \theta_9\}$, $\{\theta_6, \theta_{11}\}$ and $\{\theta_9, \theta_{11}\}$.

To evaluate the impact of rules on the performance of the shop one could consider the use of a regression model. Such a regression model is also called a *metamodel* (Kleijnen (1992)). The regression model becomes:

$$mn_out = B_0 + \sum_{j=1}^k B_j \theta_j + \sum_{j=1}^k \sum_{g=1}^k B_{jg} \theta_j \theta_g + \dots + e \quad (5.12)$$

Here mn_out is the response of the simulation. B_0 is the overall mean; B_i is the main or first-order effect of factor θ_i ($i \in \{1..k\}$); B_{ij} is the two-factor interaction between the factors θ_i and θ_j ($i \in \{1..k\}$, $g \neq j$); B_{ii} is the quadratic effect of factor θ_i ; e denotes fitting errors. According to Kleijnen (1992) it is in general not necessary to include interactions with a higher order than two. We screened for 3rd order interactions and tested the practical significance of taking these 3rd order

interactions into account in the regression model. It appeared that no 3rd order interactions occurred. In our system of defining rules also no quadratic effects occur.

The advantage of the metamodelling approach is that the effect of combinations of θ_i can be estimated without simulations. We tested this approach in two ways. We first used a design in which maximally two θ_i were used. In the other method we used a fractional factorial design. If the number of interactions is limited (in our situation only second order interactions) the fractional factorial design approach is much more efficient than a full factorial design. If however the order of interactions is not known in advance the choice of a particular fractional factorial design can lead to wrong estimations. For further discussion on experimental design approaches we refer to Rotmans & Vrieze, 1990. In the following we will present the results based on simulation experiments where maximally two θ_i 's had the value 1.

Data with which to determine the regression equation was not always accurate (when the number of samples late exceeded 10% the *mn_out* was set to zero). Some θ_i 's are not effective in certain configurations, but they can occur in a combination with another θ_i in a effective rule (less then 10% late). If this situation occurs one has the choice between two strategies. First eliminate the data from the regression analysis that are not effective, but allow all θ_i 's to be entered in the regression model. The result is a correct regression model with possibly a low standard error. But this model will be difficult to interpret as single θ_i 's may occur in the model that are not effective and without simulation one can not be sure if the regression model gives valid results. The second strategy is not to eliminate the non-effective rules and set the values of the output of these rules to zero. The resulting regression model will have a high standard error. The practical value of such a regression model is limited. These considerations led us to conclude that a metamodelling approach is in general not suitable here.

When having determined the optimal situation for *mn_out* there might also be other criteria which are important, as for example the throughput times of the various service classes. It appears that in the *hem_std.3te* and in the *hem_std.2te* examples the mean throughput times for the first priority class are too large (only throughput times less than 15 minutes are acceptable. In the *hem_std.2te* instance the throughput times of the highest service category are even higher than those of service category 2 and 3.

θ_i	1	2	3	4	5	6	7	8	9	10	11
1	0	8.27	0	0	0	0	0.43	0.40	0	4.69	3.04
2		8.27	5.94	0	8.27	9.59	0.77	12.94	7.71	6.98	0
3			0	5.35	0	4.84	0	8.26	0	4.27	5.94
4				0	0	0	0.46	0.42	0	0.56	0
5					0	0	0.43	0.40	0	4.69	3.04
6						0	0	0	0	0	1.43
7							0.43	0.48	0.42	0.42	0.44
8								0.40	1.15	0.58	0.43
9									0	0	2.09
10										4.69	5.59
11											3.04

Table 5.10 Effects on throughput time of the first priority class (*thr1*) of haematology department with two technicians (*hem_std.2te*). Mean throughput time: 4.5 hour.

		mn_out		
		no effect	decrease	increase
thr1	no effect	-	-	-
	decrease	$\theta_7, \theta_2 * \theta_3$	θ_8	$\theta_6 * \theta_{14}, \theta_2 \theta_7, \theta_9 * \theta_{14}$
	increase	$\theta_2, \theta_3 * \theta_8, \theta_4 * \theta_{10}, \theta_7 * \theta_8$	θ_3	$\theta_8 * \theta_2, \theta_8 * \theta_9$

Table 5.11 Effects of rules with regard to mean output (*mn_out*) and/or throughput time of the highest priority class (*thr1*) for haematology department with two technicians (*hem_std.2te*).

In table 5.10 (but see also table 5.11) it appears that θ_7 and θ_8 have a positive effect on *thr1* meaning that the throughput time is relatively low. Around the mean value of *mn_out* the effect of θ_7 and θ_8 is almost the same, but when *mn_out* increases *thr1* increases exponentially when θ_8 is active. This is not the case when θ_7 is active. If *thr1* is low, the standard deviation of *thr1* is also low (0.05 for *mn_out*=0.40), but the standard deviation increases quickly for higher values of *mn_out* (for example it is 0.20 for *mn_out*=0.43). In table 5.11 the rules are given that have an influence on the maximum *mn_out* and/or *thr1*.

Example: A merged job shop

To demonstrate how the performance of rules is influenced by the configuration we defined a new job shop. This job shop is a merger between the existing and calibrated job shop of haematology and the job shop of discrete chemistry. To this new job shop only 4 technicians were allocated, which is 1 less than in the original situation. The reason for this merger was that it seemed that the discrete chemistry department had too much capacity (2 technicians constantly) available, but on the other hand simulation experiments with only 1 technician showed that 1 technician was not enough. A combination with the haematology department might make the situation more efficient. This merged department is also very suitable to test the effect of allowing labour transfer, because to improve the utilization of technicians of the former discrete chemistry department intense use of labour transfer is necessary. This intense use of labour transfer makes it also possible to conduct a sensitivity analysis on the transfer time. Therefore the effect of rules on the *mn_out* of this new department were estimated for transfer times 2.5 and 5 minutes. We tested all rules with only one θ_i and all combinations of two θ_i 's. Here we determined a regression model as only one rule appeared to be not effective. The results are given in table 5.12.

var	transfer time=2.5 min R ² =0.89 SE=6.35 N=66	transfer time=5 min R ² =0.79 SE=7.20 N=66
	B	B
B ₀	274.26	307.49
θ_2	25.69	12.36
θ_3	27.10	27.66
θ_4	6.57	8.88
θ_6	8.73	5.19
θ_7	-7.66	-
θ_9	25.34	10.95
θ_{11}	8.21	10.82
$\theta_1 * \theta_2$		21.46
$\theta_1 * \theta_5$		-21.58
$\theta_1 * \theta_6$	21.03	-
$\theta_2 * \theta_3$	-23.15	-
$\theta_2 * \theta_5$		21.46
$\theta_2 * \theta_9$	-21.79	
$\theta_5 * \theta_6$	21.03	
$\theta_6 * \theta_7$	28.33	
$\theta_6 * \theta_8$	20.60	
$\theta_7 * \theta_{10}$		-22.20

Table 5.12 Regression models for *mn_out* for merged department.

Some observations can be made. First regarding the mean production that the new shop is able to produce. The maximum output is substantially lower than in the *hem_std.2te* scenario. The second observation is that the production for a transfer time of 5 minutes is higher than when this transfer time is 2.5 minutes. The third

observation is that the effectiveness of rules is very different from their effectiveness in the haematology department.

We will discuss the problem of transfer times. If labour transfer costs time, then shortening transfer time does not automatically improve *mn_out*. As the transfer time goes down, the transfer rate will increase. The net effect of this increase can imply a decrease in available production time. Another effect of high labour transfer can be that the planning of samples which do not have a high priority becomes more inefficient. In other words especially high priority samples are favoured by a high labour transfer although the efficiency of the shops then decreases of course. This is immediately evident if we would decrease the transfer time to 0. In the merged department the mean input increases to 873 samples. All possible rules appear to be effective and all rules produced a *mn_out* around the average *mn_out* for all rules. One can explain this result by realizing that our operational rules are used as means to optimize the assignment of labour. If transfer does not cost time, the transfer rate will be maximized. To check this we estimate the influence of the transfer rate on *mn_out* and *thr1* via a regression model (table 5.13). In the regression model the dummy variable transfer is set to 0 for 2.5 minutes transfer time and to 1 for 5 minutes transfer time. The β 's and B's values indicate that the transfer rate has a high impact on the mean output.

var	Effect on <i>mn_out</i> $R^2=0.87$ SE=8.34 N=132	
	B	β
B_0	277.54	
transfer	30.11	0.69
θ_2	21.49	0.45
θ_3	27.07	0.69
θ_4	6.58	0.11
θ_6	6.69	0.11
θ_7	-9.34	-0.16
θ_9	17.80	0.30
θ_{14}	7.05	0.12
$\theta_2_ \theta_3$	-18.22	-0.10
$\theta_2_ \theta_9$	-18.88	-0.11
$\theta_5_ \theta_8$	-24.71	-0.14
$\theta_6_ \theta_7$	23.02	0.13

Table 5.13 Influence of transfer time for transfer time is 2.5 and 5 minutes.

5.6.4 Workload control

We applied workload control to the Haematology Department in order to reduce the throughput time of rush samples. We made however two adaptations to the earlier example concerning the haematology department. During rush hours we assigned 3 technicians and during other hours 2 technicians. Moreover it was assumed that for each service category (except the one with the highest priority) 30% shifted to the next higher category.

The results of workload control were rather disappointing. It appeared that only 11 rules with workload control were effective. All these rules used θ_{11} . These effective rules were compared with the situation where no workload control was applied. Here we are especially interested in *thr1* because we especially implemented workload control to reduce the throughput time of rush orders. In table 5.14 the results are given for the instances where the rules with θ_{11} are applied.

	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}
W	1.50	3.15	3.70	1.00	1.60	2.01	1.02	0.84	2.23	2.91
NW	3.14	3.32	3.46	1.19	1.60	2.01	1.20	0.84	2.23	2.91

Table 5.14 Throughput time for the first priority class (*thr1*) (mean output (*mn_out*) is 425 samples) for combinations of θ_{11} with other θ_i for the situation with (W) and without workload control (NW).

Workload control is most effective in the combination $\{\theta_8, \theta_{11}, \text{workload control}\}$. It gives the shortest throughput time for a *mn_out* of 425 samples. But in general it appears that θ_7 and θ_8 give very good throughput times for *thr1* (table 5.15).

	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	θ_{11}
θ_4	1.19	3.42	3.83	3.26	1.19	2.98	0.89	0.99	3.33	1.36	1.19
θ_7	0.94	1.42	1.52	0.89	0.93	0.94	0.93	0.93	0.99	0.99	1.01
θ_8	0.80	0.72	3.43	0.99	0.80	0.96	0.93	0.80	1.33	1.22	0.99

Table 5.15 Throughput time of first priority class (*thr1*) (mean output (*mn_out*) is 425 samples) for combinations of θ_4 , θ_7 and θ_8 without workload control.

5.6.5 Scenario analysis

It is interesting to see how the importance and significance of rules change when a shift in the workload mix occurs. For this investigation the *scenario_analyzer* can be used. As an example we formulated 4 additional scenarios for the Haematology Department:

3 technicians: 30 percent shift of demand to next higher service category (*hem_30p.3te*);

- 3 technicians: 30 percent shift of demand to next lower service category (*hem_30m.3te*);
- 2 technicians: 30 percent shift of demand to next higher service category (*hem_30p.2te*);
- 2 technicians: 30 percent shift of demand to next lower service category (*hem_30m.2te*);

The scenarios with 3 technicians did not give any significant changes ($\alpha=0.05$) on the maximum *mn_out* compared to the nominal scenario. In table 5.16 for each scenario with 2 technicians the mean *mn_out*, maximum *mn_out* and the rules which realize the maximum *mn_out* are given.

scenario	mn_out	max mn_out	rules for max mn_out
hem_30m.2te	315.29	436.47	$\{\theta_8, \theta_9\}$
hem_std.2te	297.72	424.59	$\{\theta_2, \theta_7\}$
hem_30p.2te	272.50	389.11	$\{\theta_2, \theta_7\}$

Table 5.16 Most successful rules in scenarios for haematology.

The information in table 5.16 gives an immediate insight in the capacity of shop configurations under various scenarios if the maximum *mn_out* is the criterion. It also suggest the most effective rules.

θ_i	1	2	3	4	5	6	7	8	9	10	11
1				229 0.85			308 0.73	238 0.70			
2											
3							239 0.83				
4				229 0.85	229 0.85		297 0.71	298 0.72			229 0.85
5							308 0.73	238 0.70			
6							192 0.75				
7							307 0.73	296 0.76		286 0.85	286 0.81
8								239 0.69			193 0.69
9											
10											
11											

Table 5.17 Effects on mean output (*mn_out*, upper figures) and throughput time of the first priority class (*thr1*, lower figures) for scenario with 30 percent increase in service and two technicians (*hem_30p.2te*, only for *thr1* ≤ 1h).

Again it is interesting to investigate to what extent rules influence the throughput time of the first service category. In table 5.17 and table 5.18 we present maximum *mn_out* and *thr1* for those rules that have a *thr1* of less than one hour. Here again it appears that in the *hem_30m.2te* scenario rule $\{\theta_8, \theta_9\}$ performs better than rule $\{\theta_2, \theta_7\}$.

θ_i	1	2	3	4	5	6	7	8	9	10	11
1				368 0.42			315 0.40	262 0.40			
2							425 0.78				
3							307 0.63				
4				368 0.42	368 0.42		316 0.46	329 0.43		309 0.68	368 0.42
5							315 0.48	262 0.40			
6							389 0.42	331 0.43			
7							315 0.49	317 0.50	390 0.44		287 0.43
8								262 0.41	436 0.50	261 0.46	
9											
10											
11											

Table 5.18 Effects on mean output (*mn_out*, upper figures) and throughput time of the first priority class (*thr1*, lower figures) for the scenario with 30 percent decrease in service and two technicians (*hem_30m.2te*, only for *thr1* ≤ 1h).

5.6.6 Conclusions on optimization with simulation

Experiments with the rules indicate which rules are most effective in certain department configurations. It is also possible to test the effects if for example the number of available technicians or the transfer time are changed. The scenario-analysis showed that a change in the required service degree influences the optimality of rules. It also appeared that due to the criterium used in our binary search procedure it was not possible to use metamodelling as a general method for determining the sensitivity of the performance on θ_i 's.

5.7 Simulation and the DSS

The simulation module is essential for both the *experimenter* and the *scenario-analyzer*. By the manner in which we implemented operational rules we obtained a very flexible way to test well-known rules from the literature, other used rules and new rules.

It is interesting to observe here that the output of the simulation results led to discussions with the technicians involved in the management of the laboratory. The simulation output highlighted effects of rules and thus increased the insight of the management of the laboratory.

A problem is data gathering and data management. A substantial effort was necessary to gather the data. Of course many data were gathered as part of the system analysis for developing the DSS. On the other hand many additional data are needed to conduct valid experiments with the simulation model. How realistic is it to expect that these data are available in a laboratory on a regular basis? Our experience is that much is known by the technicians, the challenge is to use this data and to obtain information from it. It may be expected that once technicians involved in management are used to the DSS they will be able to feed the simulation module with their own data. It can of course not be expected that this data is always accurate or that the data is instantly correct. The DSS should provide help via sensitivity analysis. An example is the experience we had with *calibrating the Discrete Chemistry Department (two specific test analyzers)*. It appeared very difficult to estimate the parameters of the workstations. However by the taking extreme values of parameters that are difficult to estimate the technicians could easily conclude from the output of the simulation module what the correct parameter values had to be. It is clear that the data management function is very important here. One may also expect that in practice once the management of the laboratory is accustomed to our DSS data gathering is not a real problem.

Another possibility is to gather information on a routinely basis from the Laboratory Information System (LIS). Although it seems that in a fully automated laboratory this is easy, this is not true. Accurate information for our purposes presupposes a kind of sample tracking system where samples are not only registered at one or more moments during processing, but also when they are assigned to a queue. If no registration is performed when a sample arrives at a queue, no reliable information can be obtained from a laboratory information system. If a LIS would also record the above mentioned data the integration of the DSS with a LIS would be worthwhile.

6

THE OPTIMIZATION MODEL

The work described in this chapter has been developed in co-operation with Maarten Oosten, Mathematics Department, University of Limburg.

6.1 Introduction of the DSS

In this chapter the optimization module is discussed. This model aims to find an optimal clustering of workstations into job shops. In contrast with the simulation model of chapter 5 where the optimization of individual job shops was the goal and detailed data was used and detailed information was provided, here the data and information are mainly of an aggregate kind. Considerations for this approach have been discussed in chapters 2 and 3.

We first define an optimization model that takes all principle aspects of the deterministic problem into account (section 6.2). In the following section we will extend this model by dropping some of its underlying assumptions (6.3). In section 6.4 we discuss how the model can be solved. In section 6.5 an example is discussed. In section 6.6 the consequences for the DSS are discussed.

6.2 The deterministic approach

We first define a model in which the demand is exactly deterministic. This means that the future demand is predictable. The volume and service levels for the various process types are also exactly known. The demand pattern is given for a period of one shift (9 hours).

6.2.1 Introduction

A laboratory consists of workstations. A workstation is a combination of equipment and technicians that perform a certain process type. A type of process consists of a group of tests that can be performed on a certain machine and that has the same processing times $w(p)$. Here p denotes the process type. Workstations are grouped together into a job shop. The total number of job shops is indicated by N . Technicians are always assigned to job shops and directed to workstations. This means that with regard to staff assignment, we have a two-phase scheduling approach. First we assign staff to job shops, then staff is assigned to workstations. The total available staff for the laboratory remains constant during one shift. A shift is however divided into periods. At the beginning of each period technicians are assigned to job shops and they remain there for the whole period. In fact in a deterministic situation technicians are scheduled to job shops by the senior technician who is in charge of the routine management. This schedule was called in chapter 4 *manpower plan*. Within each period and within the job shop the individual orders, denoted by the index i , are assigned to workstations. Each order i has a certain process type $p(i)$. The type of process determines which workstation is needed. This is characteristic of our approach: a sample type determines the process types and this again determines the workstation. The processing time of each order is known and constant. So only a decision about when to start the production process has to be made. This scheduling has to take place in such a way that the processing of this order will be ready before its due date $d(i)$. The workstation and the human resources needed are implied by this scheduling. The

type of process, once assigned to a workstation, cannot be re-assigned. In a deterministic approach the demand for orders is assumed to be known precisely.

6.2.2 Problem definition

We are interested in optimizing the efficiency of the utilization of human resources. This goal was operationalized as minimizing the idle time of technicians within a shift. A shift consists of T time units t . Within each shift assignment periods e are identified. Each period has a begin e_b and an end e_e . The parameters e_b and e_e of a period have a value between 0 and T where $e_b \neq e_e$. Figure 6.1 presents an example where a shift consists of 4 assignments periods ($E=4$).

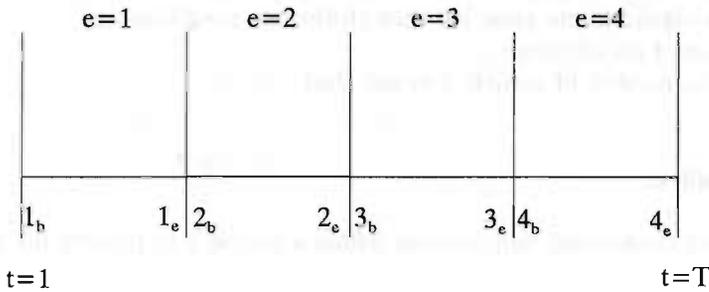


Figure 6.1 Composition of a shift.

The total number of technicians (denoted as U) in the laboratory is constant (between $t=0$ and T). U is therefore the maximum number of technicians that may be allocated to job shops in each assignment period. Within each job shop the number of technicians can vary between the different periods. It is also possible that not all technicians in a certain assignment period are assigned to job shops. Before presenting a formal description of the optimization model we have to introduce the appropriate parameters and variables.

list of parameters:

- L_{lb} : the minimum amount of work expressed in technician time that has to be done at a job shop in a single period e ;
- L_{up} : the maximum amount of work that is allowed to be performed at a job shop in one period ;
- $k(i)$: the arrival time of order i ;
- $d(i)$: the due date (due time) of order i ;
- $w(p)$: the processing time for an order of process type p ;
- $J(p)$: the set of job shops that may process orders of type p ;
- $p(i)$: process type of order i ;
- $I(p)$: set of all orders of type p ;
- F : set of pairs (p,q) of process types p and q which are not allowed to be assigned to the same job shop (forbidden combinations);
- J : set of all job shops;
- E : the number of periods e in one shift.

list of variables:

- L_{je} : the total (technician) time needed within a period e to process the jobs in job shop j ;
- Z : upper bound of the idle time of all technicians in one period;
- U_{je} : the available number of technicians in job shop j in period e ;
- B_{pj} : 1 if a process type p is assigned to job shop j ; 0 otherwise;
- A_{ijt} : 1 if order i is scheduled to be processed at time t at job shop j ; 0 otherwise;
- U_e : the number of technicians in period e .

The optimization model for technicians is:

Minimize Z

subject to

$$\sum_{j \in J(p)} B_{pj} = 1 \quad (\forall p), B_{pj} = 0 \quad (\forall j \notin J(p)) \quad (6.1)$$

$$A_{ijt} \leq B_{p(i)j} \quad (\forall i, \forall j, \forall t) \quad (6.2)$$

$$\begin{aligned} \sum_{j \in J(p)} \sum_{t=0}^T A_{ijt} &= 1 \quad (\forall i \in I(p)), \\ A_{ijt} &= 0 \quad (\forall t \in \{e_c - w(p(i)), \dots, e_c\} \forall e \\ &\quad \forall t \in \{d(i) - w(p(i)), \dots, T\} \\ &\quad \forall t \in \{0, \dots, k(i)\}) \end{aligned} \quad (6.3)$$

$$B_{pj} + B_{p_2j} \leq 1 \quad (\forall j, \forall (p_1, p_2) \in F) \quad (6.4)$$

$$L_{lb} \leq L_{je} \leq L_{up} \quad (\forall j, \forall e) \quad (6.5)$$

where:

$$\sum_{i \in I} \sum_{t=e_b}^{e_c - w(p(i))} A_{ijt} \cdot w(p(i)) = L_{je} \quad (\forall j, \forall e) \quad (6.6)$$

$$L_{je} \leq U_{je} (e_c - e_b) \quad (\forall j, \forall e) \quad (6.7)$$

$$\sum_j U_{je} \leq U \quad (\forall e) \quad (6.8)$$

$$\sum_{e=1}^E \sum_j (U_{je} (e_c - e_b) - L_{je}) = Z \quad (6.9)$$

$$A_{ijt} \in (0, 1] \quad (6.10)$$

$$B_{pi} \in \{0, 1\} \tag{6.11}$$

$$U_e \in \{0, 1, \dots, U\} \tag{6.12}$$

Constraint (6.1) expresses that all types of processes have to be assigned to a job shop and each type of process can be assigned only once. Constraint (6.2) assures that an order i is scheduled at a job shop to which the corresponding process type $p(i)$ is assigned. Constraint (6.3) describes that the processing of an order must begin between $t=k(i)$ and $t=e_e-w(p(i))$ where $k(i)$ denotes the release time of order i , e_e is the end of a period and $w(p(i))$ is the production or process time of that order. This process time is determined by the type of process to which order i belongs. Constraint (6.4) applies to types of processes that are not allowed to be assigned to the same job shop. Constraint (6.5) aims at giving the shop a minimum (L_{lb}) and a maximum size (L_{up}) in workload (expressed in process time). Both minimum and maximum size are determined by the management of the laboratory. The minimum size is mainly determined by efficiency considerations (a certain minimal workload should be processed). The maximum size is determined by span of control considerations: there is a maximum size of job shops that can be controlled. Equation (6.6) defines L_{je} , the total time needed to process samples in period e in job shop j . This time is calculated by multiplying all orders i that have been scheduled for this period by the processing times of the process types involved. Constraint (6.7) assures that the number of technicians is sufficient in all periods. Constraint (6.8) expresses that for each period $\sum U_{je}$ does not exceed the total number of technicians.

Object function (6.9) defines Z as the idle time (the difference between the available capacity and the needed capacity) in a shift.

6.2.3 Comments on this optimization model

In this model some choices are built-in. The first choice is that the only resource to be optimized is labour (number of technicians). This choice is in most cases realistic, but not always. A second choice is that all production processes that have been started in a certain period e will be completed within period e . For orders with small processing times this assumption does not have much influence upon the optimality of the solution. However for orders with a large processing time (e.g. 2 hours), the optimality of the solution may be negatively influenced. The third choice is that there is no direct assignment of orders to technicians.

Before we consider the relaxation of the second choice, we first discuss another problem, namely the number of variables that has to be taken into account. This number is too large to find a solution within a reasonable period of time. The number of variables is determined by the number of job shops N , the number of

periods within a shift, the number of different types of processes P, the number of time units T, and the number of individual orders I. The number of job shops N varies between 4 and 10, the number of periods per shift between 2 and 4 and the number of processes between 15 and 30. The number of time units T will generally be much higher in order to assure proper assignment of orders. The smallest processing time $w(p)$ determines the unit of t. This means that if the smallest $w(p)$ is 2.5 minutes and we have an 8 hours shift, T equals 240. This causes of course an explosion in the number of variables. The number of orders I also poses a problem since this number may be approximately 1000 per day.

In order to reduce the number of variables the optimization model is adapted in such a way that those parameters that cause an explosion in variables are replaced. We shall first look at the number of orders. An approach could be that we are not considering individual orders, but fractions of the total number of orders of a certain type. The constraints (6.2), (6.3), (6.6) and (6.10) then become:

$$\sum_{e=1}^E f_{pje} \leq B_{pj} \quad (\forall p, \forall j) \quad (6.2')$$

$$\sum_{j \in J(p)} \sum_{e=1}^E f_{pje} = 1 \quad (\forall p) \quad (6.3')$$

$$\sum_{p \in P} f_{pje} \cdot |I(p)| \cdot w(p) = L_{je} \quad (\forall j, \forall e) \quad (6.6')$$

$$f_{pje} \in [0, 1] \quad (\forall p, \forall j, \forall e) \quad (6.10')$$

where f_{pje} denotes the fraction of the total number of orders of type p that has to be processed in period e. Constraint (6.2') assures that a fraction of orders of production process p can only be assigned to job shop j if $B_{pj}=1$, i.e. if process p is assigned to job shop j. Evidently this should hold for each period e. Constraint 6.3' assures that all orders will be scheduled.

In the following we will use (for reasons of presentation) the amount of orders N_{pje} instead of fractions:

$$f_{pje} \cdot |I(p)| = N_{pje} \quad (6.10'')$$

The advantage of working with fractions instead of A_{jt} is that now periods are considered instead of time units. Individual orders do not longer need to be considered. This however also introduces a difficulty: what to do with constraints related to individual orders, release- and due-times? The consequence of the 'fractions' approach is that we have to set release times at the beginning e_0 of the

period e in which the sample is released and the due-time at the end of period in which the order is due. $k(i)$ becomes equal to e_e of period f (the period in which the order is due). This new due-time is denoted as f_e . This simplification introduces rounding problems. To these problems we will return later. The number of orders of process type p with release times e_b and due time f_e is called δ_{pef} .

Scheduling of processes

It is possible to make the process of scheduling orders more explicit by replacing (6.2'), (6.3') by:

$$\sum_{e=1}^f N_{pje} \leq \sum_{e=1}^f \sum_{h=e}^E \delta_{peh} \quad (\forall p \in P, \forall j \in J, \forall f : 1 \leq f \leq E) \quad (6.2'')$$

$$\sum_{e=g}^f N_{pje} \geq B_{pj} \cdot \left[\sum_{e=g}^f \sum_{h=e}^f \delta_{peh} \right] \quad (\forall p \in P, \forall j \in J, \forall g, \forall f : 1 \leq g \leq f \leq E) \quad (6.3'')$$

Constraint (6.2'') assures that the number of samples scheduled in the first f periods does not exceed the number of samples that is released in the first f periods. Constraint (6.3'') assures that the scheduled number of orders in every subset of consecutive periods is at least as large as the number of orders that has to be processed in these periods due to their release times and due times. In (6.3'') the variable B_{pj} is fully derived from the variable N_{pje} since $B_{pj} \in \{0,1\}$ $B_{pj}=1$ if and only if N_{pje} is positive in at least one period e . If furthermore integrality conditions are needed one may consider:

$$N_{pje} \in \{0, 1, \dots, \sum_{e=1}^E \sum_{h=e}^E \delta_{peh}\} \quad (6.10''')$$

Finally some remarks on the goal of the optimization model: the minimization of the number of technicians needed per shift in the laboratory. The practical value of this goal is often limited since the number of technicians may be given or there may exist in fact a scarceness of technicians. Therefore it was decided that the optimization model should minimize the maximum idle time of technicians at job shops. Then technicians can be scheduled free to do other jobs (e.g. research or administrative activities). This is expressed by replacing constraints (6.8) and (6.9) by

$$\sum_{j=1}^N U_{je} \leq U \quad (\forall e) \quad (6.8')$$

$$Z_{je} = U_{je}(e_e - e_b) - \sum_{p \in P} w(p) \cdot N_{pje} \quad (\forall j, \forall e) \quad (6.9')$$

$$Z = \max \{Z_j, \forall j, \forall e\}$$

(6.14)

6.3 The extended model

6.3.1 Introduction

The underlying assumptions in the above model should be reconsidered. First we discuss the assumption of determinism and the consequences of allowing a stochastic demand. We will use a simple example to clarify the problem. Next we deal with the fact that no distinction is made between process times of technicians and the process time of workstations. Finally the problem of rounding the outcomes of the optimization model will be discussed. This last problem is due to the fact that fractions of orders are scheduled and not individual orders.

6.3.2 The influence of unpredictability

The model in section 6.2 discusses the situation in which it is known beforehand how many orders with a certain priority will arrive and at what time. Thus everything is known with certainty. It is obvious that predictability of the arrival times of orders is far from reality. It is necessary to adapt the model in such way that unpredictability on the arrival of orders is taken into account. How should the model be adapted for this problem?

The impact of unpredictability on the arrival of samples is especially important for rush orders. These orders often have a due-time in the same period as they arrive. In a deterministic situation these orders do not cause any problem, because one can simply reserve as much capacity as the process time of rush orders requires. In a situation of unpredictability more capacity (slack capacity) should be reserved for being able to realize the required service levels. Slack capacity means idle time for technicians which is something we want to avoid as much as possible. In developing the model for the stochastic situation we assume that we are informed about the mean demand for various processes and priority levels (and thus the due times) and about the variances in demand. We use variance as a measure for variation. Especially the variance in *planning freedom* is important. Planning freedom is the degree to which the laboratory is free to process an order at a later period than the period of arrival of orders at the shop. We shall later work this concept out more formally. It is evident that if planning freedom does not exist, we have only slack capacity available as a tool. In the stochastic model we can use variations in planning freedom in order to reduce the variation in demand by a certain way of clustering.

Before offering a solution, we will first discuss by means of a simplified example how differences in demand between periods are dealt with in the deterministic model. Table 6.1 displays the processes that have to be allocated to shops. The

maximum workload that can be processed by a technician in one period is assumed to be 35 workload units. Similarly, the maximum number of technicians in a shop is assumed to be 2. This is a constraint as meant in (6.5). In a deterministic situation the allocation of processes to shops as presented in table 6.1 will be optimal. Here we have an idle capacity of 30 workload units (the maximum of idle capacity of each jobshop over all periods) in technician capacity.

shop	Process	Period 1	Period 2	Total	Max. Idle capacity
1	p_1	20	30	50	
1	p_3	10	10	20	
1	p_5	30	20	50	
idle		10	10	20	10
2	p_2	40	20	60	
2	p_4	10	30	40	
idle		20	20	40	20

Table 6.1 Optimal allocation of processes to shops.

In the deterministic situation we have only one criterion that has to be optimized, namely idle time of technicians. Suppose however that there is unpredictability about the actual workload to be processed during a period.

Now there is a risk that a shop is not able to process the required workload. If we measure unpredictability in the actual demand by the variance in demand we get table 6.2. Here we deal with a totally new situation, because to allocate capacity to shops not only the mean workload, but also the variance in workload should be taken into account. The most simple approach would be to reduce the maximum workload for each technician. This would be sufficient if for all process types in all periods the variance is the same. This is however not very realistic. A slightly more sophisticated approach would be to add a certain amount of slack capacity ('surplus') to the mean workload on the basis of the variance of workload for each process. The fraction of the variance that is added to the mean workload is dependent on the risk the decision maker is willing to accept with respect to the probability that the actual workload cannot be processed. Suppose that this chance for all processes is set to 0.05 and that the actual workload has a normal distribution, then the workload to be added to account for the variance can be calculated.

P	period 1				period 2				tot
	mean	var	surplus	tot	mean	var	surplus	tot	
p ₁	20	10	5.20	25	30	15	6.37	36	61
p ₂	40	15	6.37	46	20	7.5	4.50	25	71
p ₃	10	5	3.68	14	10	5	3.68	14	28
p ₄	10	10	5.20	15	30	25	8.23	38	53
p ₅	30	30	9.01	39	20	3	2.85	23	62
tot	110		29.46	139	110		25.63	136	275

Table 6.2 Mean and variance in workload and correction.

As can be seen in table 6.2 the total surplus of workload to be added equals 55.09 which is 25% of the mean workload. The new total workload (275) can now be allocated to the shops in the same manner as we did in the deterministic situation. For our previous optimal solution, this would imply that it becomes infeasible, as is shown in table 6.3.

shop	period 1			period 2		
	mean	var	surplus	mean	var	surplus
1	60	45	11	60	23	8
2	50	25	8	50	32.5	9

Table 6.3 Surpluses and variances in the two shops (no correlation assumed).

Of course the idle times did not change on average. But in shop 1 during period 1 the available technicians workload is 70, where for the 0.95 confidence interval 71 minutes are needed. By reallocating processes to the job shops a feasible solution

P	period 1			period 2		
	mean	var	surplus	mean	var	surplus
p ₁	20	10		30	15	
p ₂	40	15		20	7,5	
shop 1	60	25	8	50	33	8
p ₃	10	5		10	5	
p ₄	10	10		30	25	
p ₅	30	30		20	3	
shop 2	50	45	11	60	33	9

Table 6.4 Feasible solution (no correlation assumed).

can be found however, as is shown in table 6.4. The problem with this approach is that it assumes that the workload of the various processes do not co-variate. If they do actually co-variate then it may appear that after allocating the processes to shops, the actual risk of not being able to process the work is more or less than the risk the decision-maker is willing to accept (in this case 0.05). This means that either the shop is not able to process the required workload (the shop is not effective) or there is more idle capacity than is needed to cope with unpredictability.

Assume that the correlation coefficients of the demands of processes are identical in both periods and that we know their precise values. Then let S_j denote the index set of process types in shop j and let $X_e(S_j)$ be a stochast denoting the total load of a shop j in period e . Then the variance of $X_e(S_j)$ can be computed as follows:

$$\sigma^2(X_e(S_j)) = \sum_{p \in S_j} \sum_{q \in S_j} \rho_{pq} \sigma_p^e \sigma_q^e \quad (6.15)$$

Where $\rho_{ij} \sigma_i \sigma_j$ denotes covariance between processes p and q .

Assume the correlation-coefficients of the processes of our example as presented in table 6.5.

	p_1	p_2	p_3	p_4	p_5
p_1	1	0.5	-1	1	-0.5
p_2	0.5	1	0.25	-0.25	0
p_3	-1	0.25	1	0.5	0.5
p_4	1	-0.25	0.5	1	-1
p_5	-0.5	0	0.5	-1	1

Table 6.5 Correlation coefficients between processes.

If we apply this approach to the situation of table 6.1 we get the situation as given in table 6.6. Applied for the situation where variance was taken into account (table 6.4) the results as given in table 6.7 are obtained.

shop	period 1			period 2		
	mean	var	surplus	mean	var	surplus
1	60	25.78	8.4	60	7	3
2	50	19.00	8.0	50	26	9

Table 6.6 Situation in co-variance approach for first solution.

shop	period 1			period 2		
	mean	var	surplus	mean	var	surplus
1	60	37.25	10.4	60	33.11	9.5
2	50	29.68	9.0	50	30.73	9.1

Table 6.7 Situation in co-variance approach for second solution.

One can conclude that if co-variances are taken into account the original solution as given in table 6.1 can cope with the situation which is not the case when only variances are taken into account.

The approach above resembles the portfolio approach from Markowitz (Harrington, 1987). The essence is that a shop is seen as a portfolio of processes, where to each shop processes are assigned in such a way that the variance of the workload of the shop is less than the sum of variances of the processes allocated to the shops. In essence the portfolio-approach tries to allocate processes to shops that are

negatively co-variated. It is an example of a situation (as noted in chapter 2) where it is possible to reduce risk without extra costs, or where it is possible to reduce costs without extra risk, and thus a rational decision maker should take advantage of this.

Planning freedom

In the above example we did not take into account the situation that a planner has often some freedom to determine the time to begin the processing of a sample. Our optimization model did however assume the existence of buffers, since samples could be processed in another period than in the period they arrived. Thus workload can be transferred to later periods as long as the due time is realized. If there is variation in the total volume of demand, buffers can be used to smooth the production process, and this will result in less capacity needed. Production smoothing is only possible if there is planning freedom. This planning freedom can be defined as:

$$\phi_{pe} = \frac{\max \left\{ \text{maxload}, \sum_{e=1}^f \sum_{h \neq e}^E \delta_{p,eh} \right\} - \delta_{p,ff}}{\text{maxload}} \quad (6.16)$$

where *maxload* is defined as the maximum number of orders of type *p* that can be processed in period *e*. The degree to which buffers can be used depends on the freedom the shop has to schedule the production of some order without violating the external due-date. It is important to realize that the optimization model deals with tactical planning problems and is using information of an aggregate kind. Often detailed information on the future demand is not available. The intuition of the planner is very important here. Therefore we decided to extend the optimization model with a co-variance matrix.

With the co-variance matrix the planner can indicate to what extent he wants to combine processes and allocate them to the same job shop. He can express preferences by assigning penalty weights (expressed as technician time) to combinations of two processes he does not want to combine.

To the optimization model the next constraints are added:

$$\sum_{p \in P} w(p) \cdot N_{p,je} + \sum_{p \in S_j} \sum_{\substack{q \in S_j \\ q \neq p}} \text{cor}(p,q) C_{pqj} \leq U_{ej} (e_e - e_b) \quad (6.17)$$

$$C_{pqj} + B_{qj} - B_{pj} \leq 1 \quad (\forall p, q \in P, \forall j \in J) \quad (6.18)$$

$$B_{qj} + B_{pj} - C_{pqj} \leq 1 \quad (\forall p, q \in P, \forall j \in J) \quad (6.19)$$

$$\sum_{j=1}^N C_{pqj} \leq 1, C_{pqj} \in \{0,1\} (\forall j, \forall p, \forall q) \quad (6.20)$$

(6.19) and (6.20) imply that always $C_{pqj} = C_{qpj}$. The term $cor(p,q)$ can be interpreted as a time penalty for each period where processes p and q are assigned to the same job shop. The actual value of $cor(p,q)$ has to be determined by the planner.

6.3.3 Process times

In the model of section 6.2 no distinction was made between the time needed to process a sample at a workstation and the time a technician has to spend processing a sample. This is a consequence of our choice to optimize the efficiency of technicians in an aggregate way. Although we will not implement it we can give a short outline of a more detailed model.

In chapter 5 we saw that the distinction between the time needed to process a sample at a workstation and the time a technician has to spend for processing a sample is important for two reasons. The first reason is that the time a technician spends to process a sample is only a fraction of the total workstation processing time. The second reason is that this model tries to optimize the utilization of technicians and not of equipment. Therefore in the optimization model for each process type a distinction is made between the processing time per sample at the workstation ($w_1(p)$) and the time the technician is involved in processing the sample ($w_2(p)$). The workstation capacity is expressed by the next constraint:

$$N_{pje} \leq \left\lfloor \frac{e_e - e_b}{w_1(p)} \right\rfloor (\forall j \in J), \forall p \in P \quad (6.21)$$

and constraint (6.17) changing $w(p)$ in $w_2(p)$ becomes:

$$\sum_{p \in P} w_2(p) \cdot N_{pje} + \sum_{p \in P} \sum_{\substack{q \in P \\ q \neq p}} cor(p,q) C_{pqj} \leq U_{ej} (e_e - e_b) \quad (6.17')$$

6.3.4 Rounding problems

The actual optimization model is a linear programming model. As a result we deal with rounding problems. This is e.g. the case if the model proposes a certain fraction of technicians to be assigned to a certain job shop. Obviously the advice should be expressed in integers. Another problem is for example the allocated workload to the job shop if the optimization model allocates a workload of 1.1 technicians to a certain assignment period, simple rounding the figure would result in an advice to assign two technicians to this job shop for that period. However as the optimization model uses in essence information of an aggregate kind the planner should be actively involved in these rounding problems. The co-variance

matrix is also useful here (one can for example assign (negative) weights to control the rounding process).

6.4 Solving the model

In the previous section the problem has been formulated as an integer programming problem. Since the number of feasible solutions is finite, we could compute them all and select the best solution. But in practice this option is of no use, since even for moderate instances the number of feasible solutions is huge. If $f(p,n)$ is the number of ways to divide p process types into at most n job shops, then:

$$f(p,n) := \sum_{k=1}^n S(p,k)$$

where

$$S(p,k) = \frac{1}{k!} \sum_{j=1}^k (-1)^{k-j} \binom{k}{j} j^p$$

is the Stirling number of the second kind, denoting the number of ways a set of p elements can be divided into k non-empty subsets. If for example p is 20 and n is 5 then there are $8 \cdot 10^{11}$ solutions. Even if we had the disposal of a package evaluating 1 million solutions per second (which is extremely fast) and we tried to solve an instance of our problem with 5 shops and 20 process types, then still we needed to run this package for 9 days. Notice that we only accounted for different assignments of process types to job shops, but for each of these assignments there may be different possibilities for the choice of the number of technicians for each job shop in each period.

Since considering all solutions is not feasible, another approach should be considered. For some IP-problems, a 'fast' algorithm could be developed, based on some clever 'tricks', which make it possible to prune the solution space. However, there is a class of problems for which such an algorithm is unlikely to exist. This follows from complexity theory; see for a proper introduction to complexity theory Garey & Johnson, 1979.

One of the problems that is known to be NP-complete, is the so-called bin-packing problem:

Bin-packing:

"Given a set A of items, a size $s(a) \in \mathbb{Z}^+$ for each a in A , and a positive bin capacity B . Find a partition of A into disjoint sets A_1, \dots, A_k such that the sum of the sizes of the items in each A_i is B or less, and such that k is minimal".

In a way the bin-packing problem is a degenerate case of our problem, since each instance of the bin-packing problem can be solved as an instance of our problem in case our objective is to minimize the over-all idle-time. If all periods have equal

length, this objective is equivalent to minimizing the number of technicians. To see the relation between bin-packing and our problem, the reader should verify that our problem can be formulated as follows:

Lab-min:

"Given a set P of elements, a size $\omega(p) \in [0,1]$ for each element p in P , numbers $E, T, T_{\max}, N \in \mathbb{N}$, and an $(E \times E)$ upper triangle matrix $\Delta(p)$ such that each element $\delta(p)_{ef} \in \mathbb{N} \cup \{0\}$.

Find a partitioning of P into N disjoint sets P_1, \dots, P_N , a $(|P| \times E)$ matrix G , and a $(N \times E)$ matrix U , such that: each element $g_{pe} \in \mathbb{N} \cup \{0\}$, each element $U_{je} \in$

$$\{ 0, \dots, T \}, \quad \sum_{e=1}^E U_{ej} \leq T_{\max} \quad \forall j \in \{1, \dots, N\},$$

$$U_{ej} \geq \sum_{p \in P_j} \omega(p) g_{pe} \quad \forall j \in \{1, \dots, N\}, \quad \forall e \in \{1, \dots, E\}, \quad \text{and} \quad \sum_{j=1}^N \sum_{e=1}^E U_{ej}$$

is minimal."

Every nontrivial (i.e. $s(a) \leq B$ for each a in A) instance of the bin-packing problem can be reduced to an instance of lab-min in the following way:

Reduction from bin-packing to lab-min:

$P = A$, $\omega(p) = s(p)/B$ for all p in P , $E=T=1$, $N=T_{\max} = |P|$, and $\Delta(p) = [1]$ for all p in P .

The reader can verify that for each solution of any instance of the bin-packing problem a solution of the corresponding instance of lab-feas can be constructed

such that k equals $\sum_{j=1}^N \sum_{e=1}^E U_{ej}$, and vice versa. In other words: bin-packing is a

special case of our problem. If we are able to construct a fast algorithm (i.e. an algorithm requiring an execution time which is polynomial in the size of the input) solving our problem to optimality, we have also found a fast algorithm to solve bin-packing to optimality. But such an algorithm is very unlikely to exist (we could however look for a fast algorithm which yields good but not necessary optimal solutions; such an algorithm is called a heuristic. However, we proposed to change the objective function from minimizing the over-all idle-time to minimizing the maximum idle-time occurring in any shop in any period. Now the question arises whether our adapted problem is also NP-complete. And if so, could we construct an objective function for which the problem is not NP-complete? To see why the answer to both questions is negative, we firstly formulate the problem of constructing a feasible solution for any instance of our problem as follows:

Lab-feas:

"Given a set P of elements, a size $\omega(p) \in [0,1]$ for each element p in P , numbers $E, T, T_{\max}, N \in \mathbb{N}$, and an $(E \times E)$ upper triangle matrix $\Delta(p)$ such that each element $\delta(p)_{ef} \in \mathbb{N} \cup \{0\}$. Find the answer to the following question: is there a partitioning of P into N disjoint sets P_1, \dots, P_N , a $(|P| \times E)$ matrix G , and a $(N \times E)$ matrix U , such that each element $g_{pe} \in \mathbb{N} \cup \{0\}$, each element $U_{je} \in \{0, \dots, T\}$,

$$\sum_{e=1}^E U_{ej} \leq T_{\max} \quad \text{for all } j \in \{1, \dots, N\}, \quad \text{and} \quad U_{ej} \geq \sum_{p \in P_j} \omega(p) g_{pe} \quad \text{for all } e \in \{1, \dots, E\}, j \in \{1, \dots, N\}.$$

In two different ways the problem '3-partition', which is known to belong to be NP-complete, can be viewed as a special case of lab-feas. From these reductions, it follows that constructing a feasible solution for an instance of our problem is NP-complete in the number of shops as well as in the number of periods.

3-partition:

"Given a set A containing precisely $3m$ elements, where m is a strictly positive integer. Given also a bound $B \in \mathbb{Z}^+$ and for each element $a \in A$ a size $s(a) \in \mathbb{Z}^+$ such that $B/4 < s(a) < B/2$ and $\sum_{a \in A} s(a) = mB$.

Find the answer to the following question: can A be partitioned into m disjoint sets A_1, A_2, \dots, A_m such that

$$\sum_{a \in A_i} s(a) = B \quad \forall i \in \{1, \dots, m\} \quad ?"$$

Every instance of 3-partition can be reduced to an instance of lab-feas in the following two ways:

Reduction 1:

$P = A$, $\omega(p) = s(p)/B$ for all p in P , $E=T=1$, $N=T_{\max}=m$, and $\Delta(p) = [1]$ for all p in P .

Reduction 2:

$P = A$, $\omega(p) = s(p)/B$ for all p in P , $E=m$, $T=N=T_{\max}=1$, and for all p in P : $\delta(p)_{ef} = 1$ if $e=1$ and $f=E$, $\delta(p)_{ef} = 0$ in all other cases.

The reader should verify that the answer to an instance of 3-partition is 'yes' if and only if the answer to the corresponding instances of lab-feas is 'yes'. Combining this with the observation that the computation time required for verifying a yes answer is polynomial in the size of the input of the problem, we conclude that lab-feas is NP-complete.

Relaxation of the model

The afore mentioned complexity discussion does not imply that formulating our problem as an IP problem has been a purely academic exercise. By relaxing the integrality constraints to some (trivial) lower and upper bounds on the variables, the IP formulation reduces to a Linear Programming formulation. So, solving this LP-relaxation of an instance of our problem will yield a lower bound for the optimal value of the objective function. Sometimes this bound can be improved by adding valid inequalities to the model which are redundant for the IP formulation, but improves the solution of the LP-relaxation (see for the theoretical background of this approach for example Nemhauser & Wolsey, 1988).

As an illustration, we will extensively describe an example, inspired by the first example of Section 6.3. There are 5 different process types; for each of these process types an order requires 1 time-unit as process time. There are two periods, each of length 35. Furthermore, there are two jobshops, and the maximum number of technicians per shop, per period is 2. The total number of technicians available is 4. For simplicity we assume that the values of δ_{per} are determined (and not stochastic). The numbers of order-types (i.e. the values of δ_{per}) are given in the table below.

p	δ_{p11}	δ_{p22}	δ_{p12}
1	20	20	10
2	20	20	20
3	10	10	0
4	10	10	20
5	10	10	30

Since this example is very small it can be solved with a branch and bound algorithm. We will solve three versions of the model: (1) the IP-model, (2) the IP-model, U-variables relaxed, (3) the complete LP-relaxation. For each version we give the number of branches to indicate the size of the solution space and the number of pivots to indicate the relative speed to solve the model.

method	# branches	# pivots	max. idle time
(1)	33	867	7.5
(2)	3	55	0.0
(3)	-	29	0.0

A similar reasoning holds for subsets of process types. For example if the process types 1 and 2 are in the same job shop, then there should be at least $\lceil (20+20)/35 \rceil = 2$ technicians in that job shop in period 1 (as well as in period 2). So:

$$U_{ej} \geq 2 (B_{1j} + B_{2j} - 1)$$

Other inequalities here are:

$$\begin{aligned} U_{1j} + U_{2j} &\geq 3 (B_{1j} + B_{4j} - 1) \\ U_{1j} + U_{2j} &\geq 4 (B_{1j} + B_{4j} + B_{3j} - 2) \end{aligned}$$

These inequalities will be denoted as UB-inequalities, and in general we can formulate these inequalities as follows:

$$\left\lfloor \frac{\sum_{p \in S} \left(\sum_{g=e}^f \sum_{h=g}^f \delta_{pef} \right) \cdot w(p)}{f_e e_h} \right\rfloor (\sum_{p \in S} B_{pj} - |S| + 1) \leq \sum_{g=e}^f U_{gj}, \quad \forall e \leq f \in \{1 \dots |E|\}, \quad \forall S \subseteq P$$

In our example there are $2^5 - 1 = 31$ different subsets (of process types). Only 22 of them are feasible (capacity constraints). Furthermore for each subset there are 3 different subsets of consecutive periods. For each of these sequences of periods we can compute a lower bound on the number of technicians involved. So there can be 66 different inequalities for each of the two shops. However, only 17 of them are non-redundant. Adding the 34 inequalities to all three models we obtain:

method	# branches	# pivots	max. idle time
(1)	12	526	7.50
(2)	6	304	7.50
(3)	-	58	6.25

As can be seen the lower bounds are now improved. Because of this, the branch and bound algorithm should be more effective and therefore require less branches.

The UB-inequalities can be generalized by aggregating not only over consecutive periods, but also over (arbitrary) subsets of the job-shops. If we aggregate over all jobshops, then we obtain the earlier mentioned U-inequalities. Thus the U-inequalities are a special subclass of the UB-inequalities introduced here.

Since there are $(2^p - 1) * (n(n+1))/2$ different choices for S, e and f, the number of the additional inequalities presented here is tremendous. In general, this number is far too huge for the LP-relaxation in order to solve this extended formulation with an LP package. But in general we do not need all the inequalities and therefore we

can proceed as follows. We solve the LP-relaxation without any extra inequalities. Then we check the solution to determine violated inequalities. If there are violations, we add (some of) these inequalities to the model and solve it again, otherwise we stop. This process can be repeated several times. Each solution yields a lower bound on the optimal value of our objective function. We stop when we are content with the lower bound, or when there are no more violated inequalities.

6.5 Example: Instances for Tilburg laboratory

No	workstation	number of samples per day	minimum process time	maximum process time
1	Bulk chemical analyzer	445	0.5 min	1 min
2	Specific test analyzer	150	1.5 min	2.5 min
3	Specific test analyzer	195	1.5 min	2.5 min
4	Glucose and chloride analyzer	200	0.5 min	2 min
5	Bulk haematological analyzer	425	0.5 min	1.5 min
6	Small size haematological analyzer	95	1 min	2 min
7	Eye blood cell differential	55	2 min	5 min
8	ESR Erythrocytes Sedimentation Rate	125	1 min	2 min
9	Coagulation analyzer	102	2 min	4 min
10	Urine stick analyzer	102	2 min	3 min
11	Eye urine differential	43	1.5 min	2 min
12	Metabolic diseases screening tests	2	10 min	20 min
13	Metabolic diseases specific tests	2	200 min	300 min
14	Chromatography	90	4 min	8 min
15	Blood Gas analyzer	75	3 min	5 min
16	Manual techniques	25	5 min	60 min
17	Manual screening	150	2 min	3 min
18	Radio-immuno assays	70	5 min	10 min
19	Immuno assays	250	3 min	6 min
20	Reception area out-patient samples	600	1 min	1.5 min
21	Reception area in-patient samples	455	1 min	1.5 min

Table 6.12 Data used in the optimization model.

On the basis of the data given in table 6.12 job shops have been defined by the optimization model for several scenarios (several combinations of the total number of job shops and the maximum number of technicians per job shop). For each scenario we calculated a solution on the basis the maximum times for processing orders and an alternative solution on the basis of minimum times. For these calculations we used a heuristic (implemented in the DSS) which is described in appendix G. The maximum times are calculated by dividing the total maximum workload in terms of orders for a certain process type by the available capacity. The minimum time is an estimation of the required minimum capacity for

processing an order if the conditions are optimal (complete batches for processing). The maximum times indicate worst case situations. For the minimum times 20 technicians (U) were available, for the maximum times 24 technicians (U) are available. In table 6.13 the results are presented. On the basis of the calculations of our program we can conclude that the best solution is to assign 8 technicians to job shops. A maximum of 5 technicians appears to reduce the efficiency in terms of maximum idle time. It is optimal to divide the laboratory in 3 job shops when using maximum times. For the minimum times dividing the laboratory into 4 job shops is probably optimal. It appeared also that for both minimum and maximum times more solutions were feasible. The actual organization of the Tilburg laboratory is a mix of both solutions. Here three large workareas are distinguished (a workarea is a set of job shops) and one small workarea (one job shop). The last workarea functions as a buffer and provides staff to other areas if they are temporarily needed. It is also possible to check the optimality of this solution by following the approach discussed in the previous section. Taking the small amount of total idle time left in the optimal solution into account we can conclude that it does not make much sense to search for other solutions.

scenario		minimum process time 20 technicians			maximum process time 24 technicians		
# job shops	Max # technicians available	assigned # technicians to job shops / periods	max idle time (min)	total idle time (min)	assigned # technicians to job shops / periods	max idle time (min)	total idle time (min)
3	5	39	22.5	131	-	-	-
3	8	39	28.8	131	81	13	68
4	5	39	23.0	131	-	-	-
4	8	39	15.5	131	82	30	203
5	5	39	23.5	131	82	27	181
5	8	39	23.5	131	82	23	203

Table 6.13 Solution for scenarios for the optimization model.

7.1 Observations concerning the DSS

To experiment with rules for optimizing job shops discrete event simulation was used as a main technique. Apart from the flexibility of this technique, the advantage is that it directly represents the object system. For validation purposes (especially for nominal scenarios) this is desirable: the model simulates reality well. But there is also a disadvantage: long run times. Depending on the PC-type and the workload per computer minute only 1 to 5 days can be simulated. For single run simulations this is not a real problem, but when using the simulation module to conduct experiments (through the *experimenter* or the *scenario-analyzer*) it is. Our method to shorten the total time of an experiment was to use a binary search method (section 5.6.3). The simulation time was also shortened by stopping the simulation run when a certain percentage of samples late is exceeded. The simulation was stopped because the criterion is regarded to indicate that no stable solution is possible under the given circumstances. However, there are two problems with this criterion. The first problem is that some rules do not succeed at all. Although it seems likely that the rules are less effective than the average rule for which experiments could be carried out, a zero effectiveness under all circumstances is unlikely. The criterion that samples late should not exceed a certain threshold, is probably not able to accurately predict that a simulation is not stable. All simulations which are not stable are discontinued, but there is also a chance that this will happen in stable simulations. Therefore our criterion should be replaced. We are now working on a criterion that uses trend information on the number of samples late. In this method the number of samples late each day is compared to the mean number of samples late of the last x days. A second problem of the criterion is that it is cheap in a disputable way: it shortens the run time but it also causes a loss of information. We have seen that because of this the so-called *metamodelling* approach (Rotmans & Vrieze, 1990) could not always be applied. *Metamodelling* uses information from previous simulation experiments to predict the performance of other instances not simulated. A future research option would be that we initially allow the run times to be long in order to calibrate *metamodels*. An experimental design approach where only a fraction of all possible instances is simulated (see Kleijnen, 1992, and Rotmans & Vrieze, 1990) could then be used. Once accurate *metamodels* are available they can be used to predict optimal instances. The advantage of this approach is evident: the system actually 'learns' from previous experiments and simulation time is reduced.

With regard to the optimization model, there is also the problem of long run-times. For a successful DSS, these times have to be reduced. In this respect we could criticize our way of modelling the optimization problem. We opted to take a mathematical programming perspective. Although this approach has advantages we have to investigate a more direct approach to decrease run times. To diminish run times the use of a heuristic approach should be investigated further. One possible approach could be *Simulated Annealing* (see for example Press *et al.*, 1992).

The concept of our DSS is hierarchical: the optimization model concerns the entire laboratory, the simulation model the job shops. We did not choose to simulate the entire laboratory. Although it is technically feasible to extend the simulation of a

job shop to the whole laboratory, this would not make much sense for decision purposes (see our discussion on management functions). From a technical point of view there are problems with run times and the number of experiments. The run time is highly dependent on the number of events in the simulated system. In a simulation of the whole laboratory of the magnitude of the Tilburg laboratory the number of events will be enormous when compared to a simulation of a single job shop. Also the number of factors (rules, technicians, clustering of workstations) will be enormous. For making capacity planning decisions the practical value of simulating the whole laboratory seems therefore limited. On the other hand it could be of value for small laboratories. Then the simulation feature could be used at the laboratory level in the same way as it has been used at the shop's level.

What about the validity of our DSS? The validity of the components of the DSS have already been discussed. We indicated that active involvement of the technicians and laboratory management was important to gather data, check the face validity of results and discuss suggestions made by the system. How can the total concept of the DSS be validated? Does the concept underlying the DSS really give decision support? The practical involvement of the management of the laboratory is crucial, not only during the development of the concept of the DSS and testing the modules, but also in the future when the DSS is used in daily practice. The data used by the DSS should be checked frequently and if necessary adapted.

7.2 The hierarchical concept of the DSS

Our Decision Support System contains a planning function as well as a scenario function. Both functions can be used at an aggregate and a detailed level. At the aggregate level decisions are made concerning the structuring of the laboratory into job shops. The decisions on the detailed level relate to the operations within the job shops. As observed in chapter 6, especially the absence of a coupling between the aggregate level and the detailed level could be given further attention to improve the interaction between the aggregate and detailed levels. On the other hand one should realize that the optimizing model, the module at the aggregate level, is used for strategic/tactical problems whereas the simulation model, the module at the detailed level is used for tactical/operational problems. An example is the way 'workload' was used in the optimization model. Workload is an important measure of the capacity of the laboratory, but when making strategic decisions, almost none of the operational rules as used in chapter 5 are important.

7.3 Skills level and transfer time

In chapter 2 it was noted that an important goal of the laboratory is to provide services within the required due time. As the actual demand is up to a certain level unpredictable, slack capacity is always needed. This slack capacity concerns both equipment and staff. In our approach we decided to optimize on staff.

The costs of staff are higher than equipment (especially for laboratories with a moderate level of automation). Moreover, the labour market for laboratory technicians is scarce these days. Optimizing on staff has consequences for both the job shop level as well as the laboratory level. Within certain limits staff assignment should be flexible. Efficiency in the processing of samples can be further increased by using operational rules which make use of the planning freedom of orders. It is evident that our DSS makes the decision maker aware of the flexibility of staff assignment. But here is also a point for discussion. When the simulation module is evaluated by the management of the laboratory, the skills level of technicians should in principle be taken into account. Originally cross-trained technicians were assumed. During the project, when discussing the simulation model and its output with the management of the laboratory, this assumption had to be relaxed, although this change was not implemented. Scarcity of labour and budget limitations make it necessary to take into account the skills/experience level of technicians.

The simulation module does measure the transfer of technicians in departments which is an indirect indication of the skills level of technicians (see chapter 3). And as discussed in chapter 5, it is possible to relate the output (the variables transfer and workload) of the simulation runs to the extent of cross-training of technicians. However, this relation is often difficult to interpret and it is not possible to compare the results of simulations when a shop has a different mix of staff in each simulation experiment.

It appeared that sometimes the number of samples processed is very sensitive to changes in the transfer time. This means that the consequences of changes in transfer time are easy to detect, but that on the other hand for extrapolation purposes the magnitude of these changes may not be very reliable. This fact does not devalue the actual assumption of fully cross-trained technicians, because the transfer rate of labour in the job shop should give laboratory management insight in the required skills levels. But it is also evident that if technicians are not cross-trained the task of calibrating the simulation model to a specific job shop will be difficult.

7.4 Workload control

It has been pointed out that the distinction between aggregate and detailed planning is important. Both types of planning have been discussed in section 7.1. It appeared that both functions could be modelled well in our DSS and that good results were obtained.

Disappointing however is the effect of workload control which in our system is a mixture of aggregate and detailed control. Although workload had some positive effect, workload control in the involved laboratory was not as good as the effect of some detailed rules. One obvious reason could be the good quality of the tested detailed rules. Another reason could however be that as reported in the literature (chapter 3) workload control systems allow the planner to postpone due-times and therefore increase the planning freedom. As we did not allow the due-time to be violated, the effect of workload control will decrease. A third reason why workload control did not have much effect could be because of configuration characteristics. In the laboratory we studied, high priority was associated with specific workstations. The workstations, however, were almost never the bottleneck. The main problem is staff transfer. Just restricting the released workload of non-rush samples may not increase the transfer rate. This is confirmed by the outcomes of experiments with workload control (chapter 5): when workload control was applied the transfer rate was only slightly higher when compared to situations where it was not applied. Therefore it seems that other operational rules are more effective than workload control.

As discussed in chapter 3 a hospital laboratory job shop differs in several aspects from industrial job shops: the unpredictability of the exact required production, the possible arrival of rush requests, the externally constrained due-times and the fact that technicians can work on several workstations at the same time. These are the reasons that favourable rules in industrial situations such as the SPT-rule are not as effective as the FIFO-rules or variations on the FIFO rules, if necessary in combination with the HPJ-rule. Workload control was also not very effective. The special character of the hospital laboratory forbids the direct application of industrial planning methods.

7.5 Performance criteria for the laboratory

Performance criteria for the laboratory were discussed at several places in this study. In the optimization model the performance measure was the maximum idle time that could occur during a shift in one of the job shops. This performance measure was chosen after we implemented the first prototype of the optimizer. It seems to be the most realistic performance measure at the laboratory level. At the job shop level the number of jobs late and the maximum number of samples that can be processed were chosen as performance criterion. In our simulation model (specifically in the binary search procedure) we accepted maximally 10% jobs late (on any given simulated day). In fact this concession is a result of the simulation approach and the practical significance of this concession is (except for the throughput times of the highest service category) almost nil since the percentage of jobs late in the experiments was 1% at most. For the problem with throughput times of service category 1 (as discussed in chapter 5), we are dealing with an unsatisfactory but practically insignificant problem. The problem with this throughput occurs especially when the required service is uniformly low. Then, compared with the total workload, only few high priority samples arrive at the job shop. As there are only few of them, the variance in their throughput time will be

relatively high. When discussing simulation results this problem should obviously be addressed, because it may be important for the organization of the laboratory to know how high priority samples are handled in these situations.

7.6 Influence of the case-mix on the performance

The influence of the case-mix may be measured in two ways. The first way is by analysing the demand in terms of sample types and priority classes. This approach was followed in chapter 5 when the scenario-analyzer was discussed. It appeared that changes in workload can be modelled quite elegantly.

Another way to measure the influence of the case-mix would be to determine what type of patients (diagnosis, location) influence the demand for laboratory services. We performed two case studies, one for lung cancer patients (107 patients), another for obstetric patients (1251 patients). For both groups the exact demand for services with certain priority levels was studied. By means of a discriminant analysis a relation could be determined between on the one hand the diagnosis and the protocol and on the other hand the required laboratory production. Sometimes this relation is very direct. This is for example the case with IVF and bloodgas analyses (for obstetrics patients). Information about the relation between the case-mix on the one hand and laboratory production on the other hand is important for planning laboratory production. Further research seems promising here.

7.7 Future research

Future research should be directed to methods that shorten run times of simulations and 'learn' from other simulations. The metamodelling approach seems promising here. For the optimization model research should be directed to finding an efficient heuristic. By 'efficient' we mean that solutions can be generated by the system in a very short time (for example within some minutes). Research is also necessary to allow additional constraints to be put in the optimization model. One may think here for example of constraints concerning the skills level of staff and part-time workers. Of course it is also necessary to monitor the actual use of the DSS.

Our DSS supports several types of management decisions (see chapter 2). These decisions are made not only in the clinical laboratory but also in other types of laboratories (for example toxicology and bacteriology). The departments have often to fulfil the same requirements as mentioned above. The concept underlying our DSS should also be applicable to these departments. Using our DSS for these applications should, once it has passed the prototype stage, be very feasible. Maybe we could also consider the use of our DSS in other departments such as for example nursing departments, but here further research is required, because these departments can often influence directly the demand and they are directly coupled with other departments.

7.8 Conclusions

This study had the aim to investigate methods that can be used in a Decision Support System for capacity planning in the laboratory (of clinical chemistry and haematology). As such a DSS needs a planning and control model of the laboratory. We also studied the planning and control system of the laboratory. The planning and control system is described at two levels: the job shop and the laboratory. We first implemented the job shop level. Such a system contains many rules. Depending on the circumstances the effectiveness of these rules varies. Our DSS allows experimenting with these rules. In the discussed examples it appeared that because of the special character of the laboratory job shops rules which are often optimal in industrial job shops (such as the SPT-rule and workload control) are not very effective. It appeared that the FIFO-rule and variations on the FIFO rules possibly in combination with the HPJ rule perform best. The performance of the laboratory is dependent on the way that workstations are clustered into departments. Therefore we also developed a module with regard to this decision problem.

By way of example and for validation we tested and described some instances both at the shop and at the laboratory level. These examples not only demonstrate the functionalities of the DSS, but also allow the DSS to be tested by the management of the laboratory. Since the model at the detailed level describes the real situation directly at this level the values of different parameters had to be determined (calibration). The aggregate model was validated by the management (face validity).

It appeared that the DSS functions very well. Results seem to be reliable and are of interest to the management of the laboratory. The run times of the developed prototype appeared usually to be too long for practical use. Since the metamodelling approach allows us to simulate only part of the total number of possible rules and still predict the results of the combinations not considered this approach might be useful in keeping simulation time low. In our case the 10% threshold on samples late led to situations in which certain rules were not completely simulated. Since these rules are not known in advance in this case metamodelling could not be used.

The data needed for the DSS is available in the laboratory. It appeared that many data can be estimated very well by technicians. Once technicians are used to the DSS they will be able to feed the DSS with these data. Our DSS helps them to check the data by doing sensitivity analysis.

A DSS system has been designed and methods for capacity planning were implemented. Apart from the earlier mentioned time considerations the system has been evaluated positively by the laboratory management. The system will now be evaluated at several locations in the scope of the AIM programme OpenLabs.

GLOSSARY

actor: is defined by the set of actions and the set of communications it is able to perform. Abstracted is from the question which subjects are involved in the action.

action space: the set of actions which causes a transition in the state of the object system, that is a change from one state to another one. (chapter 2)

aggregation: different products or capacities are considered identical from a certain perspective. (chapter 3)

bank: conceptual store for facts concerning the object world (Dietz, 1992).

binary search procedure: For each rule half of the technically feasible (or some other arbitrary high) workload is simulated. After each simulated day the experimenter checks whether the number of samples late does not exceed the allowed maximum percentage of samples late. If the simulation for that day is successful, the simulator goes on to the next day. If each day (for the total number of days the experimenter wants to simulate) is successfully simulated, these data are averaged and stored. Then a new workload is taken which is between the workload of the last successful run and the nearest higher unsuccessful run. If a run does not succeed a workload is taken which is between the last run and the highest successful run. For each rule this process continues until the range in between a new workload must be chosen falls within the error bound. (chapter 5)

capacity: available equipment and human resources (chapter 1)

capacity utilization: that part of the total volume of capacity that is actually used during a certain planning period (e.g. a day, a week or a year). (chapter 1)

calibration: a technique for estimating system parameters in such a way that the results maximally resemble the object system's output (Kleijnen and Van Groenendaal, 1992). (chapter 1).

communication diagram: representation of a communication model in a network of actors, banks and channels (Dietz, 1993).

communication model: A model of the actors in a system and their communicative relationships (Dietz 1993).

configuration: a set of workstations. (chapter 2)

centralized staff control rule: rule stipulating that once staff is assigned, it can be reassigned during the production period. (chapter 3)

channel: conceptual store for the agenda, i.e. agreed future actions (Dietz, 1992).

cycle_time: the sum of *time_prepare_batch*, *time_process_batch* and *time_release_batch*. (chapter 5)

decision support system (DSS): an information system directed at supporting decision making of the management by focusing on *what-if* questions (chapter 1)

demand: volume, types of tests and required throughput times (chapter 1)

department: a part of the laboratory with the capability to plan its operational activities independently from other departments (chapter 2)

data: information items needed in transactions (Dietz, 1993).

data manipulation function: function to specify the problem by defining demand characteristics, the type of workstations, skill level of technicians and organizational characteristics. (Verbeek, 1991, Eiben, 1989) (chapter 2)

decomposition: the production system is divided into subsystems (laboratory management, laboratory departments, etcetera). Each subsystem performs one or more decision functions for which local goal variables have to be defined (Bertrand & Wortmann, 1981; Simon, 1981). (chapter 3)

decouple points: stocks and queues. They indicate the border of control and are used to decouple subsystems (for example departments). (chapter 3)

decentralized staff control rule: once staff is assigned, it cannot be reassigned during the production period or until the queue of the particular workstation is empty. (chapter 3)

discrete event simulation: "concerns the modelling of a system as it evolves over time by a representation in which the state variables change instantaneously at separate points in time" (Law & Kelton, 1991). (chapter 5)

deterministic model: optimization model (aims at finding an optimal clustering of workstations into job shops) where it is assumed that the future demand is predictable. The volume and service levels for the various process types are assumed to be also exactly known. (chapter 6)

effective laboratory: provides accurate information (results of one or more tests on samples placed in a defined context) in a form and within a time-span that is requested by physicians and hospital departments. (Chapter 1)

efficient laboratory: laboratory services are delivered at minimum costs. (Chapter 1)

effectiveness: the extent to which the due-times are realized. (chapter 2)

evaluation function: assesses the quality of each generated plan. (Verbeek, 1991, Eiben, 1989) (chapter 2)

event list: list of the moments on a timescale at which the state of the system changes e.g. the set up of a process, the completion of a process, the end of a working shift, etcetera. (chapter 5)

experimenter: module of our DSS which conducts experiments to test combinations of scheduling rules at the shop.

facta: information items created by transactions (Dietz, 1993).

First Come, First Served rule (FIFO): jobs are processed in the order of their arrival at the shop or the workstation. (chapter 3)

goal: a specific state that has to be realized. A goal always belongs to the state space. (chapter 2)

generation of a plan function: yields a plan for achieving one or more goals. (Verbeek, 1991, Eiben, 1989) (chapter 2)

horizontal decomposition: the production process is split up into a number of pooled, sequential or reciprocal subprocesses (Mintzberg, 1979). (chapter 3)

interdoctor variance: refers to the fact that physicians order different packages of hospital services for the same classes of patients (chapter 1).

information system: a system (an integrated whole of equipment, software, procedures, humans and data) which delivers information to someone who wants to make a decision. (chapter 1)

information model: the definition of the state space of the system (Dietz, 1992). (chapter 4).

interaction: relation between two actors if they are both involved in carrying through the same kind of transactions (Dietz, 1993).

interstriction: relation between two actors if one of them uses information items created by the other as data (Dietz 1993).

inter-reassignment time (IRT): time interval between subsequent re-assignments of staff. (chapter 2)

laboratory throughput time: the total time between the arrival of the sample at the laboratory reception and the communication of the test results. (chapter 2)

laboratory structure: the clustering of departments at one site. (chapter 2)

Least Slack per Operation rule (LSO): calculates the priority of orders as follows. First for each order the due-time tolerance (number of possible operations until due-time) is calculated by dividing the time span between the actual moment

and the due-time of the sample by the sum of processing times of each order. Then the due-time tolerance is divided by the number of remaining operations. The outcome of this procedure determines the priority (lower number have higher priority). (chapter 3)

Longest Queue (LNQ) rule: technicians are assigned to workstations or machines with the longest queue. (chapter 3)

Largest Processing Time (LPT) rule: technicians are assigned to the workstation with the largest processing time enqueued. (chapter 3)

Longest Waiting Time (LWT) rule: assigns labour to the workstation with the job in queue with the longest waiting time. (chapter 3)

management information system (mis): 'an integrated user-machine system for providing information to support operations, management, and decision-making functions in an organization. The system utilizes computer hardware and software; manual procedures; models for analysis, planning and control and decision making; and a database' (Davis & Olson, 1985, p.6). (chapter 1)

manipulation of the plan function: the option to define alternative plans. (Verbeek, 1991, Eiben, 1989) (chapter 2)

model: a set of code representing the object system. (chapter 4)

metamodel: a regression model on a set of simulations. (chapter 5)

nominal scenario: the existing situation. (chapter 5)

operational planning: refers to what Anthony & Young (1988) call task control, that is the process of assuring that the tasks are carried out effectively and efficiently. (chapter 2)

operational risk: the risk that orders are not produced according the goals of the operational planner. (chapter 2)

optimization model: this model aims at finding an optimal clustering of workstations into job shops. (chapter 6)

production system: the integrated whole of instruments, logistics and personnel. (chapter 1)

plan: a set of selected actions to obtain a goal. (chapter 2)

plan and scenario management function: allows the decision-maker to specify any configuration and any scenario. (Verbeek, 1991, Eiben, 1989) (chapter 2)

production control/production control system: 'all activities which concern the adjustment in time and place of materials and capacities to each other, so that goals with regard to the product and to the production process are fulfilled'. Van Rijn (1986, p.41). (chapter 3)

process type: consists of a group of tests that can be performed on a certain machine and that has the same processing times. (chapter 6)

planning freedom: the degree to which the laboratory is free to process an order at a later period than the period of arrival of orders at the shop. (chapter 6)

portfolio-approach (applied to the optimization problem): a shop is seen as a portfolio of processes, where to each shop processes are assigned in such a way that the variance of the workload of the shop is less than the sum of variances of the processes allocated to the shops. In essence the portfolio-approach tries to allocate processes to shops that are negatively co-variated. (chapter 6)

queued samples: samples which are processed as soon as capacity becomes available. (chapter 3)

representation of the plan function: function to provide the decision maker a specification of workstations, departments, operational rules and the number of available technicians. (Verbeek, 1991, Eiben, 1989) (chapter 2)

required capacity: the volume of capacity needed to realize the demanded production. (chapter 1)

scenario-analyzer: determines the performance characteristics of a shop configuration and a given set of operational rules under varying workload conditions.

slack capacity: redundant capacity (chapter 1)

shop throughput time: the sum of the waiting time (the time the sample is waiting before being processed) and processing time. Or in symbols: $F = W_q + W_p$. (chapter 2)

strategic planning: the process of deciding on the goals of the organization and on the broad strategies that are followed to attain them ('what' and 'how' questions) (chapter 2)

state space: the set of all states the object system (production system) can adopt. The definition of the state space depends upon the problem which is analyzed. (chapter 2)

selection function: generates proposals for decision-making. (Verbeek, 1991, Eiben, 1989). (chapter 2)

sequencing: the planner determines some order in which the samples will be processed. (chapter 3)

scheduling: sequencing but also including time-tabling. (French, 1982) (chapter 3)

Shortest Processing Time rule (SPT): gives priority to orders with the shortest processing time. (chapter 3)

staff control rules: indicate who is in control of labour assignment and what kind of assignments and reassignments are allowed. (chapter 3)

staff assignment rules: deal with the assignment of personnel to separate workstations within job shops. (chapter 3)

state space of the system: the set of states that can and are allowed to occur. (chapter 4)

stocked samples: samples which are buffered and which will be released by a decision. (chapter 3)

state of the workstation: a specific combination of (a) the phase of the workstation, (b) the presence or absence of a technician, (c) the activity status of the phase which is either waiting, working, being interrupted or suspended, and (d) the queue of samples belonging to that workstation. (chapter 5)

transaction processing system: a system in which data about the transformation system of the organization are stored and which processes information through knowledge of the context. (chapter 1)

total throughput time of a test request: the laboratory throughput time but also the time for ordering tests, collecting samples and interpreting the test results. (chapter 2)

tactical planning/management control: is situated between strategic planning and operational planning. It is directed at 'the implementation of strategies and the attainment of goals'. It has to assure 'that the organization implements its strategies effectively and efficiently' (Anthony & Young, 1988). (chapter 2)

transition space: all possible changes (transitions) from one state to another state. (chapter 2)

tactical risk: the risk of not choosing a configuration which would have had a better performance under certain demand characteristics (the actual scenario). (chapter 2)

vertical decomposition: the production system is decomposed into hierarchical ordered subsystems. (chapter 3)

workstation: a site where a technician can perform a certain process type. (chapters 5 and 6)

workload control: a technique that uses information about actual workload, forecasted demand and available capacity to release and process orders. It considers the job shop as a black box. Forecast information is used to adapt the release rate of stocked samples to the expected arrival rate of rush orders. When this arrival rate is expected to be high, the release rate is low, and vice versa. This type of workload control has the advantage that a constant and high level of capacity (labour) utilization can be realized even if rush orders are allowed. It is thus a manner of controlling the arrival of samples at workstation queues (chapter 3). It is an aggregate technique controlling the release of samples to workstations queues. (chapter 5).

- Aerts, A.T.M., K.M. van Hee, *Modelleren met een Functioneel Datamodel*, Informatie jaargang 31 nr. 12, p.901 - 1016, 1989.
- Ament, A., *Optimaal gebruik van diagnostische tests*, Maastricht, 1989 (Thesis).
- Anthony, R.N., *The Management Control Function*, Harvard Business School Press, Boston, 1965/1988.
- Anthony, R.N., D.W. Young, *Management Control in Nonprofit Organizations*, Irwin, Homewood, 1988.
- Amey, L.R., *Budget planning and control systems*, Pitman, London, 1979.
- Axsäter, S., *On the design of the aggregate model in a hierarchical production planning system*. In: Engineering and Process Economics, 4(1979) 89-97, Elsevier Scientific Publishing Company, Amsterdam.
- Axsäter, S., Jönsson, H., *Aggregation and disaggregation in hierarchical production planning*. In: European Journal of Operation Research 17 (1984) 338-350, North-Holland, Amsterdam.
- Bennington, J.L., Böer, G.B., Louvau, G.E., Westlake, G.E., *Management and Cost Control Techniques for the Clinical Laboratory*, University Park Press, Baltimore, 1977.
- Bertrand, J.W.M., Wijngaard, J.C., *The Structuring of Production Control Systems*, The International Journal of Operations and Production Management, 1986, vol. 6, blz. 5-20.
- Bertrand, J.W.M., Wortmann, J.C., *Production control and information systems for component-manufacturing shops*, Elsevier Scientific Publishing Company, Amsterdam, 1981.
- Bertrand, J.W.M., Wortmann, J.C., Wijngaard, J., *Productiebeheersing en material management*, Leiden/Antwerpen, 1990.
- Blom, J.A., *The SIMPLEXYS Experiment*, Eindhoven 1990.
- Bratko, I., *Prolog: Programming for Artificial Intelligence*, 2nd ed., Wokingham, 1990.
- Broughton, P.M.G., Hogan, T.C., *A new approach to the costing of clinical laboratory tests*, Ann Clin Biochem 1981; 18: 330-342.
- Classificatie van ziekten 1980; deel 1:systematisch*, tweede druk, SIG Informatiecentrum voor de gezondheidszorg, 1988.

- Clymer, J.R., *Systems Analysis using simulation and Markov-models*, Prentice-Hall, Englewood Cliffs, 1990, p. 160-165.
- Connelly, D.P., Willard, K.E., *Monte Carlo Simulation and the Clinical Laboratory*. In: Arch Pathol Lab Med, vol 113, July 1989.
- Davis, G.B., Olson, M.H., *Management Information Systems*, New York, 1985.
- Dietz, J.L.G., *A Communication Oriented Approach to Conceptual Modelling of Information Systems*. In: Steinholtz, B., Sølvsberg, A., Berman, L., (Eds.), *Advanced Information Systems Engineering, Second Nordic Conference CAISE '90*, Stockhol, Sweden, May 8-10, 1990, Proceedings.
- Dietz, J.L.G., *Subject-Oriented Modelling of Open Active Systems*. in: *IFIP Transactions A-4 Information System Concepts*, North-Holland, 1992.
- Dietz, J.L.G., *Business Modelling for Business Redesign*, Research Memorandum, University of Limburg, 1993.
- Eiben, A.E., *Modelling planning problems*. In: *Proceedings of 2nd Symposium on Mathematical Fundamentals of Database Systems*, Visegrád, Hungary, 1989, Springer Verlag.
- Eiben, A.E., *A Method for Designing Decision Support Systems for Operational Planning*, Eindhoven, 1991.
- Enkin, M., Keirse, M.J.N.C., Chalmers, I., *A Guide to Effective Care in Pregnancy and Childbirth*, Oxford University Press, Oxford, 1989.
- Flamholtz, E., *Behavioral Aspects of Accounting/Control Systems*. In: *Organizational Behaviour*, ed. S. Kerr, Columbus, Ohio, 1979: Grid, p. 290.
- Flynn, B.B., F.R., Jacobs. *An Experimental Comparison of Cellular (Group Technology) Layout with Process Layout*, *Decision Sciences*, Vol. 18, pp. 562-563, 1987.
- Flynn, B.B., F.R., Jacobs. *A simulation comparison of group technology with traditional job shop manufacturing*, *International Journal of Production Research*, Vol., 24, No. 5, pp. 1171-1192, 1986.
- French, S., *Sequencing and Scheduling*, Ellis Horwood, Chichester, 1983.
- Fryer, J.S., *Effects of shop size and labour flexibility in labour and machine limited production systems*, *Management Science*, Vol. 21, No. 5, 1975.

- Frowein, C., F.J. Hindriks, A. Bosman, H. Kamps, W. van der Slik, *Kostprijberekening van klinisch-chemisch onderzoek met behulp van een algemeen toepasbaar kostenstructuurmodel*. Tijdschrift voor Nederlandse Vereniging voor Klinische Chemie, 1985, nr. 10; 181-188.
- Garey, M.R., Johnson, D.S., *Computers and intractability: a guide to the theory of NP-completeness*, New York, 1979.
- Goodrich, R.G. and G.R. Hastings, *St. Luke's Hospital reaps benefits by using product line management*, Modern Health Care, Feb. 15, 1985: 157-158.
- Griner, P.F., Mayewski, R.J., Mushlin, A.I., Greenland, P., *Selection and interpretation of diagnostic tests and procedures*, Ann. Intern. Med, 94: pp. 553-600, 1981.
- Haddock, J., Seshadri, N., Srivatsan, V.R., *A Decision Support System for Simulation Modelling*. In: Journal of Manufacturing Systems, Volume 10, nr. 6, 1991.
- Hax, A.C., Candea, D., *Production And Inventory Management*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.
- Hasman, A., Merode, G.G. van, *Management Informatiesystemen in Instellingen*. In: Beleid en beheer in de gezondheidszorg, Maarse, J.A.M., Mur-Veeman, I.M. (ed.), p. 362-384.
- Hax, A.C., Candea, D., *Production and Inventory Management*, Prentice-Hall, Englewood Cliffs, 1984.
- Hee, K.M. van, *Informatiesystemen en beslissingsondersteuning*. In: Informatie jaargang 27, nr. 11 pp. 929-1032, november 1985.
- Hee, K.M., Lapinsky, A., *OR and AI Approaches to Decision Support Systems*. In: Decision Support Systems 4 (1988) 447-459, North-Holland, Amsterdam.
- Hornbrook, M. and J. Rafferty, *The economics of hospital reimbursement*. In Advances in Health Economics and Health Services Research, JAI Press, Vol. 3., 1982, page 84.
- Huang, P.Y., L.J. Moore, R.S. Russell, *Workload vs Scheduling Policies in a Dual-Resource Constrained Job Shop*, Computers and Operations Research and their applications to problems of world, Oxford, vol. 11, No. 1, pp. 37-47, 1984.
- Huang, P.Y., Rees, L.P., B.W. Taylor, *A simulation analysis of the Japanese Just-In-Time technique (with KANBANS) for a multiline, multistage Production System*, Decision Sciences, pp. 326-344, 1983.

Ibe, O.C., Trivedi, K.S., *Stochastic Petri Net Analysis of Finite-Population Vacation Queueing Systems*. In: *Queueing Systems 8* (1991) p. 111-128.

Jaarboek verloskunde 1991, SIG Zorginformatie, Utrecht, 1992.

Jackson, J.R., *Jobshop-like Queueing Systems*, *Management Science*, 10, p. 131-142, 1963.

Janse, T.A.H.M., Kateman, G., *Enhancement of Performance of Analytical Laboratory*, *Analytica Chimica Acta*, 150 (1983) 219-231.

Jain, R., *The Art of Computer Systems Performance Analysis*, John Wiley & Sons, Chichester, 1991.

Kamin, S.N., *Programming Languages: An Interpreter-Based Approach*, Addison-Wesley, Reading, 1990.

Karni, E., *State-dependent Preferences*. In: Eatwell, J., Milgate, M., Newman, P. (Eds.), *Utility and Probability*, London, 1987, p. 242-247.

Kleijnen, J.P.C., Van Groenendaal, W., *Simulation, A statistical perspective*, John Wiley & Sons, Chichester, 1992.

Kleijnen, J.P.C., Van Ham, G., Rothmans, J., *Techniques for sensitivity analysis of simulation models: A case study of the CO₂ greenhouse effect*, *Simulation*, june 58:6, p.410-417, 1992.

Kluijdt, I., Zaat, J.O.M., Van der Velden, J., Schellevis, F.G., *Voor een prikje? Het gebruikt van klinisch-chemische, hematologische en serologisch bepalingen door de huisarts*. In: *Huisarts en wetenschap* 1991; 34(2), p.67- 71.

Law, A.M., Kelton, W.D., *Simulation Modelling and Analysis*, 2nd edition, McGraw-Hill, New York, 1991.

Lee, L.C., *A Study of system characteristics in a manufacturing cell*, *International Journal of Production Research*, vol. 23, No. 6, pp. 1101-1114, 1985.

Leyten, J.F., Geer, F. van der, Scholten, M.N.M., Goldschmidt, H.M.J., *The costing of tests in a laboratory for clinical chemistry and haematology*, *Ann Clin Biochem* 1984, 21: 109-115.

Little, J.D.C., *A proof for the Queueing Formula: $L=\lambda W$* , *Operations Research.*, 9, p.383-387, 1961.

Lundberg, G.D., *Acting on significant laboratory results editorial*, *JAMA* 245:1762-1763, 1981.

- Lundberg, G.D., *Using the Clinical Laboratory in Medical Clinical Decision-Making*, American Society of Clinical Pathologists Press, Chicago, 1983.
- Maarse, J.A.M., *Politieke besluitvorming in de gezondheidszorg*. In: *Beleid en beheer in de gezondheidszorg*, Maarse, J.A.M., Mur-Veeman, I.M. (ed.), p. 191-214.
- Machina, M.J., Rothschild, M., *Risk*. In: Eatwell, J., Milgate, M., Newman, P. (Eds.), *Utility and Probability*, London, 1987, p. 227-239.
- MacStravic, R.S., *Product-line administration in hospitals*, Health Care Management Review, Spring, 1986, 11(2), 35-43.
- Merchant, K.A., *Control in business organizations*, Boston, 1985.
- Mintzberg, H., *The Structuring of Organizations*, Prentice-Hall, Englewood Cliffs, 1979.
- Molloy, M.K., *Structurally Bounded Stochastic Petri Nets*. In: *Proceedings of International Workshop in Petri Nets and Performance Models*, Madison, Wisconsin, August 24-26, 1987, p. 156-163.
- Morgan, M.M., M. Henrion, *Uncertainty, A guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*, Cambridge University Press, Cambridge, 1990.
- Morris, J.S., R.J. Tersine, *A simulation Analysis of Factors influencing the Attractiveness of Group Technology Cellular Layouts*, Management Science, Vol. 36, Nr. 12, 1990.
- Mosier, C.T., D.A. Elvers, D. Kelly, *Analysis of group technology scheduling heuristics*, International Journal of Production Research, Vol. 22, No. 5, pp. 857-875.
- Murphy, J., Henry, J.B., *Effective utilization of clinical laboratories*, Hum. Pathol., 9: p. 625-633, 1978.
- Müskens, P.J.W.M., *De Analyse ten behoeve van bewaking van dynamische processen*, Nijmegen, 1978.
- Nederstigt, P.F.P.M., *Diagnosis Related Groups: Een patiënt-georiënteerd kosteninformatiesysteem*, Nationaal Ziekenhuisinstituut, Utrecht, 1985.
- Nemhauser, G.L., Wolsey, L.A., *Integer and Combinatorial Optimization*, New York, Wiley, 1988.
- Neter, J., Wasserman, W., Kutner, M., *Applied Linear Statistical Models*, 3rd edition, Irwin, 1990.

- Nackel, J. and I. Kues, *Product Line Management: Systems and Strategies*, Hospital and Health Services Administration, March/April, 1986, p. 109-123.
- Neumann, B. et al., *Financial Management*, Nationals Health Publishing, Maryland, 1986, page 469.
- Paccaud, F., Schenker, L. (Eds.), *D.R.G. (Diagnosis Related Groups): Perspective d'utilitisation*, Lyon, 1989.
- Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., *Numerical Recipes in C*, 2nd edition, Cambridge, 1992.
- Pullen, R.D., *A Survey of Cellular Manufacturing Cells*, The Production Engineer, September, 1976.
- Render, B., Stair, R.M., *Quantitative Analysis for Management*, Allyn & Bacon, Boston, 1988.
- Rotmans, J., Vrieze, O.J., *Metamodelling and experimental design: Case study of the greenhouse effect*, European Journal of Operational Research 47 (1990) 317-329.
- Rubinstein, R.Y., *Monte Carlo Optimization, Simulation and Sensitivity of Queueing Networks*, Wiley & Sons, New York, 1986.
- Russel, R.S., P.Y. Huang, Y. Leu, *A study of Labour Allocation Strategies in Cellular Manufacturing*, Decision Sciences, Vol., 22, pp. 594-611.
- Schaepman-Van Geuns, E.J., Treffers, P.E., *Praktische verloskunde*, Bohn Scheltema & Holkema, Utrecht/Antwerpen, 1989.
- Simon, H.A., *The Sciences of the Artificial*, 2nd. edition, The MIT Press, Cambridge, 1981.
- Speicher, C.E., Smith, J.W., *Choosing Effective Laboratory Tests*, Philadelphia, 1983.
- Suresh, N.C., *Partitioning Work Centers for Group Technology: Insights from an Analytical Model*, Decision Sciences, Vol., 22, p. 772-791, 1991.
- Tapiero, C.S., *Applied stochastic models and control in management*, North-Holland, Amsterdam, 1988.
- Vandeginste, B.G.M., *Digital Simulation of a Laboratory for Structural Analysis*, Nijmegen, 1980.
- Van Merode, G.G., Winter, J., *Management Control in Health Care Organizations*. In: Health Systems - The Challenge of Change, Proceedings of the

Fifth International Conference on System Science in Health Care, Omnipress, Prague, 1992, pp. 795-798.

Vannelli, A., K.R. Kumar, *A method for finding minimal bottle-neck cells for grouping part-machine families*, International Journal of Production Research, Vol. 24, No. 2, pp. 387-400, 1986.

Van Rijn, T.M.J., *Producteren door Informeren: Informatie-eisen voor verschillende produktiesituaties*, Kluwer, Deventer, 1986.

Van Vliet, R., *Hospital Utilization, Performance, Measures and Health Status*, Thesis, Rotterdam, 1988.

Vellekoop, A.H., *Decision support for the analysis of economic risk with an application in the dredging industry*, Rotterdam, 1989.

Verbeek, P.J., *Learning about Decision Support Systems*, Amsterdam.

Wagner, H.M., *Principles of Operations Research*, Englewood Cliffs, 1975.

Warner, D.W., Holloway, D.C., *Decision Making and Control for Health Administration*, Ann Arbor, 1978.

Weeks, J.K., J.S. Fryer, *A simulation study of operating policies in a hypothetical dual-constrained job shop*, Management Science, vol. 22, No. 12, 1976.

Wertman, B.G., Sostrin, S.V., Pavlova, Z., Lundberg, G.D., *Why Physicians Order Laboratory Tests*. In: Lundberg, G.D., *Using the Clinical Laboratory in Medical Decision Making*, Chicago, 1983.

Williams, H.P., *Model Building in Mathematical Programming*, 3rd edition, John Wiley & Sons, Chichester, 1990.

Winkel, P., *Operational Research and Cost Containment: A General Mathematical model of a Workstation*, Clinical Chemistry, Vol. 30, No. 11, 1984.

Wolff, R.W., *Stochastic Modelling and the Theory of Queues*, Englewood Cliffs, 1989.

Zaat, J.O.M., Van Eijk, J.T.M., Bonte, H.A., *Mag het ook een testje minder? De invloed van een bepaling van het aanvraagformulier voor laboratoriumonderzoek*. In: Huisarts en wetenschap, 1991; 34(2), p.72-77.

Zaat, J.O.M., Van Eijk, J.T.M., *Praktijk- en artskenmerken en het aanvragen van laboratoriumonderzoek*. In: Huisarts en wetenschap, 1991, 34(2), p.78-81.

APPENDICES

Appendix A
Appendix B
Appendix C
Appendix D
Appendix E
Appendix F
Appendix G
Appendix H
Appendix I
Appendix J
Appendix K
Appendix L
Appendix M
Appendix N
Appendix O
Appendix P
Appendix Q
Appendix R
Appendix S
Appendix T
Appendix U
Appendix V
Appendix W
Appendix X
Appendix Y
Appendix Z

APPENDIX A: list of symbols

Chapter 2

λ_L	inflow of samples in the laboratory where L stands for the <i>laboratory</i>
λ_i	indicates the number of samples arriving at shop i during a certain time period (e.g. one hour)
F	denotes the shop flow (or throughput) time
W_q	the waiting time of a sample at a shop
W_p	the processing time of a sample at a shop or $1/\mu_p$
μ_p	the processing capacity in terms of the number of samples which can be processed during one time unit
μ_a	actual maximal production at the bloodgas workstation
λ_r	mean inflow of bloodgas samples per hour
γ	vacation transition rate per hour
λ	arrival rate of bloodgas sample per hour
IRT	inter-reassignment time
σ_F^2	variance of throughput times
S_E	the set of all production orders during period E
i	production order
E	production period
Eff_E	Effectiveness of laboratory job shop during E
\bar{F}	mean flow time
σ_F^2	variance of flow time
\bar{L}	mean lateness
σ_L^2	variance of lateness
P	proportion of jobs late
C_i	completion time of order i
r_i	the release time of order i
W_i	the time a job has to wait in the shop
P_i	the time a job is actually processed
T	the total labour transfer between workstations at a job shop
c_1, c_2	performance criteria (in example on scenario-analysis)
p_1, p_2	configuration 1 and 2 (in example on scenario-analysis)
w_1, w_2	scenario 1 and 2 (in example on scenario-analysis)

Chapter 5

<i>time_set_up</i>	time needed to set up the workstation;
<i>time_prepare_batch</i>	time needed for preparing the batch;
<i>time_process_batch</i>	time needed to process the batch;
<i>time_release_batch</i>	time needed to release the batch;
<i>time_shut_down</i>	time needed to shut down the workstation;
<i>batch_size</i>	maximum number of samples in the processing batch.
<i>cycle_time</i>	sum of <i>time_prepare_batch</i> , <i>time_process_batch</i> and <i>time_release_batch</i>
<i>technician_time_w</i>	the time needed for a technician to process a sample at a workstation
$V(1,w)$	number of samples in the shop queue for workstation w
$V(2,w)$	number of samples enqueued at workstation w
$V(3,w)$	number of rush samples to be expected for workstation w during the present staff assignment period e
$V(4,w)$	indicates whether there is enough workload at workstation w for processing
$V(5,w)$	mean waiting time of samples in the shop queue for workstation w
$V(6,w)$	mean waiting time of samples in workstation w queue
$V(7,w)$	smallest due time in workstation w queue (expressed as $1/(\text{smallest due time})$)
$V(8,w)$	number of samples with due-time in specified priority time range in workstation w queue
$V(9,w)$	longest waiting time of a sample in workstation w queue
$V(10,w)$	indicates whether the next phase of the workstation needs the presence of a technician
$V(11,w)$	equal to $V(4,w)$
θ_i	weight to each $V(i,w)$. If $V(i,w)$ is not taken into account, then θ_i is equal to 0, otherwise it is 1.
b_{iw}	multiplication factor for $V(i,w)$ in order to compare weights among the workstations
c_i	balancing factor to compare the weights of $V(i,w)$ among each other
<i>mn_out</i>	the mean number of processed and authorised sample per day during the simulation;
<i>sd_out</i>	the standard deviation around <i>mn_out</i> ;
<i>mn_left</i>	number of samples not processed at the end of the day;
<i>sd_left</i>	standard deviation around <i>mn_left</i> ;
<i>mn_late</i>	number of samples not authorised before due-time;
<i>sd_late</i>	standard deviation around <i>mn_late</i> ;
<i>mn_transfer</i>	number of times that technicians are transferred between workstations;
<i>sd_transfer</i>	standard deviation around <i>mn_transfer</i> ;
<i>mn_working</i>	time that technicians have worked at the workstation (percentage of total available time);
<i>sd_working</i>	standard deviation around <i>mn_working</i> ;

$util_w$	utilization degree of workstation w ;
$operator_w$	utilization degree of technicians at workstation w ;
$throughput_time_p$	mean throughput time for each priority class p .
$capacity$	the available capacity
$expected_rush_samples$	the expected number of samples arriving with a due time within the next staff assignment period
$previous_released_workload$	the workload released in previous periods and not yet processed;
$rush_demand_fraction_w$	the expected rush demand for the next staff assignment period for workstation w ;
$O_{j,e}$	the number of technicians at a certain job shop j in technician assignment period e ;
t_w	which is the time needed for a technician to process a sample at w ;
W	workload expressed in processing time;
W_o	the total time a technician is available for working;
W_u	workload for rush orders;
W_q	workload due to samples from the last technician assignment period;

Chapter 6

L_{1b}	the minimum amount of work expressed in technician time that has to be done at a job shop in a single period e ;
L_{1up}	the maximum amount of work that is allowed to be performed at a job shop in one period ;
$k(i)$	the arrival time of order i ;
$d(i)$	the due date (due time) of order i ;
$w(p)$	the processing time for an order of process type p ;
$J(p)$	set of job shops that may process orders of type p ;
$p(i)$	process type of order i ;
$I(p)$	set of all orders of type p ;
F	set of pairs (p,q) of process types p and q which are not allowed to be assigned to the same job shop (forbidden combinations);
J	set of all job shops;
E	the number of periods e in one shift;
L_{je}	the total time needed within a period e to process the jobs in job shop j ;
Z	upper bound of the idle time of all technicians in one period;
U_{je}	the available number of technicians in job shop j in period e ;
B_{pj}	1 if a process type p is assigned to job shop j ; 0 otherwise;
A_{ijt}	1 if order i is scheduled to be processed at time t at job shop j ; 0 otherwise;
i,j	index of job shops $1 \dots N$
e,f,g,h	index of periods $1 \dots E $
p,q,r	index of process types $1 \dots P $
F	set of pairs (p,q) of process types $p, q \in P$ which cannot occur in the same jobshop (' forbidden combinations ')
$J(p)$	set of jobshop-indices of those job shops equipped to handle process type p
U	the total number of available technicians in the laboratory in each period
$w_1(p)$	total processing time for any order of process type p
$w_2(p)$	time required by a technician to handle any order of process type p
e_b, e_e	begin time, respectively end time of period e
δ_{pef}	the number of orders of process type p with release-time e_b and due time f_e
B_{pj}	has value 1 if and only if process type p has been assigned to jobshop j
$C_{r,qj}$	has value 1 if and only if process type p and process type q have been assigned to jobshop j
N_{pje}	the number of orders of process type p in jobshop j during period e
U_{ej}	the number of technicians required in jobshop j during period e
Z	maximal idle time of the technicians in one jobshop in a period
S_j	index set of process types in shop j

$X_e(S_j)$ stochast denoting the total load of a shop j in period e
 $\rho_{ij}, \sigma_i, \sigma_j$ denotes covariance between processes p and q
maxload: the maximum number of orders of type p in period e
 $\text{cor}(p, q):$ time penalty for each period for assigning processes p and q to
 the same job shop

APPENDIX B: DATA OBJECTS (CHAPTER 4, 5 AND 6) AND DATA FILES

In chapters 4, 5 and 6 we mentioned at several occasions the data objects we use. The variables in these data objects are contained in several files. These files are described in the following appendices. Here we indicate in which inputfiles the variables of the data objects can be found.

data object: shop_configuration

File: Shops.d, variables 1-13

File: Workst.d, variables 1-9, 13-35.

data object: workload_mix

File: Shops.d, variables 3, 5, 14.

File: Workst.d, variables 10, 11, 12.

data object: rules

File: Shops.d, variables 11, 12, 13.

File: Workst.d, variables 8, 9.

data object: workload_mix_laboratory

File: Process.d, variables 1,2,5,7-11.

data object: Process_lab

File: Process.d, variables 1,2,3,4,6.

DATA FILES FOR THE MODULE OF THE DSS

The simulation module uses the following input files:

CONTROL.D,
SHOPS.D, and
WORKST.D.

The lay out of the latter two files, i.e., the description of the records, is contained in OVERALL.D. This file is used by the interface module for the necessary maintenance of these files. For documentation reasons and providing text for the help function in the interface we provided the facility of describing the meaning and working of the data fields in OVERALL.D. In this description the symbol > followed by a number n denotes that n lines of (help) text follow. Some of the fields are multi-valued. This is denoted by the symbol # followed by the dimension. For convenience of the reader the record descriptions are displayed together with the inputfiles.

File CONTROL.D consists of single-valued parameters which are globally valid; either they concern the configuration of the laboratory or they influence the

APPENDIX C: CONTROL.D

CONTROL PARAMETERS FOR SIMULATION EXPERIMENTS

simulation time

number of days to be simulated

100 0 10000 1

actual shop

The ID-number of the job shop to be simulated

1 1 10000 1

day shifts

number of shifts during the day

1 1 10 1

day length

length of a working day (in hours)

9 0 24 1

day start

time at which the working day starts (in hour of the day)

8 0 24 1

overstay time

time at the end of a shift a technician cannot be reallocated (in minutes)

1 1 60 1

startup time

time needed for a technician to start an operation (in minutes)

1 0 60 1

random stream

seed values for the various random streams

0 0 111111 1

workload recording

number of days on which recorded workload is kept in memory

2 0 100 1

priority interval length

this value times the time needed to process a batch defines the time for recording samples in shop parameter 8

2 1 10 1

meta model

(switch) 1: preparation is made for multiruns; 0: no preparation

1 0 1 2

accuracy
percentage of maximal workload allowed to differ from real outcome
2 1 100 2

no priority
time in minutes at the end of the last day shift having no urgent samples
45 0 500 2

stopping device
value 0: simulation does not stop until the end of simulation time; value 1: stop after a job late; value 2: stop after jobs late at the end of the day;
0 0 2 1

samples late
percentage of the day workload which is allowed to be late at
10 0 100 2

output
output is either shift oriented (0), day oriented (1) or simulation oriented (2)
0 0 2 1

mean info
switch: 1 if the mean of the shop characteristics are needed; 0 if not
0 0 1 1

maximum interactions
maximum number of positive weights
2 0 20 1

maximum due date
used in the rule (shop parameter 18) for determining the number of samples with due date at most the time given here (minutes)
0 0 600 1

interval samples late
time interval in minutes in which samples late is registered at most once (0 if registration at output time)
0 0 6000 1

end of data

APPENDIX D: SHOPS.D

This file contains the data with regard to the job shops. Decisions with regard to the assignment of the operators to job shops and to workstations, and sequencing of samples are being made on the basis of 20 shop parameters. These parameters are evaluated during the simulation (evaluation at the start of each shift and/or at a reassignment request); some of them are workstation dependent; others are job shop dependent. The shop parameters are:

- 1: number of samples in the job shop queue for the workstation under study;
- 2: number of samples in the workstation queue;
- 3: number of samples to be expected during the day for the job shop;
- 4: =1 if sufficient number of samples are waiting at the workstation; else 0;
- 5: average queuing time of the samples in the job shop queue;
- 6: average queuing time of the samples in the workstation queue;
- 7: reciproke of the smallest due date of a sample on the workstation;
- 8: number of samples in the workstation queue with due date at most two times the cyletime;
- 9: longest queuing time of the samples in the workstation queue;
- 10: =1 if the workstation under study is interrupted while batch processing; else 0;
- 11..14: copies of 1..4. 14 is in the text indicated as V_{11} ;
- 15..17: free;
- 18: number of samples with due date at most the time given in control parameter 'maximum due date';
- 19: =1 if operator is present at the workstation under study; otherwise 0;
- 20: number of batches at the workstation under study (0 if operator is absent).

In the figure on the next page the record structure is presented.

(Field#, Name, Type)

1 **job shop name** text

2 **ID-number** int

3 **day workload** int > 1

Number of samples coming in during the day.

4 **operator request** int #11 > 3

In the simulation a day is split up into shifts (a number of day shifts and one night shift; the maximal number of shifts is 11). Here the minimal number of operators for each possible shift is collected.

5 **maximum workload** int > 1

Maximum number of samples the shop can handle.

6 **authorization time** int > 1

Time (in hours) between two authorizations (during day time only).

7 **control queue length** int > 1

Minimal number of samples in the control queue for authorization.

8 **reassignment weight** real > 3

Value equals 1 if reassignment takes place on the basis of the shop parameters (the 'reassignment rule'); 0 if it is done by using the phase weights of the workstations; any real between 0 and 1 is the weight the reass. rule will have.

9 **workload control** switch > 2

Equals 1 if the samples have to be buffered at the beginning of the shop; else 0.

10 **transfer time** real > 1

time (in minutes) needed to transfer an operator from one workstation to another.

11 **reassignment rule** int #20 > 2

weights on the shop parameters for the (re-)assignment of the operators to the workstations during a shift.

12 **workstation weights** int #20 > 10

workstation dependent weights on the shop parameters. These weights ensure that the workstations can be compared.

The following code is used:

0: workstation weights are constant;

1: weights equal the cycletime (time needed to process a batch);

2: weights equal cycletime/batchsize;

3: reciprocal of cycletime;

4: batchsize/cycletime;

else: the weights as filed with the workstation data (field 9: assignment std-rule).

13 **sequencing rule** int #20 > 2

weights on the shop parameters for the assignment of the samples in the job shop queue to the workstations (not used).

14 **interarrivals** real #24 > 2

The interarrival times of the samples during the day, and specified for each hour (interarrival time 0 stands for 'no sample arrivals').

The next three pages contain the calibrated parameters for the shop configurations discussed in this study.

haematology																				
1	400																			
3	0	0	0	0	0	0	0	0	0	0										
750	2	100	1	0	5															
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0		
0	0	0	1	0	0	1	0	0	0	5	0	0	0	0	0	0	0	0		
1	1	1	1	1	1	3	3	1	1	1	1	1	1	0	0	0	0	0		
0	0	0	0	0	0	0	0	6.86	4.58	6.27	3.76	8.14	4.62	7.79	6.53	3.90	0	0	0	0

batch_chemistry																						
3	450																					
2	0	0	0	0	0	0	0	0	0	0												
570	2	100	1	0	0																	
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0			
0	0	0	1	0	0	1	0	0	0	5	0	0	0	0	0	0	0	0	0			
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0			
0	0	0	0	0	0	0	0	15.90	3.34	5.18	2.87	17.20	4.06	5.05	7.13	8.28	0	0	0	0	0	0

discrete_chemie																						
8	300																					
2	1	0	0	0	0	0	0	0	0	0												
150	2	100	1	0	0																	
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0			
0	0	0	1	0	0	1	0	0	0	5	0	0	0	0	0	0	0	0	0			
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0			
0	0	0	0	0	0	0	0	6.06	4.33	7.46	3.12	8.68	4.86	5.54	6.36	6.38	0	0	0	0	0	0

APPENDIX E: WORKST.D

In the figure on the next page the record structure is presented.

1															
2	3	4	5	6	7										
8															
9															
10	11														
12															
13	14	15	16	17	18	19									
20															
21															
22															
23															
24															
25	26	27	28	29											
30	31	32	33	34	35										

WORKST.D >4

This file contains the data wrt. the workstations: each record specifies or quantifies a workstation. In the simulation model a workstation is precisely a part of the job shop where one operator can perform a (part of a) test; the resources such as machines and instruments are thought to be present.

(Field#, Name, Type)

1 **workstation name** text

2 **ID-number** int

3 **job shop** int > 1

ID-number of the jobshop the workstation is part of.

4 **quantity** int > 7

number of workstations of this type. The method of filing an incoming sample for this type of workstation is handled by the distribution field. A rejected sample is relocated in the same way as others. If this field is valued -1 the workstation acts as a transition node: a sample is directed directly to one of the next workstations (without recording for this workstation) according to the principles as explained in

the helptext of the fields 15 to 19. further, if this field is valued 0 it is not considered by the simulation.

5 **distribution** int >5

if there are more workstations involved then this value defines how an arriving sample has to be distributed:

0: arbitrary;

1: equally;

otherwise: first workstation up to the max. queue length.

6 **maximum queue length** int >1

For its use see distribution field.

7 **priority handling** int >3

This field specifies how the priority sample should be adapted after being processed; value 0: sample's priority changes to 4 or 5; 1..5: pr. changes into this value; 6: no changes; 7: new arrival time; 8: priority one level higher.

8 **phase weights** int #7 >4

In the simulation model a workstation is split up into 7 phases: these are called 'setup', 'batch preparation', 'batch processing', 'batch release', 'shut down', 'inactivity', and 'ready'. The weights in this field express the necessity of an operator in each phase of the workstation.

9 **assignment std-rule** int #20 >3

workstation dependent weights on the shop parameters for the (re-) assignment the operators within the job shop. These weights ensure that the workstations can be compared.

10 **day workload weight** int >1

number of samples coming in during the day.

11 **night workload weight** int >2

number of samples coming in during the night (in regular/standard week) (not used).

12 **priority weights** int #5 >5

There are 5 priority levels, corresponding to the maximal allowable throughput times 15 minutes, 90 minutes, 4 hours, one day, and 10 days. Level 1 and 2 are considered to be 'urgent' (also 3 if there is only one day shift). Now each incoming sample is assigned a priority level with probability proportional to the weights in this field.

13 **rejection workstation** int >3

ID-number of the workstation a sample is directed after rejection by the first or final authorization. Here and for the next 3 fields a 0 means the job shop queue and any non-existing ID-number the authorization queue.

14 **standby workstation** int >2

ID-number of the workstation a sample is directed in case the workstation is not available because of break down or shut down (only for urgent samples).

15 **first next workstation** int >2

ID-number of next workstation for processing, chosen with probability as expressed in field 19.

16 **second next workstation** int >2

ID-number of next workstation for processing if first next workstation is not chosen.

17 **first rejection probability** int >4

percentage expressing the probability that a sample is rejected at release (by the operator). As for 'rejected' samples (see field 11) the priority may change along the lines described in field 4. One slight difference is that rejected samples which are allocated for the same workstation keep the same priority.

18 **second rejection probability** int >2

percentage expressing the probability that a released sample is rejected for final release (by the authorization). See also field 17.

19 **relocation probability** int >3

percentage expressing the probability that a released sample is relocated to the workstation with ID-number in field 15 (if prob. is 0 the sample is released for final authorization).

20 **mean processing times** real #5 >3

Here data is collected of the processing times. In 5 of the phases some time is needed to perform the operations; these phases are setup, batch preparation, batch processing, batch release and shut down. The times are in minutes.

21 **variance processing times** real #5 >1

These values express the density of the processing times around their mean.

22 **distribution** int #5 >4

The distribution values consist of one of the four numbers 0,1,2 and 3; the value 0 corresponds to the situation that the processing times are fixed, i.e. is there is no uncertainty (the deterministic case), value 1 corresponds to the normal distribution, 2 to the exponential distribution, and 3 to the uniform distribution.

23 **operator presence** switch #5 >3

The operator presence values are switches: it equals 1 if the operator is needed during the whole operation; 0 if he may leave the workstation after the necessary preparations.

24 **proportionality** switch #5 >3

switches: a value 1 ensures that the mean time in the first row is proportional to the number of samples in the batch; and 0 if it is for the batch as a whole (for the setup and shutdown phase it has no meaning).

25 **batch size** int >1

the maximal number of samples a batch unit can hold

26 **minimal batch size** int

27 **calibration action** int >5

number of processed samples after which calibration has to take place (implementation: an empty batch is processed without preparing and releasing; processing time is computed assuming a batch with one sample. The time needed equals a fraction of the processing time; the other features like distribution etc., are the same as for the batch processing phase)

28 **reassignment action** int >1

number of processed batches after the operator calls for reassignment.

29 **batch processing action** real >3

value is multiplied by time needed to process a batch (cycletime); the resulting value is the time within a due date triggers batch processing regardless the number of queued samples (in case operator is present)

30 **shut down action** int >1

number of processed samples after which a shut down takes place.

31 **calibration fraction** real >1

fraction of the processing time needed for the calibration action

32 **implementation action** int > 1

number of processed samples after which an implementation takes place

33 **implementation time** real > 1

time in minutes needed for the implementation

34 **throughput time** real > 1

workstation time in minutes needed for a sample to be completed (not used)

35 **technician time** real > 2

technician time in minutes needed for a sample to be completed (only used in workload control for releasing samples from the shop queue).

CALIBRATED PARAMETERS FOR WORKSTATIONS:

WORKSTATIONS FOR HAEMATOLOGY

Bulk haematological analyzer_1																
1	1	1	1	0	6											
1	2	5	3	1	0	1										
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
400	0															
3	25	51	21	0												
1	1	2	99	0	0	100										
60	3	34	0	0												
0	0	0	0	0												
0	0	0	0	0												
1	1	0	0	0												
0	0	0	0	0												
39	20	400	0	1												
0	1	0	0	0	0											

Bulk haematological analyzer_2

2	1	1	1	0	3															
1	2	5	3	1	0	1														
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0																			
3	25	51	21	0																
1	2	4	99	4	4	10														
0	0	0	15	30																
0	0	0	0	0																
0	0	0	0	0																
0	0	0	1	1																
0	0	0	0	0																
39	20	1000	0	1																
0	1	0	0	0	0															

Eye blood cell differential																			
4	1	0	1	2	5														
1	2	5	3	1	0	1													
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0																		
0	0	30	70	0															
0	0	4	5	4	7	10													
10	5	2	7	10															
0	0	0	0	0															
0	0	0	0	0															
1	1	0	1	1															
0	0	0	1	0															
12	6	1	0	1															
0	1	0	0	0	0														

Small size haematological analyzer																	
3	1	5	1	2	5												
1	2	5	3	1	0	1											
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	0																
100	0	0	0	0													
1	100	3	5	3	2	10											
0	10	1	2	10													
0	0	0	0	0													
0	0	0	0	0													
1	0	1	1	1													
0	0	1	1	0													
15	1	5	0	1													
0	1	0	0	0	0												

WORKSTATIONS BATCH CHEMISTRY

Bulk chemical analyzer_1																			
21	3	0	1	2	5														
1	2	5	3	1	0	1													
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
300	0																		
5	10	25	60	0															
22	100	21	5	21	0	0													
60	5	6	0	0															
0	0	0	0	0															
0	0	0	0	0															
1	1	0	0	0															
0	0	1	0	0															
1	1	48	0	1															
0	1	0	0	0	0														

WORKSTATIONS DISCRETE CHEMISTRY

Specific test analyzer																	
56	8	0	1	2	5												
1	2	5	3	1	0	1											
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
100	0																
0	0	80	20	0													
0	0	0	10	56	5	0											
0	11	6	5	0													
0	0	5	3	0													
0	0	1	3	0													
1	1	0	1	1													
0	0	1	0	0													
30	3	3	0	0													
0	1	30	5	0	0												

Specific test analyzer_2																	
57	8	0	1	2	5												
1	2	5	3	1	0	1											
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
200	0																
0	10	80	10	0													
0	0	0	10	56	5	0											
0	11	6	5	0													
0	0	5	3	0													
0	0	1	3	0													
1	1	0	1	1													
0	0	1	0	0													
30	3	3	0	0													
0	1	30	5	0	0												

APPENDIX F: PROCESS.D

Record description:

- 1. Name
- 2. ID, 3. max_workload, 4. op_time, 5. real_workload
- 6. assignment periods
- 7. priority class 1
- 8. priority class 2
- 9. priority class 3
- 10. priority class 4
- 11. priority class 5

INPUT DATA FOR OPTIMIZATION MODEL: the appropriate process times are given in chapter 6. In the following for field 4 an arbitrary value is given.

Bulk chemical analyzer

1				100			1			445
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
5	5	0	0	0	0	0	0	0	0	0
0	25	35	35	20	20	0	0	0	0	0
50	50	50	50	50	50	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

Specific test analyzer

2				35			2.5			150
0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	5
0	2	2	0	0	0	5	5	5	5	45
0	10	10	0	0	0	0	0	0	0	0
0	0	10	10	10	10	10	10	0	0	0
0	0	0	0	0	0	0	0	0	0	0

Specific test analyzer

3				35			2.5			195
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	5	10	5	0	0	0	10	5	30	0
0	10	10	0	0	0	0	0	0	0	0
0	0	10	20	20	20	20	10	10	0	0
0	0	0	0	0	0	0	0	0	0	0

Glucose and chloride analyzer

4			100			2			200
0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	1	0	0	0	2
5	5	0	5	0	0	0	5	0	25
0	0	0	10	20	20	20	10	10	0
0	0	0	0	0	0	20	20	20	0
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0

Bulk haematological analyzer

5			80			1.5			425
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	20	20	10	10	0	0	10	5	0
5	15	15	25	25	5	5	5	0	0
0	0	0	25	45	45	45	45	45	0
0	0	0	0	0	0	0	0	0	0

Small size haematological analyzer

6			50			2			95
0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1
5	10	5	5	5	5	5	5	5	30
0	0	0	0	0	0	0	0	5	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Eye blood cell differential

7			15			2.5			55
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	2	2	0	0	0	0	0	0	0
0	0	5	10	10	10	5	5	5	0
0	0	0	0	0	0	0	0	0	0

ESR Erythrocytes Sedimentation Rate

8			60			2			125
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	2	2	1	0	0	0	0	0	0
0	10	10	10	5	0	0	0	0	0
0	0	10	10	10	10	15	15	15	0
0	0	0	0	0	0	0	0	0	0

Immuno assays

19						90			10				145
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	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	20	0	0	0	0	0	0	0	0	0	0	0	0
0	50	50	25	0	0	0	0	0	0	0	0	0	0

Reception area out-patients samples

20						60			1.5				600
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	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	25	100	125	100	50	25	100	50	25	100	50	25	25
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Reception area in-patients samples

21						80			1.5				455
0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	5	5	5	5	5	5	5	5	5	5	5	5	10
50	75	75	50	25	15	15	30	15	30	15	30	15	50
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	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

APPENDIX G: HEURISTIC FOR THE OPTIMIZATION PROBLEM

As an alternative to the approach described above also a program has been developed using a heuristic. In fact both programs can also be used together where the former approach gives bounds for the heuristic solution. We explain the heuristic by describing the program steps.

Step 1. Order the process types in a list L

To achieve such a list we should use a criterion, expressing somehow how hard it is to schedule a process type within any job shop. The following arguments should be considered:

(a) the longer the period between release-time and due-time, the better the average freedom of that period is:

$$\frac{\sum_{g=1}^G \sum_{h=g}^E \delta_{pgh}(h-g)}{\sum_{g=1}^E \sum_{h=g}^E \delta_{pgh}}$$

(b) the more time is required in total over all orders of a certain process type, the worse; the "total workload" for the process type p is:

$$w(p) \sum_{g=1}^E \sum_{h=g}^E \delta_{pgh}$$

(c) the larger the process time for an individual order, the worse; the importance of the difference between the process times is dependent on the actual value of the process times: the smaller the worse; this could be expressed with:

$$\sqrt{w(p)}$$

The proposed measures are quite arbitrary. To combine them into one criterion, again there are several options. In this case we suggest simply to multiply them with each other, yielding

$$\varphi(p) = w(p) \sqrt{w(p)} \sum_{g=1}^E \sum_{h=g}^E \delta_{pgh}(E-h+g)$$

Step 2 Assign the first n elements of L to different job shops

So the harder process types are not in the same job shop. Now for each job shop the minimum and maximum number of technicians required must be calculated.

Repeat until list L is empty

Step 3 Take the first element from L and assign it to a job shop

Compute for each job shop the consequences of assigning the process type to that job shop, and select the job shop with the 'best' result. To avoid infeasibilities the minimum and maximum number of technicians required for each job shop should be monitored carefully.

Repeat for $e=1$ to E :

Step 4 Order the job shops in a list S_e

To achieve S_e , a criterion similar to the one in step 1 could be used.

Repeat until S_e is empty:

Step 5. Take the first element of S_e and assign a number of technicians to this shop in this period.

For example the maximum number of technicians. Anyhow this will have implications for the minimum and maximum number of technicians in other job shops, as well as in the remaining periods $e+1, \dots, E$.

Repeat for each job shop $j=1..n$
repeat for each period $e=1$ to E :

step 6. Order the process types of shop j in period e in a list L_{ej} .

repeat until L_{ej} is empty:

step 7. Take the first element of L_{je} and assign a number of orders to that process type to period e .

This will have (severe) implications for the minimum and maximum number of orders that can be assigned to the remaining periods.

Of course, this heuristic can be implemented in many ways. If the number of technicians available is reasonable compared with the amount of work to do, a heuristic like described here will yield a feasible solution in many cases. At least,

this solution can be used as a starting point for some improvement heuristic. Furthermore, by generating several different lists of process types L , a large set of different solutions can be generated quickly. Then, a manager can select the solution he prefers (e.g. based on criteria not explicitly mentioned).

This study aims at the development of a decision support system (*DSS*) for clinical laboratory capacity planning. A typical characteristic of clinical laboratories is that the future demand is difficult to predict. This is also true for operational periods (for example for one shift). As a consequence clinical laboratories need a flexible production system to provide services efficiently and effectively. The *DSS* gives decision-makers advices on the following problems:

- How can the laboratory be clustered into job shops?
- Which planning rules can be used within job shops?

In chapter 2 the planning and control system is discussed. A definition of such a system is necessary for the *DSS* to support management. The system is elaborated for the clinical laboratory in chapter 3. It lays emphasis on the way technicians are assigned to laboratory processes and on the way samples are planned for processing. Important is the distinction between detailed and aggregate techniques for planning and control. Detailed techniques concern for example rules for assigning samples to workstations on the basis of specific order characteristics. An example of an aggregate technique is the allocation of staff to job shops on the basis of the expected workload.

Chapter 4 analyzes information aspects of planning in the clinical laboratory as well as the *DSS*.

Chapter 5 discusses the simulation model, the way this model is implemented in the *DSS*, and the simulation experiments which have been carried out. The simulation module aims at simulating job shops to determine the performance of planning rules, the influence of demand characteristics (in terms of volume and required throughput times) and the available staff capacity. In this way it is possible to define and test planning rules for different demand situations in a very flexible way. Methods are discussed for collecting data. Statistical methods for determination of the number and length of simulation runs are also discussed. To be able to estimate the performance of instances which have not been simulated a regression-analysis is conducted on a sample of possible instances. It appears that this so-called metamodeling method offers good but not always easy interpretable results.

Chapter 6 discusses the model optimizing the clustering of workstations into job shops. This model tries to find such a clustering of workstations that minimizes the maximum idle time of staff in each assignment period. The results of the model are positively valued by laboratory management.

Chapter 7 discusses conclusions and suggestions to improve the performance of the *DSS*. It appears that the *DSS* performs well, but that improvements could be realized by changing the criterion for checking the results of a simulation run. Changing this criterion could result in metamodels which are easy to interpret. Additional research could also be directed to methods which decrease runtimes.

Deze studie is gericht op het ontwikkelen van een besluitvormingsondersteunend systeem (*DSS*) ten behoeve van het planning van capaciteit voor ziekenhuislaboratoria. Hierbij gaat het met name om klinisch chemische en hematologische laboratoria. Een typisch kenmerk van ziekenhuislaboratoria is dat de toekomstige vraag moeilijk voorspelbaar is. Dit geldt ook voor operationele planningsperioden (zoals binnen één dienst). Het gevolg hiervan is dat laboratoria een flexibel productiesysteem nodig hebben om efficiënt en effectief diensten te kunnen leveren. Het besluitvormingsondersteunend systeem geeft de besluitvormers adviezen omtrent de volgende type problemen:

- Hoe kan het laboratorium worden verdeeld in afdelingen?
- Welke planningsregels kunnen worden gehanteerd binnen de afdelingen?

In Hoofdstuk 2 worden algemene aspecten van een besluitvormingsondersteunend systeem besproken en wordt ingegaan op een aantal centrale begrippen. Verder biedt dit hoofdstuk een korte analyse van de laboratorium structuur en de planningsproblemen waarmee het laboratorium-management wordt geconfronteerd. Duidelijk wordt gemaakt dat in een laboratorium sprake moet zijn van een plannings- en beheersingssysteem en dat een *DSS* ten behoeve van managementvragen rekening moet houden met dit plannings- en beheersingssysteem.

In hoofdstuk 3 wordt nader op dit plannings- en beheersingssysteem ingegaan. Aan de orde komen zowel een stuk theorie als een toepassing hiervan op het laboratorium. Eerst wordt de samenhang van capaciteitsbeslissingen met andere type beslissingen besproken. Het gaat hierbij met name om de manier waarop beschikbare menskracht wordt toegewezen aan laboratoriumprocessen en de wijze van planning van de monsters die moeten worden geanalyseerd op de werkstations. Belangrijk is het onderscheid dat wordt gemaakt tussen gedetailleerde en geaggregeerde beheersingstechnieken. Bij gedetailleerde beheersingstechnieken gaat het bijvoorbeeld om regels voor het toewijzen van orders aan werkstations op basis van specifieke orderkenmerken. Bij geaggregeerde beheersing gaat het bijvoorbeeld om het toewijzen van personeel aan afdelingen op basis van de verwachte gemiddelde werklast.

In Hoofdstuk 4 worden de informatie-aspecten van het ziekenhuislaboratorium beschreven. Bovendien wordt de structuur van het besluitvormingsondersteunend systeem besproken door middel van het aangeven van de modules en de datastromen daartussen.

Hoofdstuk 5 beschrijft het simulatie-model, de wijze waarop de model is geïmplementeerd en de simulatie-experimenten die zijn verricht. De simulatiemodule maakt het mogelijk laboratorium job shops te simuleren. In zo'n simulatie wordt voor een gegeven job shop (een verzameling werkstations) nagegaan hoe de prestatie worden beïnvloed door de planningsregels, de vraagkenmerken (in termen

van volume en vereiste doorlooptijden) en de beschikbare personele capaciteit. Op deze manier is het op een zeer flexibele manier mogelijk planningsregels te definiëren en te testen onder verschillende vraagomstandigheden en in verschillende type job shops. Tevens worden methoden behandeld voor het verzamelen van de data. Statistische aspecten komen ook aan de orde bij het bepalen van het aantal simulaties en de lengte van iedere simulatie-run. 'Metamodellering' om de gevoeligheid van planningsregels te bepalen werd ook getest. In deze methode wordt een regressie-analyse uitgevoerd op resultaten van simulaties om de prestatie van instanties te kunnen schatten die niet werden gesimuleerd. Het blijkt dat goede maar niet altijd even gemakkelijk te interpreteren metamodellen kunnen worden geschat.

In hoofdstuk 6 wordt het zogenaamde optimalisatie-model besproken. Dit model is gericht op het optimaliseren van de organisatie van het gehele laboratorium door middel van het clusteren van de werkstations op een zodanig manier dat de maximale leegloop van personeel die per toewijzingsperiode per job shop kan optreden wordt geminimaliseerd. Een dergelijke model wordt beschreven voor zowel deterministische als stochastische vraagkenmerken. Het model blijkt herkenbare en door het laboratorium management positief gewaarde oplossingen te genereren.

In hoofdstuk 7 'Discussie en conclusies' worden aspecten van het besluitvormingsondersteunend systeem besproken die, ondanks dat de DSS bruikbare resultaten oplevert, mogelijk voor verbetering in aanmerking komen. Te denken valt met name aan het afkapcriterium dat wordt gebruikt om te bepalen of een bepaalde simulatie-run bruikbare resultaten oplevert. Een verbetering van dit criterium zou metamodellen opleveren die gemakkelijk kunnen worden geïnterpreteerd. In het algemeen dient ook verder onderzoek te worden gedaan naar de mogelijkheden om runtijden te verkorten.

Tijdens mijn studie economie aan de Erasmus Universiteit te Rotterdam heb ik in het kader van mijn hoofdvak Interne organisatie een onderzoek gedaan naar de implementatie-problemen van flexibele fabricagesystemen. Daarbij was mijn interesse gewekt voor de interactie van techniek en organisatie. Mijn werkgever waar ik na mijn afstuderen kwam te werken, de HEAO-limburg, stelde mij in staat mijn informatica kennis te verdiepen, door middel van een aanvullende studie informatica aan de Technische Universiteit te Eindhoven. In 1987 kwam ik te werken bij de vakgroep Beleidswetenschap aan de Rijksuniversiteit Limburg op het vakgebied Bestuurlijke Informatiekunde. Mijn promotoren van het eerste uur, prof. dr. ir. Arie Hasman en prof. dr. Hans Maarse brachten mij op het spoor mijn belangstelling voor produktiesystemen en informatica toe te passen op het steeds meer actuele probleem van de planning in ziekenhuizen. Arie Hasman heeft mij op zeer stimulerende wijze begeleid bij het uitvoeren van het onderzoek en bij het schrijven van dit proefschrift. Aan hem ben ik veel dank verschuldigd. Hans Maarse wil ik bedanken voor zijn adviezen en de ruimte die hij mij heeft gegeven bij de invulling van het onderzoek.

In 1990 werd besloten het onderzoek toe te spitsen op de klinisch chemische en haematologische laboratoria. Dr. Jan Leijten introduceerde mij in het Centraal klinisch chemisch en haematologisch laboratorium te Tilburg. Daarmee was de basis gelegd voor een vruchtbare samenwerking. Samen met zijn opvolger (en mijn co-promotor) Dr. Henk Goldschmidt werd het besluitvormingsondersteunend systeem geconcretiseerd, hetgeen resulteerde in een prototype van de simulatie-module in het begin van 1991. Subsidies van de EC Programma in het kader van het Advanced Informatics in Medicine fonds, maakte een verdere ontwikkeling van het besluitvormingsondersteunende systeem mogelijk als onderdeel van het OpenLabs project. Bovendien kwam daarmee nog een andere samenwerking tot stand, namelijk die met de Vakgroep Wiskunde. De betrokkenheid van de vakgroep Wiskunde kwam op meerdere manieren tot uitdrukking. Prof. dr. ir. drs. Koos Vrieze was bereid als promotor op te treden. Dr. Jean Derks legde zich toe op de ontwikkeling van het simulatie-model en stond altijd klaar om mij van advies m.b.t. alle aspecten van mijn proefschrift. Drs. Maarten Oosten werkte de optimalisatie-module uit. Vanaf 1991 was Bob Schoenmaker van de vakgroep Beleidswetenschap als medewerker bij het project betrokken. Vanuit het Academisch Ziekenhuis Maastricht is Dr. Fred Nieman vanaf 1989 betrokken geweest bij het ontwikkelen van het onderzoeksmodel. Ik wil al deze mensen danken voor hun inzet en steun.

De medewerkers van het Centraal klinisch chemisch en haematologisch laboratorium wil ik bedanken voor hun enthousiaste en onontbeerlijke steun aan het onderzoek.

DIV te Tilburg verzorgde het toegankelijk maken van een groot deel van de benodigde data. Deze data werd beschikbaar gesteld door het St. Elisabeth ziekenhuis en het Maria ziekenhuis te Tilburg. Ik ben deze instellingen hier zeer erkentelijk voor.

Hoewel mijn aandachtgebied Bestuurlijke Informatiekunde een soms wat esoterische indruk maakte op mijn collega's Beleidswetenschappers heb ik mij op mijn vakgroep altijd thuis gevoeld. Mijn collega's wil ik daarvoor graag bedanken. Ook met Richard Janssen en Jan Winter van de vakgroep Economie van de Gezondheidszorg heb ik graag samengewerkt in onderzoek en onderwijs. Van de zelfde vakgroep ben ik ook veel dank verschuldigd aan Jan van Emmerik die mij altijd hielp als hij mij kon helpen.

Prof. dr. ir. J.L.G. Dietz, prof. dr. A.R. Bakker, prof. dr. P.J. Brombacher, prof. dr. ir. A. Kolen, prof. dr. R.M. Leidl wil ik danken voor hun bereidheid in de beoordelingscommissie zitting te nemen en voor hun waardevolle commentaar.

De lengte van mijn dienstverband bij de Rijksuniversiteit Limburg, wijkt nauwelijks af van mijn 'huwelijkse' tijd met Birgit. Beloften van mijn kant dat ik mijn leven als HET klaar is, zal verbeteren werden met ongeloof ontvangen, tegelijkertijd vond zij het vanzelfsprekend dat dit proefschrift geschreven zou worden. Haar vriendschap en geloof in mijn kunnen waren daarbij onmisbaar. Met veel blijdschap denk ik aan Goord voor wie observatie en experiment vanzelfsprekende methoden zijn.

Curriculum vitae

Frits van Merode werd geboren op 18 november 1958 te Geleen. Hij volgde de MAVO (Geleen, 1971-1975), de MEAO-CE (Sittard, 1975-1978), HEAO-BE (Sittard, 1978-1981). Vervolgens studeerde hij aan de Erasmus Universiteit te Rotterdam Economie (kandidaats-diploma, 1983; doctoraal-diploma, 1985) en Wijsbegeerte (doctoraal-diploma, 1991). Van medio 1985 tot medio 1987 was hij werkzaam bij de HEAO-School Limburg te Sittard als docent Bestuurlijke Informatiekunde, Automatisering en Bedrijfsadministratie. In september 1987 trad hij als universitair docent in dienst bij de vakgroep Beleidswetenschap van de Faculteit der gezondheidswetenschappen aan de Rijksuniversiteit Limburg.

