STELLINGEN

behorende bij het proefschrift

BUSINESS REENGINEERING IN INFORMATION INTENSIVE ORGANIZATIONS

Remko C.J. Dur

8 september 1992
I

Ontwerpers besteden bij het ontwikkelen van informatiesystemen weinig aandacht aan de inrichting van de te ondersteunen bedrijfsprocessen. De inrichting van deze processen moet echter expliciet als één van de ontwerpvariabelen worden beschouwd.

II

Geautomatiseerde hulpmiddelen voor informatiesysteemontwikkeling richten zich primair op het toegankelijk en gebruiksvriendelijk vastleggen van modellen. Door dynamische modelleer-technieken op te nemen in dergelijke hulpmiddelen kan het accent verschuiven naar het effectief ondersteunen van de analyse, de diagnose en het oplossen van problemen in organisaties.

III

In een vakgebied waar het vervaardigen van hulpmiddelen een essentieel onderdeel is van wetenschappelijk onderzoek, moet er ter bevordering van de effectiviteit van dit onderzoek een beter evenwicht worden gevonden tussen onderzoek (research) en ontwikkeling (development). Het gericht toewijzen van ondersteunend technisch personeel aan dergelijk onderzoek is hiervoor een middel.

IV

Bij het in rekening brengen van kosten voor rekeningcourantdiensten aan particulieren gaan banken voorbij aan het feit, dat deze dienstverlening een middel is geweest om spaargelden aan te trekken. De kosten/opbrengsten-verhouding op basis waarvan men tot doorberekning besluit, zou dan ook over het gehele dienstens scala aan particulieren moeten worden bepaald.
V

Zoals reeds gebruikelijk bij onder meer juridische en fiscale adviezen, zou het advies van een assurantie- of hypotheekadviseur moeten worden doorberekend aan de klant. Hierdoor kan de provisiestructuur tussen maatschappijen en tussenpersonen worden geëlimineerd hetgeen een objectief advies mogelijk maakt.

VI

Met de steeds mondialere economie is het voor bedrijven mogelijk om het ontwerp, de produktie en de distributie van nieuwe produkten volledig buiten de eigen landsgrenzen uit te voeren. In een dergelijke economie verliest de handelsbalans zijn waarde als evaluatiemiddel voor de handel tussen landen.

VII

In een 'intake'-procedure, zoals die binnen de ergotherapie wordt toegepast, worden de capaciteiten van een persoon afgezet tegen de van die persoon verwachte prestaties. Langs deze weg wordt bepaald welke vormen van training en welke hulpmiddelen noodzakelijk zijn. Een dergelijke procedure kan model staan voor een werkwijze voor het ontwikkelen van beslissingsondersteunende systemen.

VIII

Het verhogen van het aantal knoppen op een muis, totdat het degenereert tot een rollend toetsenbord, verlaagt het gebruiksgemak ervan. Leveranciers van één-knoppige muizen hebben dit correct ingeschat.
IX

Wanneer het gebruik van principes uit de logistiek voor de herinrichting van informatieverwerkende bedrijven, ook bij nieuwe produktontwikkeling in die bedrijven wordt toegepast, kan dit bizarre gevolgen hebben. Men denke bijvoorbeeld aan het produkt just-in-time levensverzekering.
BUSINESS REENGINEERING IN INFORMATION INTENSIVE ORGANIZATIONS
BUSINESS REENGINEERING
IN INFORMATION INTENSIVE ORGANIZATIONS

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus,
Prof. drs. P.A. Schenck,
in het openbaar te verdedigen
ten overstaan van een commissie
aangewezen door het College van Dekanen
op dinsdag 8 september 1992 te 14.00 uur

door

Remko Cornelis Johannes Dur

informatica ingenieur

geboren te Nieuwkoop
Dit proefschrift is goedgekeurd door de promotor:

Prof. Dr. H.G. Sol

en door de leden van de promotiecommissie:

Prof. Drs. B.K. Brussaard
Prof. Dr. Ir. J.J.M. Evers
Prof. Dr. Ir. M. Looijen
Prof. Dr. J.J. Ramondt
Prof. Dr. I.Th.M. Snellen
Many of our job designs, work flows, control mechanisms, and organizational structures came of age in a different competitive environment and before the advent of the computer. They are geared toward efficiency and control. Yet the watchwords of the new decade are innovation and speed, service and quality.

Michael Hammer, 1990

For my parents — for their unremitting support

For Annemieke — who is the sunshine of my life
Published and distributed by
R.C.J. Dur
P.O. Box 356
2600 AJ Delft,
The Netherlands

The author acknowledges support by the
Technology, Labour and Organization Programme (T.A.O.)

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Dur, Remko Cornelis Johannes

Business Reengineering in Information Intensive Organizations /
Remko Cornelis Johannes Dur. - [S.l. : s.n.]. - Ill
Proefschrift Technische Universiteit Delft. - Met lit. opg.,
reg. - Met samenvatting in het Nederlands.
Trefw.: probleemoplossen (management) / organisatieontwikkeling.

© 1992, by R.C.J. Dur

No part of this publication may be reproduced, stored in a retrieval system, or
transmitted in any form by any means, mechanical, photocopying, recording or
otherwise, without the prior permission of the author.
Contents

1 Improving organizational effectiveness
   1.1 Information technology and organizations .................. 1
   1.2 The growing need for adaptability .......................... 2
   1.3 Narrowing the scope ......................................... 4
   1.4 Research approach and outline of the study .............. 7

2 Business reengineering: Theories and methods
   2.1 Introduction .................................................. 11
   2.2 A problem solving view ...................................... 12
   2.3 Problems and human problem solving ....................... 16
   2.4 Lessons learned from existing methods ................... 18
   2.5 Towards effective support ................................... 21
   2.6 Refining the research question ............................. 24

3 A dynamic modelling approach
   3.1 Introduction .................................................. 27
   3.2 Entity modelling .............................................. 27
   3.3 Business process modelling ................................... 33
   3.4 Analysis of dynamic model behavior ....................... 44
   3.5 Summary and conclusions ................................. 51

4 A dynamic modelling support environment
   4.1 Introduction .................................................. 53
   4.2 Analysis of the business reengineering process .......... 54
   4.3 Functional requirements of a DMSE ......................... 69
   4.4 Summary and Conclusions ................................. 88
Contents

5 Case 1: A governmental department
  5.1 Introduction ........................................... 89
  5.2 Specification .......................................... 91
  5.3 Validation ............................................ 107
  5.4 Diagnosis, solution finding and evaluation .......... 113
  5.5 Summary and conclusions ............................ 119

6 Case 2: An insurance firm
  6.1 Introduction ........................................... 121
  6.2 Specification .......................................... 123
  6.3 Validation ............................................ 146
  6.4 Diagnosis, solution finding and evaluation .......... 153
  6.5 Summary and conclusions ............................ 170

7 Case 3: A municipal department
  7.1 Introduction ........................................... 173
  7.2 Specification .......................................... 175
  7.3 Validation ............................................ 193
  7.4 Diagnosis, solution finding and evaluation .......... 199
  7.5 Summary and conclusions ............................ 210

8 Epilogue
  8.1 Introduction ........................................... 213
  8.2 Research findings .................................... 214
  8.3 Consequences for information systems development .. 218

References .................................................. 221

Author index ................................................ 237

Subject index ............................................... 241

Samenvatting ............................................... 247

Curriculum vitae .......................................... 255
A preface has two intended audiences: those who are interested in reading the book, and those who have made such reading possible.

For the first audience, I can be brief. This thesis presents a model based approach to organizational change where the focus is on organizations that produce goods or provide services which have a high information content. The challenges these organizations face induce them to find new and more effective ways to produce and market their goods. Changes that are required to the organization’s business processes or even to its commercial strategy, however, are very hard to design and implement. In light of these difficulties an approach is presented which emphasizes the need to understand the current business processes of an organization before effective changes can be designed.

For the second audience, I must elaborate, as many people have contributed to the writing of this thesis. First I want to thank my advisor Henk Sol, whose seemingly endless energy and enthusiasm have been a great source of inspiration to me during the past years. His comments, advices and encouragements have been plentiful, and the speed at which these were provided has truely impressed me.

Secondly, acknowledgements are due to Peter ten Bruggencate, Ad Grool, Peter van der Jagt, Albert Jansen, Jacques Martini and Ben Reijs who facilitated empirical research within the organizations they represent. They have provided an important base for evaluating the research findings in a practical setting. For this empirical research I am also indebted to the MSc students Peter Allemeinkinders, Ingo van Eijden and Kees Trommel who have invested much of their time, energy and skills in the research. I am thankful for the great challenges they provided by their constant requests for improvements in the approach and the modelling tools.

I want to thank my colleagues at the Department of Information Systems in Delft, and especially my friends of the Kongsj, for making the past four years so pleasant and enlightening. In particular I thank Richard de Jong for convincing me, over and over again, that we are working on the same subject: logistics, and Pieter Bots for all the discussions we had and for his indispeasable contribution to our ceaseless joint effort to ‘beat the bugs’.
Preface

With great gratitude I can look back on the years that lay behind me, where my parents have been an unremitting source of support and encouragement for me. I am grateful for the unconditional opportunities they have always provided.

Most of all I want to thank Annemieke for her love and support, and for her warm and lively companionship I enjoy so much.

Delft, June 1992

Remko C.J. Dur
1.1 Information technology and organizations

The progress in information technology over the past decades has been astonishing, and even more, the impact of this technology on organizations has been tremendous in many business fields. What started as an expensive and unstable technology, managed by an expert staff department far away from the every day operations of a firm, has penetrated deeply in organizations where it is supporting a wide range of business activities varying from process control to the support of management decision making.

Historically, the application of information technology in organizations has been characterized by a strong 'technology push'. When introduced, new technologies were superimposed upon the current processes and organization structures with the purpose of increasing their efficiency (Huber and McDaniel 1986b; Keen 1986). The high costs of the technology induced a prevalence of technical issues (Sol 1982; Hirschheim 1985) and invoked an emphasis on investment justification by means of calculating efficiency gains (Lincoln 1986; Berger et al. 1988).

To an increasing extent, however, interests are shifting from technical issues to organization related issues. Emphasis is on an organization's effectiveness in the market place, and information technology is seen as one of the means that can be used to facilitate improvements thereof. More and more, the impact of information technology beyond the support of existing business processes is recognized, strikingly so in the financial service industry (Clemons and Weber 1991). Recognition of this fact is invoking managers in organizations to shift their attitude from a rather detached one to one where they themselves determine feasible applications of the technology which serve the strategic objectives of their organizations (Keen 1991).
Improving organizational effectiveness

The change in attitude is demonstrated by a survey regarding key issues in information systems, as perceived by general and information technology managers. The technical issues that ranked high in 1983, such as 'end-user computing', 'integration of technology' and 'software development', were replaced in 1986 by organization related issues like 'competitive advantage', 'organizational learning' and 'IS role and contribution to organizational performance' (Brancheau and Wetherbe 1987). This last issue, ranked 4th in 1986, even gained importance in a later study conducted in 1988, where it ranked 2nd place (Watson and Branchear 1991). Further evidence for the shift in interest is given by Straub and Wetherbe (1989), who asked an expert panel to rank a series of technologies in order of their impact on organizations in the 1990s and early 2000s. According to these experts, the major changes in the next decade will not be in computational capabilities nor in new ways to manipulate data to support managerial decision making. It is the panel’s appraisal that the real gains in productivity will come from new organization forms that enhance the distribution of information and knowledge to knowledge workers in the organization.

As a consequence of the shift in interest, the role of information technology in organizations is changing from an efficiency improving technology towards that of an enabling technology, which can facilitate the introduction of new and more effective organization design options (Huber 1984, 1990). The principal impediment to this use in the past has been the inclination to design information systems around existing structures and processes, rather than to think imaginatively about what the organizational design could be if the potentials of modern technologies were exploited (Strassmann 1985; Huber and McDaniel 1986b). With this new role of information technology, its potential use for the organization will be considered in conjunction with new and more effective business processes (Robey 1983; Chen and Nonnamaker 1989; Hammer 1990; Drucker 1991a), or even with new organizational objectives and strategies (El Sawy and Pauchant 1988; Keen 1988, 1991). Questions concerning the application of information technology in organizations will hence less often arise from efforts that try to obtain an efficiency improvement, although this will always remain important. Instead, applications for information technology will to a growing extent originate from a process of rethinking organizational objectives and of creating new and more effective business processes to attain these objectives.

1.2 The growing need for adaptability

The changing role of information technology in organizations concurs with a transition from an industrial society to an information or service society. The changing products, service level demands and grounds for competition that evolve in this society, clearly affect the way in which organizations operate and the way in
Section 1.2  The growing need for adaptability

which they attempt to attain a sustainable competitive position (Masuda 1980; Toffler 1980; Martin 1981).

This post-industrial society (Bell 1973; Simon 1973a) is characterized by more and increasing turbulence (Drucker 1980; Naisbitt 1982; Huber 1984). New technologies and the effective use thereof will shorten the time spans for research and development, yielding shorter product life cycles. Similar advances in the fields of advertising and distribution will enable competitors to steal markets more quickly than before. Limitations that withstood such competition, such as geographical or cultural distances, are greatly diminished as improved communication and transportation technologies are implemented on a near-universal scale (Keen 1988). In general, individual events will reoccur more frequently, and be shorter in duration, stimulating the high tempo and turbulence of this environment.

Besides the increase in turbulence, post-industrial society is also characterized by an increase in complexity (Huber 1984). In their prediction of new directions for the 1990s, Naisbitt and Aburdene (1990) state some causes for this increase. New technologies have changed the importance of scale and location and extended the power of individuals. Entrepreneurs are able to start a business without towering initial investments. Also, communication technologies which enable people to move their work place away from the office pave the way for more contract work to be awarded to self-employed people. As a consequence, human action will less often be organized in collective structures but will shift towards interacting networks of loosely coupled systems (Sol 1992) or 'metabusinesses' (Keen 1989). The increase in diversity of competition will issue a growing demand for goods and services that suit individual needs. This, combined with the evolving 'globalization' causing an ever increasing potential extent of organizational activities, as well as an increased 'reach and range' of their technical systems (Keen 1991), causes the environment to be significantly more complex.

The increase in turbulence and complexity of the environment imposes greater demands on organizations, as it will affect the pace at which organizations need to adapt and change. Change will become the norm, and unpredictability a basic reality of business. Due to the fact that organizations need to stay compatible with their environment (Ackoff and Emery 1972; Duncan 1973; Quinn and Cameron 1983), organizational decision making will be called for every time the environment changes to a state that is incompatible with the organization. Although different coping strategies exist, like moving to a different environment (Miles and Snow 1978), changing the environment to a more compatible state (Weick 1979) or reliance on slack resources and other buffers (Galbraith 1977), the organization will often have to adapt to the changing demands. Therefore, decisions have to be taken regarding changes in the organization's strategies and business processes. The heightened turbulence of the organization's environment will require that these organizational adaptations be more frequent and faster.
The demand for frequent change, however, is opposed by the complexity of the decisions related to such changes. Often, an organization’s response to environmental complexity has been to increase the organizational complexity. Management layers, procedures and controls are added, driving organizations from simple structures to more complex ones, requiring a multitude of coordination and control mechanisms (Mintzberg 1979). Many large organizations today are approaching a level of complexity which can hardly be dealt with adequately. As private and public organizations grow larger and larger, they are becoming less and less flexible and responsive. These increasingly cumbersome and bureaucratic organizations are ill-suited to deal with the dynamics of globalization, declining margins, increased competition, and other destabilizers of the business status quo. The impact of required actions, which might imply radical changes in the technologies and processes employed or in what the organization markets as its goods or services, are, however, hard to assess accurately, and hence decision making about such changes is difficult (Huber and McDaniel 1986a).

1.3 Narrowing the scope

The contradiction between the required adaptability of organizations on the one hand, calling for higher frequency and speed of organizational change, and the complex nature of the decision making on such changes on the other hand, portrays a gap that needs to be closed. Mechanisms must be put in place which facilitate an ongoing organizational enhancement and change process through which 'experimenting' or 'self-designing' organizations can evolve (Hedberg et al. 1976, 1977; Nystrom et al. 1976).

Instead of focusing our research on the construction of such general mechanisms, applicable in any organization, a restriction is made towards a specific domain: information intensive organizations, where the adjective information intensive is used to point out that the product devised by the organization has a high information content and, hence, the main business processes of the organization concern information processing.

The selection of this domain is prompted by two considerations:

1. The implications of the increased turbulence and complexity of the post-industrial society, as outlined in the previous section, are particularly discerned by organizations within the selected domain.

2. The organizations in the selected domain have great difficulties in taking adequate measures to meet new demands.
To elaborate the first point, we point out some radical changes taking place both in the private and public sector.

In the private sector, a striking example of a competitive environment invoking change is the approaching European common marketplace. Especially in the service industry, drastic changes are required in order to sustain a competitive position in this new market (Drucker 1991b). Large financial service firms are emerging due to alliances, joint ventures, mergers and buy outs. Other service firms prefer what is called a market focus strategy (Porter 1980) and select a particular buyer group, product line or geographic market, in order to serve this market better than its competitors. Each in their own way, these organizations need to find an answer to the new challenges they face.

In the public sector, responsibilities are more and more decentralized, where local authorities attend to the needs of civilians within budgetary and legislative constraints imposed on them by central government. Hence, public organizations and institutions will need to deal with this shifting responsibility and the changes in the way in which their services should be provided.

To elaborate our second consideration for focusing on information intensive organizations, we can refer to the meagre productivity improvements made in the service sector, outdistanced by its manufacturing counterpart, despite the enormous capital investments that were made (Roach 1991; Michels and Welsh 1991). The fact that services provided by information intensive organizations can not be grasped physically, cause the knowledge about its business processes to be dispersed and partial for every organization member. Information technology that is applied in such organizations often does not lives up to expectations (Strassmann 1985; Bowen 1986; Morell and Fleischer 1988), illustrating the difficulty that these organizations have in finding effective courses of action.

With the selection of this domain, our research is focused on the process of changing information intensive organizations. From the previous section, the need for such organizations to change their structures and processes in order to adapt to shifting requirements is clear. The complexity of devising and evaluating changes in organizations calls for support offered to the decision makers that are responsible for attaining the correct adaptations. As stated above, part of the complexity of these changes in information intensive organizations stems from the intangible characteristics of their products. As opposed to a manufacturing process where the transformation from raw materials to the final product can be observed with relative ease, the process of 'making' a service is hard to envision, comprehend and analyze. However, without a thorough understanding of the business processes, constructing changes may be impossible, other than by making an 'intelligent guess'.
The main incentive for our research is the mentioned need for support for improving business processes in information intensive organizations. Specifically, the objective of our research is . . .

... to support the analysis of business processes in information intensive organizations, and to support decision making on rearrangements in these business processes for the purpose of improving the organization's performance.

As we narrow our research down to the support for improving business processes in information intensive organizations, we introduce a term that will be used to denote such a process. From now on, we shall use the term business reengineering to indicate this process of analyzing and rearranging business processes.

Without claiming to be complete, we wish to point out three reasons why business processes can be hard to conceive and comprehend, in order to cast a light on the kind of support that is required for business reengineering:

1. Organizations often embody years of evolution. Over time, an organization has accumulated much tacit knowledge: knowledge that resides in the heads and hands of people, knowledge that is never made explicit and yet is vital to the organization's functioning.

2. Organizations allow for many vantage points from which they can be viewed. In fact, every individual within an organization will have his or her own ideas about how the organization operates.

3. Organizations tend to rapidly become too large for even the most capable manager to have detailed knowledge of the business processes and of the structure, i.e. the people and their formal and informal operating procedures, of their part of the organization, let alone other parts.

In order for people to be able to deal with these complicating properties of organizations, they can resort to models (George 1972; Van Gundy 1981). Hence, the first part of our research question is:
Section 1.3  Narrowing the scope

1. Which modelling techniques can, when applied to the modelling of information intensive organizations, enhance the understanding of an organization for the people who need to design and decide about desired changes?

In addition to such techniques, however, support is required for managing and directing the use of these techniques throughout the change process. Such support could for instance address the creative process of conceiving potential changes, facilitate effective means of communication between the designer(s), management and the work force, support the evaluation of potential courses of action, guide the organization through implementation strategies, and more. Therefore, the second part of our research question addresses this support:

2. Which facilities should be provided by an environment that supports the process of business reengineering in information intensive organizations?

The research described in this thesis is an attempt to come up with answers to these questions. The way in which this research is conducted is explained below.

1.4 Research approach and outline of the study

This chapter has exposed a problem that exists for organizations to meet the requirements posed by a frequently changing environment. Given the demands posed by this environment, organizations should be able to make changes in the technologies and business processes they apply. When the focus is on information intensive organizations, the intangible characteristics of the services they 'make' along with the tacit and dispersed knowledge about their business processes, cause the contrivance, evaluation, planning and implementing of adequate changes to be extremely complicated tasks.

Based on this discerned discrepancy between the need and the possibilities for change, a tentative research question has been stated focusing on possible techniques and on required support for using these techniques throughout the endeavour of analyzing and rearranging an organization — a process for which the term business reengineering was introduced.
To address the research question we start with a literature review on organizations, organizational change and human problem solving, in order to deepen our understanding of the process that is to be supported. Based on our tentative research question and the literature review, a theory is proposed for improving the support for business reengineering.

Figure 1.1 Overview of the research approach
To test this theory, a case research strategy is used. This is primarily prompted by the fact that our subject of interest, the process of organizational adaptation which we have called business reengineering, requires a research approach which does not harm the natural setting. All relevant aspects within an organization which can influence the process of change need to be taken into consideration in order to derive useful research findings. The specific case research strategy we employ involves intensive participation of the researcher in the process analyzed — a research strategy called action research (Susman 1978; Mansell 1991). The advantage of this research strategy is the fact that we will be able to gather an in-depth and first-hand understanding of the change process within actual organizations (Benbasat et al 1987).

To enable case research, specific modelling techniques and a modelling support environment incorporating the essential features of our theory have to be developed. The construction of such techniques and of the support environment is performed in parallel to conducting the first case study. We emphasize that this case study is not used in an inductive fashion for constructing our theory, but rather as a base for testing different options regarding the techniques and support elements. Hence, the first case study is conducted using different versions of the prototype support environment.

Using the techniques and support environment devised, two additional case studies are conducted. In these case studies, the process of business reengineering is followed through, and the results of applying our techniques are collected. Apart from the fact that the prototype support environment was still under development during the first case study, all three case studies are performed in an identical fashion to insure that their results can be considered equivalent. Finally, the findings of the separate case studies are combined, and the general research findings are stated, where the results are evaluated in light of the posed theory.

The research approach described above is depicted in figure 1.1. The different steps distinguished in the approach are addressed by separate chapters.

In chapter 2, the notion of business reengineering is investigated in more detail, primarily based on literature. At the end of the chapter our theory is posed in terms of what Lakatos (1970) would call a 'research programme' for improved support for business reengineering.

In chapter 3, specific modelling techniques are proposed based on the premises of this theory, followed by a description of a modelling support environment in chapter 4. Both the techniques and the support environment are described in their final form. Some comments regarding preceding prototypes of the support environment will be made in chapter 5, where our first case study is discussed.
The case studies that were used to test the usability of the techniques as well as that of the support environment are described in chapters 5, 6 and 7. Finally, the research findings are discussed in chapter 8.
2 Business reengineering: Theories and methods

2.1 Introduction

The need for organizations to increase the frequency and the speed at which they adapt to changes in their environment, and the difficulty for organizations to meet these requirements of adaptability, is illustrated in the previous chapter. For such a process of organizational adaptation, the term business reengineering was introduced, which became the pivot around which our research question was stated.

In this chapter, the notion of business reengineering is examined in light of some theories on organizations and organizational change. The objective is to devise a theory on how to improve the support for the business reengineering process in information intensive organizations — the scope to which we shall confine ourselves.

At the outset of this chapter a problem solving view on the business reengineering process is proposed. By taking such a view as a starting point, the systeological problem (Welke 1977) within a process of change, i.e. the reason why change is required, is emphasized, which offers a fair amount of insurance that the endeavour does not end up solving the wrong problem (Mitroff and Featheringham 1974). As such, a problem solving view has proven to be quite fertile within information systems development (Sol 1982, 1988, 1991; Bots 1989).

After the process of business reengineering, that incorporates a problem solving view, is made clear, existing systems engineering methods are reviewed and evaluated. This evaluation yields a number of guidelines on how to attain more effective support for the process. These guidelines are further explored by reviewing the notion of organization and organization structure, and are translated into a set of modelling concepts and specific requirements for modelling support. Finally, our findings are stated in a more concise research question at the end of this chapter.
2.2 A problem solving view

Although organizations are often characterized as dynamic systems, this is definitely not reflected by their reluctance to change (e.g., Keen 1981). Lewin (1947) and Schein (1961) introduced a model for organizational change reflecting this compulsion of organizations to maintain their current structures. In their model, organizational change is preceded by what they call an 'unfreezing' stage, in which a climate for change is created by establishing an awareness for the need to change. The actual 'change' or 'move' phase, in which the current organization is analyzed and the required change is designed, developed and implemented, is then succeeded by a 'refreezing' stage, in which such a change is institutionalized.

The Lewin and Schein model of organizational change correctly implies that organizations do not by default adapt or evolve over time. Rather, organizational change is triggered explicitly at a certain point in time. In the previous chapter, external events in an organization's environment were indicated as such triggers, invoking changes in an organization. More generally stated, organizations are inclined to adapt when by doing so, a situation that displeases the people in the organization can be resolved. To indicate a trigger for change in organizations, we will use the term business problem, which is, in the line of definitions presented by Ackoff (1974, 1981), Agre (1982) and Sol (1982) defined in the following fashion:

Definition 2.1

A business problem is a situation in an organization which meets the following conditions:

1. people are dissatisfied with the performance of the organization in light of its objectives (or they anticipate that this will be the case in the near future),
2. they have alternative courses of action available, implying changes in the business processes or technologies applied in the organization,
3. the choice made can have a significant effect, and
4. they are in doubt as to which alternative should be selected.

We note that we will not differentiate between problem and opportunity, as we agree with Smith (1989, p. 966) who argues that an "...'opportunity' draws attention to potential goods, instigating thoughtful 'problem solving' activity". Hence, we will use the term 'problem' in reference to both.

The fact that we view a business problem as a trigger and starting point for an organization to change, implies that the process of business reengineering is a problem solving process aimed at resolving this business problem. In order to
investigate this process in further detail, we depart from the general model of the problem solving process presented by Mitroff et al. (1974) as depicted in figure 2.1.

![Diagram](image)

**Figure 2.1** The problem solving process (Taken from Mitroff et al. 1974, p. 48)

In this model of problem solving, four 'phases' are connected by six activities. Starting at the left-hand side of the model, the first phase is a **PROBLEM SITUATION** (I), the recognition of which usually initiates a problem solving process, and which can be *conceptualized* in order to derive a **CONCEPTUAL MODEL** (II). The **CONCEPTUAL MODEL** sets out in broad terms the definition of the particular problem that will be solved. From the **CONCEPTUAL MODEL** a **SCIENTIFIC MODEL** (III), i.e. a formal model of the problem situation presented in terms of the field variables indicated in (II), can be *modelled*. The **SCIENTIFIC MODEL** can be *validated* with the **PROBLEM SITUATION**, but can also, by the activity of *model solving*, be used to derive a **SOLUTION** (IV). This **SOLUTION** can trigger re-conceptualization and model refinement by means of *feedback (in the narrow sense)*, but can also be *implemented* in the **PROBLEM SITUATION**.
The model is general in the sense that it does not impose an ordering of activities, as do the more commonly known phase-models (Lang et al. 1978) of problem solving. A specific instance of a problem solving process can start in any of the four phases, and can, by performing a consecutive set of activities, address one or more other phases of the model.

We elaborate this general model of problem solving in order to make a clear distinction between descriptive models, i.e. models that describe the current organization and the business problem that is perceived, and prescriptive models, i.e. models that represent alternative solutions to the problem. In figure 2.2 the business reengineering process is depicted, showing these two types of models. Within the process, the following modelling activities are distinguished:

1. **Problem specification**

   This activity can be mapped onto a sequence of activities in the process depicted in figure 2.1, which starts at the left-hand side of the model, and follows through the activities of conceptualization, modelling and validation (I→II→III→I). This sequence of activities is performed in order to learn about the business problem at hand, and results in a certain 'level of understanding' or 'yardstick', which is shown to be valid, and from which actual problem solving activities can depart.

2. **Problem diagnosis**

   Where problem specification is aimed at the establishment of a definition of the problem and an assessment of its importance, problem diagnosis goes a step further by trying to derive the forces that cause the perceived problem. When projected in the model in figure 2.1, problem diagnosis can be contained in the specification cycle (I→II→III→I), as well as in the activity of problem solving because insight in the cause of a problem indicates directions for solving it (Mintzberg et al. 1976; Ramakrishna and Brightman 1986; Smith 1989).

3. **Solution finding**

   This activity is called model solving in figure 2.1. The activity is performed in order to get one or more alternative solutions for the perceived problem. The model depicted in figure 2.1 only distinguishes one solution, whereas a number of alternatives will usually be considered. To emphasize these alternative courses of action available for resolving a business problem, the distinction between descriptive and prescriptive models is made. An evaluation of the effect of the alternative solutions is required to differentiate between these alternatives in order to make a well reasoned choice.
4. Implementation and post-evaluation

Once a solution is chosen, it can be implemented in the organization, creating a new reality in which the perceived business problem will probably be solved. In this new situation, the actual effect of the implemented change can be assessed, and hence a cyclic evolutionary business reengineering process can be maintained.

![Diagram](image)

**Figure 2.2** The business reengineering process

The objective of the business reengineering process should not be to design the new business processes, structures and information systems of an organization in one large 'blue print'. The cyclic nature of the model should be used in order to adapt the organization gradually. The process will hence be iterative, switching between analysis, design and evaluation and implementing certain changes before starting a new business reengineering process all together. The effect will be an organization which changes gradually at times, faster at others, and which remains aware of improvements all the time, rather than invoking incidental drastic changes (Quinn 1978; Goldratt and Cox 1984; Meier and Sprague 1991).

In the derived model of the business reengineering process, the use of a model which describes the existing organization is emphasized. By doing so, a thorough understanding of the business problem that needs to be solved can be attained, before
changes in the business processes, structures and the technologies applied in the organization are devised. Such an understanding should not only be achieved by an analyst or designer, but is also very important for the people performing the modelled business processes and the management that is involved. By knowing the causes for the perceived business problem these people will be able to think of effective changes, and they will also be motivated to transform their ideas into required action. As a consequence, the models that are constructed during the business reengineering process should be easy to comprehend by non-modelling experts, and they should induce people to search actively for possible misinterpretations that may have led to modelling errors. In brief, the involvement of the people in the organization should be stimulated.

A second reason for stressing the importance of a descriptive model is the fact that such a model can be used to evaluate alternative proposals for change. By comparing the prescriptive models representing such proposals with the descriptive model, the difference between the current and the proposed organization is made clear. Also, if an assessment can be made of the effect of the proposed changes on the organization, and specifically if the extent to which the business problem is solved can be determined for these changes, choosing between the alternative courses of action will be made easier.

2.3 Problems and human problem solving

Before we turn to the support for the process described above, additional theories on problems and human problem solving are reviewed to acquire a better understanding of the way in which people in organizations address and resolve perceived business problems.

In literature regarding problem solving, there is a well known distinction between structured and ill-structured problems (e.g., Reitman 1964), although different terminology is often used, and the distinction may have quite different interpretations. One of the more common definitions is that a problem is structured if the problem itself, or its look-alike, is known beforehand, and the organization already has a procedure to respond to it (Simon 1973b; Mintzberg et al. 1976). Others define a problem to be structured if it can be precisely formulated, and handed over to an expert who can solve it without requiring any further information (Sølvberg 1975; Checkland 1981). We prefer to define 'structuredness' in terms of attributes of the problem itself and its solutions (Sol 1982; Young 1984).
We will use the definition given by Sol, who states that a structured problem should meet the following conditions:

1. the set of alternative courses of action or solutions is finite and limited,
2. the solutions are consistently derived from a model of the problem situation that shows a good correspondence with reality, and
3. the effectiveness or the efficiency of the courses of action can be numerically evaluated.

We agree with Bots (1989) who states that a problem is not ill-structured or structured regardless of its context, as problem solving activities for an initially ill-structured problem may transform this into a less ill-structured or even a structured problem. We can state that, at the outset, business problems will most always be ill-structured, as they do not meet one, or possibly even any of the requirements stated above for a structured problem. In most cases (1) a seemingly endless variety of solutions exists, (2) a valid model of the current organization is not available, or (3) the impact of different courses of action can only be estimated, using an educated guess.

The initial ill-structuredness of business problems can be ascribed to the fact that the analyst is not sure about certain aspects of the problem situation. This may be caused by lack of information, in which case the term 'uncertainty' is appropriate (Galbraith 1977), but it may also be caused by ambiguity, i.e. the existence of multiple and conflicting interpretations about the problem situation, in which case the term 'equivocality' is more appropriate (Weick 1969, 1979).

The way in which the two sources of doubt are resolved are quite different. Uncertainty can be resolved by attaining more information: as information increases, uncertainty decreases. In contrast however, equivocality implies the analyst does not know which questions to ask, rather than not having the answers, and hence new data or information may not resolve this (Daft and Macintosh 1981). Equivocality is reduced by defining or creating an answer, rather than by learning an answer from the collection of additional data (Daft and Lengel 1986).

For resolving equivocality, it should be possible to pose different, perhaps even contradicting models of the problem situation, and to establish a better understanding from an analysis of these different points of view (Mason 1969; Mitroff and Betz 1972). Also, the notion of 'learning models' as proposed by Churchman (1971) serves such a purpose.

Apart from problem structure as a determinant factor, an assessment of the way in which people address and resolve problems can yield further requirements for effective problem solving support. One of the most important contributions in this
respect has been the critique of economic theories on decision making by Simon (1947). Two assumptions that are made in these economic theories are that decision makers are assumed (a) to be completely informed, and (b) to maximize something. This mode of decision making is what Simon calls *objective rational*.

In order to derive at a model of human decision making, which acknowledges the cognitive limitations of decision makers, Simon (1957) introduces the principle of *bounded rationality*. In the first place, a decision maker does not have or can not deal with all the relevant information, and hence constructs a simplified model of reality. He behaves rational only with respect to this model — only in terms of his limited knowledge of alternatives and consequences. Apart from limitations in knowledge, the decision maker also has a limited amount of time and energy available for decision making. Instead of viewing a decision maker as a person who tries to *maximize* some outcome, it is argued that decision making should be seen as a *satisficing* activity, where the decision maker will search for (at least) one alternative which fulfills some minimum standard of satisfaction, rather than investigating all the imaginable alternatives and all their possible outcomes (Simon 1956). This does not imply that a person will choose a satisfying alternative if better ones are at hand, but it does limit the amount of time and energy spent searching. Also, the ease or difficulty of finding satisfying solutions will effect the level of aspiration, i.e. the minimum standard of satisfaction, for similar problems in the future (Child and Whiting 1949).

We note that Simon replaced the term objective rational by substantive rational and bounded rational by procedural rational in his later work (e.g., Simon 1969, 1977), but we prefer to use the terminology of his earlier work, as this emphasises cognitive limitations causing the later decision making mode.

In supporting decision making within the business reengineering process, we emphasize the importance of acknowledging the cognitive and time limitations of decision makers (Mintzberg 1975, 1976), which poses quite different requirements. Therefore, bounded rational decision making is chosen as a point of departure, from which possible problem solving support can be considered.

### 2.4 Lessons learned from existing methods

Now that a problem solving view on the business reengineering process and some characteristics of human problem solving are discussed, we can turn to the support for this process in information intensive organizations. In this section we review existing information systems development and systems engineering methods and establish some guidelines on how effective support can be attained.
Section 2.4 Lessons learned from existing methods

Information systems development methods provide assistance for designers who plan, analyze and design information systems (Olle et al 1988). Such methods contain a set of working procedures that can be followed and a set of modelling techniques that can be used during the development process. As a consequence of the need for high quality information systems and for a productivity increase in the development process (Sol 1985), and the great number of attempts that have been taken to meet these needs, many different development methods have evolved (Bubenko 1986).

Although the activities and modelling techniques offered by these methods as well as their underlying philosophies can be quite different (Hirshheim and Klein 1989; Walsham 1991), the methods are similar in their attempt to describe a so-called object system (Langefors 1963, 1978) with a set of predefined terms. The 'Structured Analysis' method (DeMarco 1979; Yourdon 1989), for instance, uses four basic concepts (i.e., data flow, data store, external entity and process) to model an organization. Similar methods are ISAC (Lundeberg 1982), Jackson's method (Jackson 1983) and Information Engineering (Macdonald 1986). Similar sets of concepts are provided by general systems engineering methods, such as SADT (Ross et al 1977).

By applying methods like these, a model of an organization can be constructed which consists of a set of interrelated diagrams, each representing certain aspects of the modelled organization. With such diagrams, the existing organization as well as proposed changes can be described.

Two problems can be identified for such a set of diagrams:

1. The set of modelling concepts used may not fit the mental models of the people in the organization. A discrepancy between the modelling concepts employed in these methods and the concepts familiar to the people in the organization forces people to spend significant effort in transforming their views of a target system to a description constrained by the syntax of a certain method. As a consequence, the involvement of these people in the process can be greatly diminished (Bostrom and Heinen 1977a,b; Land and Hirschheim 1983; Guinen and Bostrom 1986).

2. Each diagram provides a partial view of the model. Validation of the model as a whole is difficult due to this dispersed nature. Also, it is hard to envision designed changes in the organization if these changes are only communicated using such a set of diagrams. In brief, the recognizability of the models leaves much to be desired (Kung and Sølvberg 1986).
To elaborate the second problem, it is interesting to view the support which is offered by computer based CASE tools for many development methods. The fact that interrelated diagrams are used which need to be kept consistent implies that such computer based tools are a prerequisite for effectively using the development methods (Martin 1986; Yourdon 1986). An area that is neglected by existing CASE tools, however, is the support for clarifying a design or its implications to the participants in the organization. Instead, these CASE tools are primarily interested in the activities of model construction and model verification according to the rules set up for the modelling technique(s) applied (Vonk 1988a,b; Chen et al 1989).

To our opinion, effective support for design activities should not be directed to deriving designs that satisfy the requirements of consistency and completeness according to formal rules of the techniques applied. Rather, it should help to envision the impact of a design on the organization’s business processes and its bottom line (Marsden and Pingry 1988), or better yet, determine the impact on what Keen (1991) calls the organization’s top line, i.e. the kind of services offered to client’s and the way in which these services can best be provided. An example of a direction by which such support can be realized is set out by Bell (1988), Dachouffe and Lesuisse (1989), Mosser et al. (1989) and Wierda (1991) who regard visualization and 'hands-on' model manipulation as key prerequisites for enhancement of the support offered by design tools in the preliminary stages of a design process.

Shortcomings similar to the ones indicated above can be found in a separate category of methods which serve as a precursor to the information systems development methods, such as Business System Planning (Lederer and Putnam 1987) and Critical Success Factors (Rockart 1982). Initiated by the framework presented by McFarlan, Nolan and Norton (1973), these methods support an organization in attaining directions for information systems development by departing from the organization’s objectives and strategies. Recently, new methods have been developed in this area which are based on the ideas of Porter (1980, 1985) regarding competitive strategy. By using Porter’s strategic targets and strategic thrusts these methods are able to focus managerial decision making on important issues at an early stage (Parsons 1983; Wiseman and MacMillan 1984; Porter and Millar 1985; McFarlan 1984; Ives and Learmonth 1984).

The strength of such precursory methods is the fact that they focus attention on the purpose for which information systems are developed. The objectives of an organization are the starting point from which ideas on how to use information technology to attain these objectives can arise. The disadvantage of such methods is, however, that these methods only present a one-dimensional subdivision or a two-dimensional grid of subjects and considerations to which information technology can direct its impact (Lee and Adams 1990; Bergeron et al. 1991). As such, these methods do help to set an agenda of subjects that need to be addressed, but they do not extend their support to the actual contrivance and evaluation of applications of
information technology that should evolve in the course of addressing such an agenda. The assessment of the actual contribution of a specific system to the firm's success is left up to management's insight (Marsden and Pingry 1988), or is determined after the system is put to use (Lincoln 1986; Strassmann 1988; Lee and Adams 1990). Any support for analyzing the consequences of different courses of action is lacking, and hence choosing between alternatives is left up to management's insight as well.

In short, our review of existing methods and tools has pointed out two important areas for improvement.

The first deals with the establishment of adequate participation in the process from people who are not specifically trained in certain modelling techniques. The use of recognizable concepts is proposed as a way to attain such participation. Also, specific support for model validation which compensates the dispersed nature of the models is discerned to be important.

The second area which is considered important is the support for the actual solution finding activity. In this respect, a lack of support for evaluating the effect of solution alternatives is pointed out above.

Although our experience and expertise largely stem from the information systems field, the indicated lack of support is also apparent outside this field. In an empirical study of innovation of products and processes in organizations, for instance, Parker found that "...action was rarely taken to improve the ability of staff to recognize opportunities or to solve problems" (1982, p. 213). Also in this area support for stimulating imaginative thought on changes that can help the organization improve its performance is lacking.

2.5 Towards effective support

In this section, the general guidelines for devising effective support for the business reengineering process as indicated in the previous section, are elaborated. First a set of concepts that can be used to model organizations is determined by reviewing basic organization literature. Next, a framework for the evaluation of organizational performance is discussed, and a proposal for evaluating organizational changes is derived from this.

Organizations are complex systems in which people cooperate in order to attain a goal for a longer period of time, which they could not have achieved individually (Kieser and Kubicek 1983). Hence, in the study of organizations, two considerations are important: differentiation by which an overall task that is too large to be addressed by a single person is subdivided into several subtasks that can each be
assigned to different people, and *integration* by which individual actions are combined in order to achieve successful completion of a task as a whole (Lawrence and Lorsch 1967; Kieser and Kubicek 1983; Child 1977). These considerations have effected theories on organization design to a large extent. To quote Mintzberg (1979): "Every organized human activity — from the making of pots to the placing of a man on the moon — gives rise to two fundamental and opposing requirements: the division of labour into various tasks to be performed, and the coordination of these tasks to accomplish the activity."

The considerations of differentiation and integration are interrelated, as the choice for a certain division of labour imposes requirements for coordination. Galbraith (1973, 1974) indicated this trade-off between the two considerations, by introducing the notion of 'uncertainty'. This is defined to be the difference between the information required to perform a certain task, and the information that is available. The extent to which people can perform tasks realizing the goal set for them, is dependent on the amount of task uncertainty. Hence, coordination is required in order to minimize the task uncertainty for individuals, given a subdivision of tasks amongst them (Mintzberg 1979, 1988). As Strassmann (1988, p. 16) puts it, "The required information processing is the outcome, not the cause of organization design".

Based on the above, we propose a set of concepts to model organizations. The first concept relates to the work that is performed in the organization, and, due to our focus on information intensive organizations, is called *information processing task*. The second concept relates to the individuals and departments which perform the required work, and, again due to our domain, is called *information worker* (Strassmann 1988). The information processing tasks that are assigned to information workers represent the division of labour within the organization. The way in which these tasks performed by individuals are combined represents the coordination. Finally, a third concept is proposed to represent the object that is processes, and is called *information item*. When all the required information processing tasks for an information item have been performed, this item represents the product devised by the organization.

Now that a set of concepts is derived, a way to determine the effects of a change in an organization needs to be made clear in order to be able to provide support for the evaluation of solution alternatives.

Bots and Sol (1988) distinguish three different perspectives from which organizational performance can be reviewed: the *macro* perspective, the *meso* perspective and the *micro* perspective.

The *macro* perspective encompasses the organization as well as the entities within its environment, and focuses on improvements in the overall 'production chain' (Wierda 1991; Streng and Sol 1992). Adjustments that are made at this level,
impose requirements for the internal organization which can in turn be inspected from the other two perspectives.

The *meso* perspective is concerned with the business processes within an organization, and improvements at this level include those which are related to these processes and are often stated in terms of characteristics of the product made within these processes. As a consequence, from the meso perspective the focus is on the coordination of different information workers active within the same business process.

The *micro* perspective is concerned with the workplace of an individual information worker in the organization, and improvements at this level aim for an increased performance of such an individual, possibly attained by the application of information technology (Sprague 1986).

Within the business reengineering process, all three perspectives are important. As the need for organizational change often originates from external causes in the environment, the objective of the process can be established at the *macro* level. To realize these objectives, certain changes can be made to the business processes. These changes are made clear from the *meso* perspective. Also, improvements can be attained by providing a better support to individuals at their work place. These improvements are observed from the *micro* perspective.

The three perspectives can not be regarded in isolation from one another. The effect of changes that are made at an individual's work place (micro level) need to be determined for the business processes in which this individual participates (meso level). Also, a specific arrangement of tasks at the meso level, imposes working conditions at the micro level which also need to be taken into consideration. Finally, the improvements that are obtained, viewed from the micro and the meso level, need to be made clear at the macro level because the improvements are not a purpose in themselves, but are directed to solving business problems identified at the macro level. This last step is what Keen (1988) would call 'making the business case'.

The interrelated characteristics of the three distinguished perspectives, implies that special support should be available for determining the effect of solution alternatives for the organization. We believe that static modelling techniques can hardly be used to determine these effects. We give two examples to demonstrate our point:

1. If more adequate support is provided at an individual's workplace, the effect on the overall business process can only be determined, if this business process is followed through time taking the changed support into account. If the task performed by the individual within the process is not a critical one, the effect on the overall process might be negligible.
2. If a change is made to a business process, the effect on an individuals work load can hardly be determined based on the new process descriptions. Considerations like the number of items processed, and the quality and time constraints imposed, need to be taken into account in order to be able to determine the effect for the individual. Again, timing aspects play an important role.

To provide support for the evaluation of alternative solutions, modelling techniques should be used which facilitate the simulation of business processes over time. We will use the term *dynamic modelling* (Sol and Van Hee 1991) to indicate such techniques.

### 2.6 Refining the research question

In the previous sections, theories on organizations, organization structure and problem solving have been used to enhance our understanding of the notion of *business reengineering*. A problem solving view on the business reengineering process was posed, where a so-called *business problem* was seen as the trigger, implying that the focus of the process should be on resolving such a problem.

An evaluation of existing information systems development and general systems engineering methods yielded two important areas for improved support, which were elaborated in the previous section.

In this section our findings are combined in order to derive what Lakatos (1970) would call a 'research programme' for supporting the business reengineering process. Such a programme consists of a set of presuppositions, called the 'hard core', which constitute a theory which is not questioned in the research. This point of departure can not be rejected, regardless what the outcome of the research will be. In addition to this 'hard core', a research programme also contains a set of hypotheses, which are subject to doubt. These hypotheses are called the 'protective belt', as negative research findings will call for rejection of these additional hypotheses, rather than that of the 'hard core' itself.

The 'hard core' for our research programme is as follows:

1. Organizations are systems that have a purpose which is externally defined. They need to induce change when external forces endanger the fulfilment of their objectives.
2. The impact of changes in organizations can be reviewed from the micro perspective, taking the view point of an individual’s work place within the organization, from the meso perspective, showing the coordination of individual action within business processes in the organization, and from the macro perspective, focusing on the organization interacting with its environment.

3. Information technology can be applied in organizations to offer task specific support to individuals to obtain improvements viewed from the micro perspective. Information technology can also be used to establish lateral and vertical communication channels or to impose certain forms of coordination to obtain improvements viewed from the meso perspective (within the organization) or the macro perspective (between organizations).

4. Support for decision making in organizations, and hence the support for the business reengineering process, should be based on the concept of bounded rationality.

Using this 'hard core' as our point of departure, we pose a theory for improving the support for the business reengineering process. This theory is projected by the following 'protective belt':

1. By providing comprehensible models of an organization’s business processes and by facilitating a mechanism to evaluate the impact of changes to the business processes from the micro, meso and macro perspective, adequate support for the business reengineering process in information intensive organizations is obtained.

2. Comprehensible models of information intensive organizations can be devised by using three concepts which are easy to use for non-modelling experts: information worker, information processing task and information item.

3. The effect of changes in information intensive organizations can be made clear from the micro, meso and macro perspective if the behavior of the business processes can be exposed, i.e. if these business processes can be simulated over time.

4. The modelling approach envisioned by the preceding statements can be adequately supported by an environment comprised of support tools which are constructed using information technology available today.
It is clear that, despite the fact that the basic ideas for improving the support for business reengineering have been portrayed, specific techniques and support will need to be constructed before these ideas can be tested. Therefore, the following two chapters will propose such specific techniques and a specific support environment, which will then be used in empirical studies in order to test the viability of our theory.
3 A dynamic modelling approach

3.1 Introduction

In the previous chapter several propositions were made for improving the support for business reengineering in information intensive organizations. In this chapter a specific approach for modelling organizations is posed which incorporates these propositions and which enables the testing of our theory.

The objective is to establish a model of an organization which not only represents it in a static way, but which can also be used to analyze its dynamic aspects, i.e. its behavior. In the presentation of our approach, we will first discuss static modelling techniques for both the entities and the business processes of an organization. Following these, a special mechanism is introduced to simulate the concurrent activities of the people in an organization over time. By applying this mechanism to the static representation of the organization, its behavior can be analyzed.

3.2 Entity modelling

In the previous chapter, a view on information intensive organizations has been derived which encompasses three concepts: information workers perform information processing tasks in order to process information items. In this section these concepts are transformed into a set of entity types or classes (Hoare 1972; Smith and Smith 1977) which can be used to model organizations. The objective is to attain a set of generic or super classes that provide a vocabulary for describing organizations. For a specific organization, specializations of these super classes can be made (Sciore 1989), and specific entities that exist in the organization can be represented by instances of such specialized classes. In figure 3.1 these notions are made clear graphically.
The modelling concepts derived in the previous chapter make a distinction between active entities in an organization, i.e. information workers, and passive entities, i.e. information items. Such a distinction is quite common in modelling approaches (e.g., Brodie and Silva 1982; Bracchi and Pernici 1984; Kaye and Karam 1987) and will be represented by two distinct super classes. Passive entities are classified into a super class named item, and are characterized by attributes only, whereas active entities are classified into a super class named actor and are characterized by attributes as well as skills. The skills that are assigned to Actors represent the information processing tasks of which they are capable. Hence, both the static aspects, covered by the attributes, as well as the behavioral aspects, represented by the skills, are covered by a single description format.

In short, the two super classes we distinguish are:

```
super class simple actor
  attributes
  skills
end super class

super class simple item
  attributes
end super class
```

The reason why the prefix simple is used here will be made clear shortly, when the compound equivalents of these classes are introduced.

Entity classes that are defined to model a specific organization will inherit the structure of one of the super classes stated above. To specify such entity classes we use a very general language which does not distract ones attention to considerations of syntax. The specification format we use is the following:
class Name ← name of super class
attributes
  first attribute
  :
  last attribute
skills
  first skill
  :
  last skill
end class

The '←' symbol is used to specify that the entity class is a specialization of the super class mentioned behind the '←' symbol. Examples of class specifications are given below, using this notation which we apply throughout the next chapters:

class Person ← simple actor
attributes
  Name
skills
  Write a book
end class

class Book ← simple item
attributes
  Title
  Author
  ISBN Number
end class

With the two super classes distinguished so far, single active and passive entities in organizations can be modelled. In many cases, however, it is required that such single entities can be aggregated into compound entities. Examples of such compound entities are a department which consists of several people, and a dossier which encompasses different information forms. To model such compound entities, two additional super classes are introduced:

super class compound actor
attributes
  constituents
end super class

super class compound item
attributes
  constituents
end super class

Analogous to the specification of simple classes, the attributes of a compound class are defined at the class level. Constituents, however, are not specified at the class level, but rather at the level of instances. Once an instance of a compound class is created, entity instances can be added to and removed from it. The only restriction is that a compound actor can only contain instances of actor classes, and a compound item can only contain instances of item classes.
Some examples of compound entity classes are:

```object
class Team <- compound actor
  attributes
  Name
  Competition ranking
end class

class Dossier <- compound item
  attributes
  Identification number
  Subject
end class
```

Obviously, we could have specified constituent relationships at the class level, rather than at the instance level. In the example, we might have specified that the class Team should have exactly one constituent of class Coach and at least two constituents of class Player. This would however restrict our specification possibilities at the instance level, which would reduce its flexibility. For this reason, the specification of constituents at the instance level is preferred. Figure 3.2 illustrates the way in which compound entities are specified.

---

**Figure 3.2** Constituents of a compound actor specified at the instance level

30
In short, four super classes are described for modelling entities in organizations. By using specialization, instantiation and (dis)aggregation, these classes can be used to model the active and passive entities of an organization. We wish to point out some favourable aspects of this line of modelling:

1. Static and behavioral aspects of real world phenomena are specified in a single description format. This combination is commonly known in object oriented modelling approaches which are based on the work of Dahl and Nygaard (1966), Quillian (1968), Raphael (1968) and Weinreb and Moon (1981). By employing an object oriented modelling approach one is able to think of a real world phenomenon in terms of what it is (attributes) and in terms of what it does (skills) although these aspects are not separated (Bots 1989). The advantages of such an approach have been well recognized in the information systems field (e.g., Sol 1982; Woo and Lochofsky 1986; McIntyre and Higgins 1988; Tsichritzis 1989).

2. Although exceptions to the rule can be conceived, the distinction between entity classes and entity instances basically provides a separation of the conceptual level and the empirical level.

3. The incorporation of skills into the description of an actor facilitates an encapsulation of its description for other entities in the model (Lockemann 1989; Nierstrasz 1989). By doing so, the actor’s behavior can be determined by the actor itself, where other entities can do no more than 'request' this actor to perform one of its skills. Hence, autonomous behavior of actors in organizations can adequately be represented (Dur and Bots 1992a).

These considerations make that the line of modelling proposed offers a comprehensible and flexible representation of an organization’s entities. The one-to-one relationship between real world phenomena and their model equivalents enhance the comprehensibility of a model, especially for those who are not specifically trained in modelling activities.

Before we turn to the discussion of business process modelling we conclude this section with a definition of what we will call an entity model of an organization:
Definition 3.1

An entity model $E_O$ of an organization $O$ is a 6-tuple
$E_O = (CA_O, CIA, IAO, HO, CO, NO)$, where:

- $CA_O = \{ca_1, ..., ca_n\}$ is the collection of all entity classes for $O$
  which are specializations of the super class simple or compound
  actor,

- $CIA = \{ci_1, ..., ci_m\}$ is the collection of all entity classes for $O$
  which are specializations of the super class simple or compound
  item,

- $IAO = \{i_1, ..., i_k\}$ is the collection of all entity instances of classes
  contained in $CA_O$, i.e. the collection of actors,

- $HO = \{i_1, ..., i_l\}$ is the collection of all entity instances of classes
  contained in $CIA$, i.e. the collection of items.

The set $IO$ is used to denote all the entity instances, i.e. $IO = IAO \cup HO$.

- $CO: IO \rightarrow CA_O \cup CIA$ is the function specifying the class of an entity
  instance, and

- $NO \subseteq IO \times IO$ defines the constituent relationship.

The predicate class denotes function $CO$, where $class(i) = c$ implies that
$i \in IO$ is an instance of class $c \in CA_O \cup CIA$.

The predicate constituent denotes the constituent relationship $NO$, where
constituent($i_1, i_2$) implies that $i_2$ is a constituent of $i_1$, with $i_1, i_2 \in IO$.

Before we can state the properties for an entity model, an extra predicate is
required:

Definition 3.2

An entity is encompassed by another entity if it is a constituent of that
entity or any of its constituents:

$encompass \subseteq IO \times IO$

$encompass(i_1, i_2) \equiv constituent(i_1, i_2) \lor$

$\exists i_3 \in IO \ [constituent(i_3, i_2) \land encompass(i_1, i_3)]$
Using this extra predicate we can state the properties that are valid for a correct entity model:

**Property 3.1**

An entity which is a constituent of an actor, must itself also be an actor:

\[ \forall i_1, i_2 \in I_o \ [\text{class}(i_1) \in CA_o \land \text{constituent}(i_1, i_2) \Rightarrow \text{class}(i_2) \in CA_o] \]

An entity which is a constituent of an item, must itself also be an item:

\[ \forall i_1, i_2 \in I_o \ [\text{class}(i_1) \in CI_o \land \text{constituent}(i_1, i_2) \Rightarrow \text{class}(i_2) \in CI_o] \]

**Property 3.2**

The constituent relationship \(N_o\) may not contain cycles. Hence, if an entity is encompassed by another entity, this entity can never encompass that entity:

\[ \forall i_1, i_2 \in I_o \ [\text{encompass}(i_1, i_2) \Rightarrow \neg\text{encompass}(i_2, i_1)] \]

This concludes our presentation of entity modelling. In the next section, the modelling of business processes will be discussed.

### 3.3 Business process modelling

The entity model presented in the previous section includes, among others, the super class *simple actor* which is used to model information workers in organizations. The skills of such an actor represent the information processing tasks the actor can perform. In the entity model, however, these actors are represented in total isolation from one another, whereas coordination is required in order for these actors to establish one of the organization’s business processes as a whole. Therefore, the sequencing of tasks performed on the same information item(s), perhaps by more than one actor, needs to be made explicit. In this section, a technique for representing such sequencing and coordination is presented.

In the technique which is used in our approach, the central notion is that of a *task*. After Bots (1989), we consider a task to be some activity or combination of activities which is performed on some item in order to achieve a certain goal.
Within the technique, two operations are important:

1. **Decomposition**

   A task can be defined recursively in terms of subtasks. Such subtasks pertain to a part of the activities required for the task as a whole, and are directed to attaining a subgoal. The task as a whole is completed once all of its subgoals have been attained. A task which is decomposed into subtasks will be called a *compound task*, whereas one which is not decomposed is called a *simple task*.

2. **Sequencing**

   The subtasks that constitute a task as a whole are performed in a certain order. This order of subtask execution should be made explicit, implying that sequencing, parallelism and synchronization ought to be represented.

   To represent both decomposition and sequencing, so-called *task structures* are used. Each compound task has a task structure in which its subtasks as well as their sequencing are made explicit.

   A task structure consists of *components* which denote the subtasks, and *transitions* which represent the sequencing of these subtasks. A component at which a given structure starts is called an *initial component*. In a task structure, more than one initial component may exist, implying that parallel tracks of subtask execution may evolve.

   Besides indicating a subtask, each component can also specify the actor whom is to perform that subtask. Within the same task structure, different components can relate to different actors, and, hence, the required coordination of these actors is captured in such a structure. Those components for which no actor is specified are performed by the actor to whom the task as a whole is assigned, in order to preclude ambiguity in the specification.

   For a transition a condition can be specified, implying that the transition is only valid if this condition is met. Such a condition can be an expression which evaluates to a true or false value. In such an expression, attribute values of the item that is being processed, as well as those of the actor that processes it, can be included.

As an example, we will model the process according to which a cheque is dealt with in a financial firm. The overall task of treating a client’s check will involve subtasks like inspecting the client’s balance, transferring the money from one account to the other, and printing and sending the balance reports. The compound task of treating a client’s check is Accounting’s responsibility. Apart from Accounting, however, *Operations* and the *Relation manager* are also involved. For this example, the following entity classes are specified:
class Check ← simple item
  attributes
  Amount
  Cleared
...
end class

class Accounting ← simple actor
  attributes
  skills
  treat client's check
end class

class Operations ← simple actor
  attributes
  skills
  print and send balance report
end class

class Relation manager ← simple actor
  attributes
  skills
  send check back to client
end class

Figure 3.3 shows the task structure for the task 'treat a client's check'. In the graphical representation, components are depicted as rectangles. These rectangles are shadowed if the component represents a compound task which is itself specified in further detail (e.g., the task 'transfer money' in figure 3.3).

Figure 3.3 Graphical representation of task structures
Initial components are emphasized by using bold lines. In each component, the name of the actor is stated at the top, above the name of the task referred to by that component. We call to mind that the components for which no actor is specified are performed by the actor to whom the compound task as a whole is assigned. Transitions are depicted as arrows. Finally, possible conditions are written alongside the arrow of a transition.

In the figure it is made clear that different components within the same task structure can refer to the same subtask (e.g., 'check client’s balance'). Although the task to which such components refer is the same, the transitions departing from the component may differ.

The example serves to show how the specification of individual actors, each with their own skills, are complemented by task structures making clear the way in which these skills are combined to establish a business process as a whole.

Having introduced task structures by means of an example, we will now define them formally. We will start out with the definition and properties of the so-called functional model of an organization. This is the model which consists of all the tasks, simple and compound, that have been identified for a given organization. Following this, the definitions of task and task structure are given, and some additional properties are defined.

**Definition 3.3**

A functional model $\mathcal{F}_O$ of an organization $O$ is a 4-tuple $\mathcal{F}_O = (T_O, H_O, S_O, F_O)$, where:

- $T_O = \{t_1, ..., t_n\}$ is the collection of all tasks modelled for $O$,
- $H_O \subseteq T_O \times T_O$ denotes the 'task hierarchy' for $T_O$,
- $S_O$ is a set of 'task structures', and
- $F_O : T_O \rightarrow S_O \cup \{\Lambda\}$ is the function assigning task structures to tasks, where $\Lambda$ denotes the empty task structure which is assigned to simple tasks.

The predicate subtask will be used to denote the task hierarchy relation $H_O$, where $\text{subtask}(t_i, t_j)$ means that $t_i$ is part of the decomposition of $t_j$, i.e. $(t_i, t_j) \in H_O$.

The predicate structure denotes function $F_O$, where $\text{structure}(t) = s$ implies that $s \in S_O$ is the task structure assigned to task $t \in T_O$. 

36
We also define an additional predicate which is used to specify the properties of a functional model.

**Definition 3.4**

A task is a *descendant* of another task if it is part of the decomposition of that task, or of any of its subtasks. The predicate *ancestor* will be used to denote the inverse relation.

\[
\text{descendant} \subseteq T_o \times T_o \\
\text{descendant}(t_1, t_2) \equiv \text{subtask}(t_1, t_2) \lor \\
\exists t_3 \in T_o \ [\text{subtask}(t_3, t_2) \land \text{descendant}(t_1, t_3)]
\]

Using this additional predicate, the properties that should be valid for a correct functional model $F_o$ can be defined.

**Property 3.3**

The task hierarchy $H_o$ may not contain cycles. Hence, if a task is a descendant of another task, it can never be an ancestor of that same task.

\[
\forall t_1, t_2 \in T_o \ [\text{descendant}(t_1, t_2) \Rightarrow \neg \text{ancestor}(t_1, t_2)]
\]

**Property 3.4**

The function $F_o$ does *not* have to be injective, hence, different tasks can be mapped onto the same structure.

For given $t_1, t_2 \in T_o$: $\text{structure}(t_1) = \text{structure}(t_2) \Rightarrow t_1 \neq t_2$

**Property 3.5**

A decomposition of a task should at least contain two subtasks.

\[
\forall t_1, t_2 \in T_o \ [\text{subtask}(t_1, t_2) \Rightarrow \exists t_3 \in T_o \setminus \{t_1\} \ [\text{subtask}(t_3, t_2)]]
\]

Although this last constraint could be abandoned without consequences for our formal definition, we impose it to emphasize the fact that decomposition should be used to acquire a description at a more detailed level, rather than as a rename mechanism amongst tasks.
The main concepts used in the functional model are *task* and *task structure*. We define both of these concepts below. Before we do so, however, we call to mind that the sets $CA_o$ and $IA_o$, which were introduced in the previous section, denote the sets of actor classes and instances respectively for organization $O$. Similarly, the sets $CI_o$ and $II_o$ denote the set of item classes and instances.

As explained earlier a task represents an activity which is performed on an item in order to attain a certain goal. Formally we will represent a task by the following 4-tuple:

**Definition 3.5**

A task $t$ is a 4-tuple $t = (i, O, d, o)$, where:

- $i \in CI_o$ is the class of the item that is processed, i.e. the item that is the *input* to task $t$,
- $O \subseteq CI_o$ is the set of classes which are the *output* of task $t$,
- $d : II_o \times IA_o \rightarrow \mathbb{R}$ is the function which specifies the *delay* associated with task $t$, and
- $o$ denotes the *operation* associated with task $t$.

The predicate *delay* is used to denote $d$, where *delay*$(i, a)$ returns a real value which is a sample of this delay function, where $i \in II_o$ with $class(i) = c$, and $a \in IA_o$.

The input $i$, and the output $O$, are used to represent the type of item that is processed by task $t$ and the type of item(s) that is made available after task $t$ has been finished. In most cases, the input and output class will be identical, where only the attribute values of the item that entered the task are altered.

The delay function of a task is used to represent the amount of time that an actor will be occupied when performing the task. As can be seen in the definition, this delay may depend on the attribute values of the item that is processed as well as the attribute values of the actor performing the task. As an example, the delay function for the task 'check client’s balance' in figure 3.3 might be:

$$duration \leftarrow normal(Amount/1000 \text{ min}, 1 \text{ min})$$

implying that the duration of this task is a normal distributed value with a mean that depends on the amount of money transferred and a standard deviation of 1 minute.

By performing a task the attribute values of the item that is processed can be altered. This may also be the case for the attribute values of the actor performing the task. These operations on the item and actor are specified by $o$. As part of the operation $o$, new items may also be created by the actor. Again we will use the task
'check client's balance' as an example. The operation performed on a check by this task might be:

\[ \text{Cleared} \leftarrow \text{random} \geq 0.3 \]

where \( \leftarrow \) is an assignment and \text{random} is a function that returns a random value between 0 and 1. The attribute values assigned in an operation of a task are in most cases used in the conditions for transitions leaving that task, as is the case for the attribute 'Cleared' in figure 3.3.

The definition of task as described above pertains solely to simple tasks. In the case of a compound task, both the delay and the operation are determined by its subtasks. Hence, for these compound tasks the concept of \textit{task structure} still needs to be defined.

\[ \text{Definition 3.6} \]

A \textit{task structure} \( S \in S_0 \) is a 6-tuple \( S = (C_5, I_5, S_5, A_5, D_5, T_5) \), where:

- \( C_5 \) is a set of components,
- \( I_5 \subseteq C_5 \) is the set of initial components,
- \( S_5: C_5 \rightarrow T_5 \) is a function relating components to their corresponding tasks,
- \( A_5: C_5 \rightarrow IA_5 \cup \{\Lambda\} \) is a function relating components to their corresponding actors, where \( \Lambda \) is the symbol for no actor,
- \( D_5 \) is a set of conditions, and
- \( T_5 \subseteq C_5 \times C_5 \times D_5 \cup \{\text{true}\} \times C_5 \) is a relation defining the transitions that exist between different components.

The predicate \textit{task} will be used to represent function \( S_5 \), where \( \text{task}(c) = t \) implies that component \( c \) refers to task \( t \).

Similarly, the predicate \textit{actor} is used to represent function \( A_5 \), where \( \text{actor}(c) = a \) implies that component \( c \) refers to actor \( a \).

The predicate \textit{trigger} will be used to represent relation \( T_5 \), where \( \text{trigger}(c_1, c_2, d, cl) \) implies that component \( c_2 \) is triggered by component \( c_1 \) under condition \( d \), where an item which is an instance of class \( cl \) is passed on to the next component.
We will define the properties that exist for a correct task structure \( S \) after we have defined an auxiliary predicate.

**Definition 3.7**

Two components are connected if a path of triggers exists between them.

\[
\text{connected} \subseteq C_s \times C_s \\
\text{connected}(c_1, c_2) \equiv \text{trigger}(c_1, c_2) \lor \exists c_3 \in C_s [\text{trigger}(c_1, c_3) \land \text{connected}(c_3, c_2)]
\]

The first properties that we define, establish a relation between the task hierarchy, and the components of a task structure.

**Property 3.6**

All the subtasks of a task \( t \) should be referenced by at least one component of \( \text{structure}(t) \).

\[
\forall t \in T_0 \ [\text{subtask}(t_1, t) \Rightarrow \exists c \in C_{\text{structure}(t)} \mid \text{task}(c) = t_1]
\]

**Property 3.7**

Every component of the structure of a task \( t \) should reference a subtask of \( t \).

\[
\forall c \in C_{\text{structure}(t)} \ [\text{subtask}(\text{task}(c), t)]
\]

Hence, in a structure for a compound task \( t \), every one of its subtasks is referenced at least once, and no tasks are referenced that are not subtasks of \( t \). Besides these properties, task structures should meet an additional constraint:

**Property 3.8**

Every component should either be initial, or should be connected with an initial component.

\[
\forall t \in T_0 \ \forall c \in C_{\text{structure}(t)} \ [c \in I_{\text{structure}(t)} \lor \exists c_2 \in C_{\text{structure}(t)} [c_2 \in I_{\text{structure}(t)} \land \text{connected}(c_2, c)]]
\]

The final properties for a task structure certify that the type of item which is passed along a transition is indeed produced by the preceding task, and that it matches the item class expected by the next subtask:
Property 3.9

The class of the item that is passed along a transition should be produced by the preceding task:

\[ \forall c_1, c_2 \in C_s \ \forall d \in D_s \cup \{true\} \ \forall cl \in Cl_0 \ [\text{trigger}(c_1, c_2, d, cl) \rightarrow cl \in O_{\text{task}(c_2)}] \]

Property 3.10

The class of the item that is passed along a transition, should be the same as the class of item expected by the task referenced by that transition:

\[ \forall c_1, c_2 \in C_s \ \forall d \in D_s \cup \{true\} \ \forall cl \in Cl_0 \ [\text{trigger}(c_1, c_2, d, cl) \rightarrow cl = i_{\text{task}(c_2)}] \]

In the definitions provided above, a transition is defined by a source and destination component combined with a condition and a class for the item that flows along the transition. This definition of a transition is extended to include timing aspects as well as a number of additional operations on the item passed along. The following extensions are made:

1. Each transition has a delay function which determines the amount of time that elapses after the previous task is finished, before the next task can proceed. Especially if different actors are responsible for subsequent subtasks in a task structure, such a delay will be required to represent the amount of time required for the transportation of the item considered.

2. For a transition, an urgency is specified to make clear the importance of the following subtask. The actor responsible for the next task will be notified of this urgency, and will be able to determine how soon he should start to work on it.

3. Although new items can be created with the operation {o}, in the previous task, where this new item is passed along a transition, a special operator is included which can make a copy of the existing item. This special operator is called copy context and if desired, this operator will make sure that a copy and not the original item is passed on to the next subtask. Such an operator is especially useful in situations where carbon copies of the same item are processed in parallel in different parts of the organization.

   If a copy context operator is used in a transition, the arrow for that transition is displayed with a dotted line (see figure 3.4 a).
4. A second operator which can be specified for a transition is the *join* operator. With this operator, the destination subtask is not triggered by each individual item that passes the transition, but is rather triggered at a time when the specified condition evaluates to true. In the mean time, any item that is passed along the transition is added to a compound item created specifically for this purpose. As soon as the condition is met, the compound item containing all the items that were added to it, is passed to the next task at once.

If a join operator is used in a transition, a funnel is depicted along the arrow (see figure 3.4 b, d).

5. The third and final operator that can be added to a transition is the *split* operator, which is the reverse of the join operator. For a transition where a split operator is used, the compound item that flows along the transition is split into its individual pieces. The next subtask is then performed separately on each of these items.

When a split operator is used, an up-side-down funnel is depicted along the arrow (figure 3.4 c, d).

![Diagram](image)

(a) Copy context  (b) JoinUntil(condition)  (c) Split  (d) Join and Split

**Figure 3.4** Graphical representation of transitions

As an example of the *copy context*, *join* and the *split* operators, a task structure is depicted in figure 3.5. We use the example of the financial firm which was used earlier, where the task 'send check back to client', which is performed by the relation manager, is described in more detail. For each bounced check, the relation manager writes a letter to the client explaining the reason why his check was refused. Also, a copy of the check is made and stored in the client’s file. The letters that were made during the day are collected until 6 p.m., and are printed by the operations department during the computer system’s run at night. The next day, the stack of letters is split again, and each letter is sent separately to the appropriate client.
Figure 3.5 Example of using the join and split operators

Hence, the join and split operators ensure that the task 'print letters' is only performed once, regardless the number of letters that need to be printed.

The definition of a transition, incorporating the extensions explained above, is the following:

**Definition 3.8**

A transition $T \in T_S$ of a task structure $S$ is a 9-tuple $T = (x_T, y_T, c_T, i_T, d_T, u_T, cc_T, j_T, s_T)$, where:

- $x_T, y_T \in C_S$ are the origin and destination component respectively,
- $c_T \in D_S$ is the condition,
- $i_T \in CI_0$ is the class of the item that is transferred along the transition,
- $d_T \in IR$ is the delay, and
- $u_T \in IN$ is the urgency.
Furthermore,

- \( cc \gamma : \Pi_0 \rightarrow \Pi_0 \) is the \textit{copy context} operator,
- \( f \in \{ \text{NoOperation, JoinUntil(condition)} \} \) is the \textit{join} operator, and
- \( s \in \{ \text{NoOperation, Split} \} \) is the \textit{split} operator for transition \( \exists \).

In short, we can state that the functional model defined in this section is a way to represent the required coordination between actors who perform tasks within the same business process. A task structure, which is the representation used for this coordination, has two essential characteristics: (1) besides representing the activities of which an overall task is comprised, their ordering is also made explicit, and (2) tasks that are identified at one level of abstraction can themselves be defined in a more detailed structure, yielding an hierarchical description (Pooley 1989).

Techniques for modelling processes, like the task structures used in our approach, can be used effectively in communicating with non-modelling experts (Mosser, Di Felice and Lochoovsky 1989), and, provided an adequate graphical representation is used (Scanlan 1989), offer a powerful communication vehicle amongst analysts and designers.

By using a small number of building blocks, i.e. components and transitions, and by offering a decomposition mechanism by which detailed specifications can be subdivided into small chunks, comprehensible models can be attained.

### 3.4 Analysis of dynamic model behavior

Up till now, static representations of the entities and business processes of an organization are discussed. In order to devise effective support for the business reengineering process, however, it was argued in the previous chapter that a model should also make explicit the behavior of the organization's entities over time. Such an analysis of behavioral aspects can help to determine the correctness of the assumptions that are made during the modelling effort, and can also be used to anticipate the effect of certain changes to the organization.

Some authors view behavioral modelling as an extension to data modelling (e.g., Schiel 1985; Lazarević and Mišić 1991) or to process modelling (e.g., Conrath et al. 1992). Although such extensions can certainly be useful for examining the dynamics of a system from a data or process point of view, we prefer to consider behavioral modelling separately. An entity model is used to represent the actors and items in an organization. The functional model is used to represent the required coordination amongst actors which insures that business processes as a whole can be performed correctly. With behavioral modelling the analyst attempts to represent
the interaction of actors processing items over time (see also Bravoco and Yadav 1985).

Due to the fact that in an organization many actors will perform tasks in parallel to one another, behavioral modelling requires a representation of concurrency. The easiest way to establish this concurrency is to use a hardware configuration which would have multiple processors, each representing an individual actor. These processors could then exchange messages or information items, very similar to the actual interaction of actors in the organization. In fact, with the task structures as a representation of the required coordination, this approach would resemble to a large extent the so-called 'data flow' architectures often used in multiple processor computing (Dennis 1980; Wise 1986).

Alternatively, a hardware configuration with a single processor can be used, where a special mechanism is introduced to establish concurrent processes. One way to do so is by introducing separate system states and by representing a system’s behavior as discrete state changes. A formalism which is often applied for this type of modelling is that of Petri nets (Petri 1962) or variations thereof (e.g., Jensen (1986, 1991) and Genrich (1986)). In a Petri net the system’s states are represented by so-called places, and state changes are represented by transitions. The edges in a Petri net always connect nodes of different types, i.e. a place to a transition (the place is an input place to the transition) or a transition to a place (the place is an output place of the transition).

After a Petri net of a system is devised, so-called tokens can be put into the net in several places, and the transitions can start to 'fire' according to specific firing rules. The basic firing rule for a Petri net states that a transition may only fire if a token is available in all of its input places. If the transition fires, these input tokens are consumed, and tokens are produced for all of its output places.

The formal basis of Petri net models, and hence their aptitude to prove a system’s properties in a formal manner, has led to its wide spread use for the modelling of systems in general (Sargent 1988; Kamper 1989; Balbo et al. 1989), and for information systems modelling specifically (Massapati and White 1989; Tsalgaridou and Loucopoulos 1991). Also, Petri nets have been used in information systems development methods, where they are combined with other modelling techniques, e.g., with entity modelling (Kappel and Schrefl 1991; Van Hee and Verkoulen 1992), or logic (Lee 1988).

Its virtue of formality is, however, also its major drawback. Petri nets are rather abstract representations of reality, where a system can only be described in terms of places and transitions. Petri nets tend to be hard to comprehend for those who are not trained in using them. Also, Petri net models that need to describe complex situations will become very large, encompassing dozens of places and transitions. This, combined with its poor naming conventions, make that it is very difficult to comprehend such a model, hindering the validation of its correctness.
A different, less restrictive way of representing concurrency is provided by the sequence control mechanism which was introduced by Dahl and Nygaard (1966) in the SIMULA language. With this mechanism, events that occur in a system which is modelled are time stamped, and executed in a chronological order. The current system time is the time associated with the event that occurred most recently, and new events can only occur at a time equal to or greater than this current system time. When more than one event is scheduled for the same time, these events are dealt with consecutively, without changing the system time. Hence, parallel processing is facilitated even when only a single processor is available. For this reason the term quasi-parallel processing is often used for such a mechanism.

In an extension of SIMULA called DEMOS (Birtwistle 1979) a set of general concepts including entity and resource is provided which can be used to represent complex systems with relative ease. For an entity in DEMOS a process can be specified which represents the different activities that constitute the entire life-cycle for that entity. As part of this life-cycle process an entity can acquire and release various resources. As such, these two concepts provide an easy way to represent complex coordination and interactions in the actual situation modelled. In addition to the entity and resource, more complex concepts are also provided by the DEMOS context, such as a co-opt mechanism which is used to represent entities that need to cooperate with one another during their life-cycle process. As such, The DEMOS context provides a range of modelling concepts, where it is left up to the designer to determine which of these concepts can best be used for his specific purposes. Hence, the language facilitates system simulation without imposing one single conceptual framework for simulation modelling (e.g., Hills (1973) and Derrick et al. (1989)).

In our approach, we adopt the sequence control mechanism as a way to represent concurrency due to the fact that it does not restrict the designer by imposing a single conceptual framework, like was the case with Petri net modelling where only states and transitions could be used. The sequence control mechanism will be our representation of time, where a chronological execution of events in the system modelled is guaranteed even if these events take place concurrently.

The conceptual framework we use in our approach is not adopted from the simulation languages mentioned above. Rather, it is tailored to our specific needs. By doing so, analysis of dynamic model behavior is possible without any translation between the modelling concepts used in the entity and functional model. As our conceptual framework we adopt the concepts that were already introduced in the static representation techniques discussed in the previous sections. Hence, this contains the concepts of actor, item and task. Basically, only two types of events can occur within this framework: (1) an actor is requested to perform one of his skills on an information item; and (2) an actor finished the information processing task he was working on.
After an event of the second type has occurred, this actor may in turn invoke others to proceed with the processing of the information item he has just been working on, and hence an event of the first type may evolve from this.

One could argue that a third type of event exists, namely the arrival of an information item that needs to be processed. However, such an event is equivalent to the first type of event identified, due to the fact that when an item is received the responsible actor will be asked to perform the task associated with the initial component in the task structure specified for the specific information item.

Before we discuss the precise behavior of an actor for both of the identified events, we will first introduce an agenda for actors which is used to keep track of the tasks they still need to perform.

On an agenda jobs are placed which represent tasks that need to be performed by an actor. Hence, such an agenda serves as a 'to-do' list. Simple actors have to take care of the jobs on their agenda themselves. Compound actors, however, merely delegate their jobs to one of their constituents. To do so, each compound actor is equipped with a series of evaluation functions by which the constituent who is least busy, or is working on a job which has the least urgency, can be found. A suited constituent can not always be found instantly, implying that jobs may reside on the agenda of a compound actor for some time, until an appropriate constituent who can perform the job is found at a later point in time.

We define a job and an agenda in following way:

**Definition 3.9**

A job \( j \) is a 6-tuple \( j = (t_j, p_j, c_j, i_j, u_j, r_j) \), where:

- \( t_j \in T_o \) is the task, either simple or compound, defining what work the job entails,

- \( p_j \) is the parent job or \( \Lambda \) if no such job exists,

- \( c_j \in C_{\text{structure}(p_j)} \cup \{ \Lambda \} \) is the constituent in \( \text{structure}(t_j) \), or \( \Lambda \) if \( p_j = \Lambda \),

- \( i_j \in I_o \) is an item, simple or compound, called the context item, which is to be acted upon while task \( t_j \) is performed,

- \( u_j \in \mathbb{N} \) is the urgency, and

- \( r_j \in \mathbb{R} \) is the remaining time for job \( j \).
Definition 3.10

An agenda $g$ of actor $a$ is a 3-tuple $g_a = (A_a, W_a, c_a)$, where:

- $A_a$ is a set of active jobs,
- $W_a$ is a set of waiting jobs, and
- $c_a \in A_a \cup \{\Lambda\}$ is the current job being performed by actor $a$, where $\Lambda$ implies the actor has no current job.

The set of active jobs $A_a$ contains all the jobs that are currently being performed. Besides the current job, all jobs that have been started, and which refer to compound tasks, are part of $A_a$. The set of waiting jobs $W_a$ contains all the jobs that still need to be served by actor $a$.

Besides an agenda, every simple actor also has a rule-book which helps him (or her) to determine the most important job from the set $W_a \cup c_a$. This rule-book is triggered every time new jobs are added to the waiting list, or when the actor is finished performing its current job. Although this rule-book can contain production rules, specifying in detail the process by which the most important job is determined, an effective and simple set of rules can be adequate to describe many organizations:

1. If the actor is busy performing a job, and another job is waiting for him which has a significantly higher urgency, this urgent job is considered to be the most important, i.e.

$$c_a \neq \Lambda \land \exists j \in W_a \land u_j - u_{c_a} > \text{threshold} \land \neg \exists k \in W_a \land [u_k > u_j] \rightarrow \text{return}(j)$$

where threshold determines the difference in urgency that is required for an actor to interrupt his current work.

2. Otherwise, if the actor is performing a job currently this current job is still considered to be the most important, i.e.

$$c_a \neq \Lambda \rightarrow \text{return}(c_a)$$

3. If the actor is not busy at the moment, and there are jobs waiting to be performed, the most urgent job is selected, i.e.

$$\exists j \in W_a \land \neg \exists k \in W_a \land [u_k > u_j] \rightarrow \text{return}(j)$$
4. Finally, if none of the rules specified above have fired, the actor will become inactive, as no job can be found for him to do, i.e.

\[ \text{return}(\Lambda) \]

In order to be able to analyze the behavioral aspects of the organization modelled with an entity and functional model, each actor is equipped with an agenda as defined above, and a sequence control mechanism is used to administer consecutive events in the organization. We can now describe precisely the model behavior for both of the event types we distinguished earlier. We will first discuss the behavior of an actor who finishes his current job, and then proceed with the behavior of actors who are asked to perform a job, where we make a distinction between a job being received by a compound and by a simple actor:

1. A simple actor \( a \) finishes its current job \( c_a \)

When a simple actor finishes its current job \( c_a = (t_{ca}, p_{ca}, c_{ca}, i_{ca}, u_{ca}, r_{ca}) \), this job is removed from its agenda, or from its active job list \( A_a \) to be precisely. If \( t_{ca} \) is a simple task, then the code of \( t_{ca} \) is performed upon item \( i_{ca} \), and possibly affects the attribute values of \( i_{ca} \) and may also affect the attribute values of actor \( a \).

If the jobs parent job \( p_{ca} \neq \Lambda \), then the component \( c_{ca} \) is passed to \( \text{structure}(t_{pca}) \), which then 'knows' which component in its structure has terminated, and hence can determine the jobs that need to be initiated by means of its transitions.

For each transition \( T = (x_\pi, y_\pi, c_\pi, i_\pi, d_\pi, u_\pi, cc_\pi, j_\pi, s_\pi) \) in \( \text{structure}(t_{pca}) \) where \( x_\pi = c_{ca} \), the following procedure is followed: If the condition \( c_\pi \) is met in the context of the item \( i_{ca} \) and the attribute values of simple actor \( a \), then a new job \( j = (t_j, p_j, c_j, i_j, u_j, r_j) \) is created with the following values:

\[
\begin{align*}
t_j &= \text{task}(y_\pi) & \text{The task for } j \text{ is the task referenced by } y_\pi. \\
p_j &= p_{ca} & \text{The parent job for } j \text{ is the parent job of } c_{ca}. \\
c_j &= y_\pi & c_j \text{ is the destination component } y_\pi \text{ of transition } T. \\
i_j &= cc_\pi(i_{ca}) & \text{The context item for } j \text{ is the result of the } \text{copy context} \text{ operator } cc_\pi \text{ performed on the context item of } c_{ca}. \\
u_j &= u_\pi & \text{The urgency of } j \text{ is the urgency associated with transition } T. \\
r_j &= \text{duration}_{\text{task}(y_\pi)} & \text{The remaining time for job } j \text{ is a sample from the duration function of the task referenced by } y_\pi.
\end{align*}
\]

The new job is placed on the agenda of \( \text{actor}(y_\pi) \), if this is not equal to \( \Lambda \), and the actor of \( p_j \) otherwise.
This procedure is slightly more complicated when the *join* or *split* operator is used. If the join operator \( j_x = \text{JoinUntil}(\text{condition}) \), no new job is created, but the result of the copy context operator performed on context item \( i_{e_a} \) becomes a constituent of a compound item associated with the transition. If at some point in time the condition evaluates to true, a job identical to \( j \) is created, with this compound item as its context item \( i_j \). If the split operator \( s_j = \text{Split} \) and the context item \( i_{e_a} \) is a compound item, then for each constituent of \( i_{e_a} \) a job is created which is identical to \( j \), but which has this constituent as its context item \( i_j \).

After all the relevant transitions have been processed, the simple actor \( a \) will trigger its rule-book in order to determine its next job. Preceding this search, the actor will have to inspect the agendas of those compound actors by which he is encompassed. On these agendas jobs may be waiting which are more urgent than the jobs the actor has himself.

For all the actors that have received new jobs due to the transitions, the procedure explained at 2 and 3 below is followed.

If no transition has 'fired', the parent job \( p_j \) is reviewed to see if this job is finished. This does not necessarily have to be the case due to the fact that parallel tracks of subtask execution may exist, where not all of these tracks are finished yet. If it turns out that \( p_j \) is indeed finished, the procedure described above is followed through once again.

2. A new job \( j \) is placed on the agenda of a simple actor \( a \)

If job \( j \) refers to a simple task, it is added directly to the list of waiting jobs for \( a \), i.e. \( j \) is added to \( W_a \). If, however, job \( j \) refers to a compound task, it is added to the list of active jobs \( A_a \) and for each of the initial components in \( \text{structure}(t_j) \) a new job is created and added to the appropriate agenda (this does not necessarily have to be the agenda of \( a \)).

Next, the rule-book of \( a \) is triggered in order to find the most urgent job on its agenda. If the selected job is different from the actors current job \( c_a \), and \( c_a \neq A \), then this current job needs to be interrupted. To do this, the remaining time for job \( c_a \) is determined and stored in \( r_{c_a} \). The event that was scheduled to warn the actor when its current job would be finished is disposed. Next, job \( c_a \) is removed from the active list \( A_a \) and placed in the waiting list \( W_a \).

Finally, actor \( a \) will make the selected job its new current item, again assuming this selected item is different from the job that was already current. The actor will also schedule an event for time \( \text{Now} + r_{c_a} \), where \( \text{Now} \) represents the current simulation time. Once the simulation time reaches this point, actor \( a \) will be triggered to perform the procedure described under 1 above, as by this time the current task will be finished.
3. A new job \( j \) is placed on the agenda of a compound actor \( a \)

When a compound actor \( a \) receives a new job on its agenda, an evaluation function will be performed to determine the constituent best suited to assign the job to. If no suited constituent is found, the job will remain in the waiting list \( W_a \) until a constituent removes it from there. However, if a suited constituent is found, the job is passed on to this constituent, at which time procedure 2 or 3 is repeated, dependent on the fact whether this constituent is itself compound or simple.

With these descriptions of an actor’s behavior in case of an event as identified in our approach, the definition of the modelling techniques is finished. The agenda and the sequence control mechanism discussed in this section enable the designer to simulate an organization’s behavior over time, based on the static descriptions of this organization in an entity and a functional model.

3.5 Summary and conclusions

At the closing of the previous chapter a number of propositions were made pertaining to an effective way of supporting a business reengineering process. Two characteristics which are considered essential in these propositions are the use of modelling concepts which closely relate to a person’s existing mental model of an organization, and the incorporation of behavioral aspects of the organization in the analysis effort. In this chapter these propositions have been translated into a specific approach to model organizations.

First, we have incorporated the modelling concepts of information worker and information item in a so-called entity model of an organization. For such an entity model, four super classes were defined which provide a vocabulary for describing these organizational entities. The active entities of an organization, i.e. information workers, were represented by the super classes simple actor and compound actor, where the latter was used to represent a group of information workers like a department. The information items in an organization were represented by the super classes simple item and compound item, where the latter was used to represent a combination of information items like a file or dossier. Items were represented by a set of attributes. In addition to such attributes, the definition of an actor also included a set of skills which resemble the information processing tasks of which the actor is capable, where a task was defined as an activity performed by an actor on an item in order to obtain a certain goal.

When an organization’s entities are modelled, specialized entity classes can be defined which inherit the structure of one of the super classes distinguished.
Following this, instances of these specialized classes can be made which represent the actual entities which exist within the organization modelled.

The second model which we have presented in this chapter is the *functional model*. In such a functional model, the required coordination between the actors in the organization was made clear. To represent this coordination, *task structures* were introduced. A task structure can be used to define a task recursively in terms of a number of subtasks. A task structure not only makes clear which subtasks belong to a task but also specify the order in which such subtasks are performed. The subtasks in a task structure may be performed by different actors. Hence, the required coordination between actors is captured explicitly by such a task structure.

At the highest level of abstraction, a task structure represents one of the organization’s business processes, and it encompasses all the tasks that need to be performed in order to process a specific type of information item. By decomposing the subtasks in such a high level structure, a more detailed specification of one of the aspects of the process as a whole can be obtained. As decomposition can again be performed at the next level, the functional model will eventually become a hierarchical description of the organization’s business processes.

In the final section of this chapter, we introduced a mechanism which can be used to analyze the dynamic model behavior of the static entity and functional models. To represent the concurrent activities of actors in time, the sequence control mechanism which was introduced in the SIMULA language was selected. This mechanism facilitates the analysis of concurrent processes without imposing a restrictive conceptual simulation modelling framework.

In addition to the sequence control mechanism, the concept of agenda was introduced which is used by actors to keep track of the tasks they still need to perform. As such, the agenda serves as a ‘to-do’ list for actors.

The two events that were distinguished in our approach are (1) an actor is asked to perform one of its skills on an item, and (2) an actor finishes his current task. For both of these events a description of the actor’s behavior was given. Using these descriptions, the behavior of a specific organization can be exposed and analyzed. As such, the approach presented in this chapter indeed encompasses the characteristics that were proposed in the previous chapter. In the next chapter, the process of model construction will be analyzed in order to attain an overview of the required support for the modelling endeavour as a whole.
4 A dynamic modelling support environment

4.1 Introduction

In the previous chapter an approach for modelling information intensive organizations was described. In order for this approach to be applied successfully, adequate computer support is essential. For the entity model and functional model changes to descriptions made earlier can be very tiresome if only pencil and paper were used. Also, a manual modelling procedure would make it impossible to check and enforce the properties that have been set up for these different model types. Finally, the analysis of the dynamic model behavior of an entity and functional model with the agenda and sequence control mechanism put in place, would be impossible without computer support.

In this chapter the support required for the modelling approach is discussed. The objective is to design an environment which enables analysts to construct, refine, change and analyze the models that are used in the business reengineering effort. Besides the support for actual model construction and manipulation, the environment should enhance the communication with and between people in the organization as the purpose of the endeavour is to create an awareness within the organization of potential improvements. For this reason we use the term environment in preference to system, as this emphasizes the positioning of the support within the context of the interacting people involved in the process, whereas the term system conveys an image of a technical device, comprised of hardware and software, which is regarded in isolation from its context.

In the presentation of our dynamic modelling approach in the previous chapter the modelling techniques that are used to model an organization were described. However, the way in which such models are actually devised within the business reengineering process, is not discussed yet. In an attempt to derive a set of requirements for modelling support, a clear understanding of the modelling process
is essential. Therefore, we start in section 4.2 with an analysis of the modelling process. This analysis yields a set of required modelling tools, which should be offered by the support environment.

In section 4.3 the different modelling tools that were discerned are combined in an overview of the functional requirements for a dynamic modelling support environment (DMSE). In this section, the functionality of these tools is also described in more detail.

During the research a prototypical implementation of the DMSE has been constructed. In section 4.4 several consideration concerning this implementation are discussed.

### 4.2 Analysis of the business reengineering process

To represent processes in organizations, task structures were introduced in the previous chapter. These task structures facilitate a description of a task in terms of the subtasks of which it is comprised, and the order in which these are performed. As such, task structures provide a representation which can just as well be used to describe the business reengineering process itself. Therefore a task structure for this process, as well as for some of its subtasks, is devised in this section. Using these task structures, the required support can be determined by distinguishing the essential support tools for each of the subtasks distinguished in these structures. The combination of all these support tools then constitute the functional requirements of the DMSE.

In chapter 2 a process of problem solving was advocated for business reengineering. In figure 2.2 on page 15 a representation of this process was given, making explicit both descriptive and prescriptive models, as well as the different modelling activities that are performed in order to construct, verify and evaluate these. The activities that are distinguished can be subdivided into three areas: (1) specification and validation, which result in a descriptive model; (2) diagnosis, solution finding and evaluation, which yield a solution strategy based on a thorough understanding of the problem at hand; and (3) implementation and post-evaluation, by which the solution strategy is implemented in the organization and its effect is determined.

In figure 4.1 a task structure for the business reengineering process is given. In this task structure the different modelling activities which are distinguished in figure 2.2 are represented by tasks. The transitions connecting the tasks represent the order in which these tasks are performed. This can be in a straight linear fashion, but it may also contain iterations.

Each of the subtasks in figure 4.1 is discussed separately in subsequent sections. In our research, we emphasize the first five subtasks which are directed to establishing an understanding of an existing organization and the problems perceived.
within it, and attaining a set of potential improvements which may resolve these problems. As indicated by the shadowed rims in figure 4.1 separate, more detailed task structures will be given for most of these subtasks.

![Diagram](image)

**Figure 4.1** The task 'Business reengineering'

### 4.2.1 Specification

The specification task is performed in order to obtain a *descriptive* model of an organization. Such a model is used in the process as a frame of reference by which existing problems can be made clear and analyzed, and which can be used to evaluate the effect of proposed solutions.
During the specification task, a vague initial understanding of the organization and its perceived problems is transformed into a detailed entity and functional model of the organization. Evidently, such a transformation is not performed entirely at once. Therefore, we make a distinction into different levels of detail:

1. At first, a preliminary set of concepts is set up which will later be used to specify the organization. Hence, these concepts set out in broad terms the variables in terms of which the specification of the organization is made. The term *conceptual model* is used to denote this set of concepts. As part of this conceptual model, the entity classes that are considered important are identified, and a preliminary specification of these in terms of some attributes and skills is given. Also, tasks are identified for the organizations main business processes which are considered to be important for the modelling effort. For these tasks, a preliminary structure can be devised, the details of which can be dealt with at a later time.

2. Following the conceptual model, a *structural model* of the organization is set up. Similar to the conceptual model, the structural model comprises both the entity and functional model of an organization, although the level of detail for both is increased. In the structural model, a final specification of the identified entity classes is given. Also, a set of entity instances of these classes can be made. Finally, the task structures for the identified tasks are described in more detail. During the specification of such a structural model, a trade-off between model complexity and relevance needs to made almost on a continual basis. Simplifications are made to entity class descriptions and task structures where possible, as long as the value of the model as vehicle of analysis of the existing organization and its perceived problems is not harmed.

3. Based on the structural model, a *model system* (Sol 1982) is set up. The term model system is used to denote the executable model of an organization. Such a model system can be attained by determining the required input data such as delays and priorities in by incorporating these into the task structures specified in the organization’s functional model. Furthermore, the agenda and sequence control mechanisms are linked into the model system implying that the model system can be executed to expose the behavior of the modelled organization.

A task structure for the specification task is given in figure 4.2, where the models distinguished above are each addressed separately.

Again, some iterations may be required between the three modelling activities. We discuss each of the subtasks comprised in figure 4.4 separately below.
Conceptualization

Conceptualization is the first step to be performed in a modelling process. It is directed to attaining a set of modelling concepts which are considered to be important for describing the current organization and the problems perceived within it.

A task structure for the conceptualization task is depicted in figure 4.3. As a starting point for the conceptualization task, an initial global problem formulation can be specified which offers a first demarcation of the scope that is of interest during the modelling effort. Due to the complexity of the organization and the limited, partial and contradicting images that exist of its structure and performance, this initial problem statement is often superficial and may address a very diverse set of aspects. In fact, a number of iterations between conceptualization and the specification of the structural model or the model system may be required in order to establish a clear understanding of the problem(s) in question.

During the conceptualization, entity classes are identified, and possibly specified globally in terms of an initial set of attributes or skills. Similarly, a first list of tasks can be identified for the main business processes considered important for establishing a clear understanding of the current organization. Also, an initial global structure can be specified for these tasks. For each entity class and task that is identified, its purpose should be made clear in light of the problem statement that is agreed upon earlier. If such a purpose can not be found, it is better to leave it out in order to restrict the complexity of the resulting conceptual model.
The result of identifying or specifying an entity class or a task is evaluated. It is determined whether additional actions are required for the conceptual model, or if the problem formulation needs to be revised. If neither of these are called for the modelling process will proceed with the specification of the structural model. The conceptualization may be resumed if a revision of the conceptual model is called for at a later point in time.

**Specification of the structural model**

The structural model encompasses the entity class definitions in their final form, as well as the detailed specifications of the identified tasks. The task structure for this task is depicted in figure 4.4. During the specification of the structural model, entity classes are (re)specified and entity instances are created. Similarly, the task structures are (re)specified to incorporate the details that were left out earlier.
During its specification, the resulting structural model is checked to see if it still needs to be refined, or if certain parts of the model have been described in too much detail. In this last case, a number of simplifications can be made in order to reduce the complexity of the model while preserving its essential aspects. For the entity model, a general class specification can be made for entity classes which were observed separately during the conceptualization, but which no longer need to be discriminated. Also the definition of entity classes may be simplified by leaving out the attributes and skills for which a purpose can no longer be found. For the functional model, detailed task structures may have been described for certain tasks, which turn out to have no contribution to the understanding of the organization in question. Such tasks can then again be regarded as a black box for which no further decomposition exists.

Verification of the structural model

Once a structural model is specified, a verification can be performed to ascertain its correctness.

For the entity model, a textual description of each class and its purpose in the structural model can be made up. The skills that were identified for the actors can be checked against the final task structures to see if no skill has been left out. A summary overview containing all the identified classes, preferably using some graphical representation scheme, can be used to determine the completeness of the model, and to see if the level of aggregation in the final descriptions is consistent throughout the model.
For the functional model, the task structures which have been devised can be discussed with the different departments that participate in the processes modelled, to check for any misinterpretations that might have been incorporated in these task structures. Also, the assumptions and simplifications that were made during the modelling effort can be checked to see if these might have harmed the correspondence between the model and reality.

Specification of the model system

As stated earlier, the term model system is used to denote an executable model which can be used to expose the behavior of the system that is modelled, i.e. an organization in our case. In some approaches, such a model system can be specified separately from the input data. In the Siman language (Pegden et al 1990), for instance, a distinction is made between a model frame and an experiment frame, where the latter contains the input data specified for the structural model contained in the model frame. In our approach, such a distinction is not made. The input data such as delays and priorities are specified in the simple tasks and transitions in the functional model. Hence, we use the term model system in reference to the structural model in which all of the required input data has been specified.

A task structure for 'specification of the model system' is given in figure 4.5. For each piece of input data, the correct value has to be determined and incorporated into the model system.

For a model system to be complete, the following types of input data need to be collected:

1. For each highest level compound task, i.e. tasks which are not a subtask of any other task, the triggering event needs to be determined. For instance, in our example of the financial firm in the previous chapter, this trigger would be an incoming check. For such an event, its time and frequency of re-occurrence needs to be specified.

2. For each simple task contained in the functional model, the time it takes to perform that task needs to be specified. Such a delay specified for a simple task can be constant, although in most cases it will be a stochastic value specified by a distribution function.

3. The delay for each transition in the task structures comprised in the functional model, need to be specified.

4. For each alternative path in the task structures the probability of following that path is required.
In addition to these input data, the priorities for each skill of every actor contained in the entity model should be determined, and perhaps specific rules should be added to the actor's rule-book if these differ from the default rules mentioned on page 48-49.

To determine the correct values for the input data distinguished above, several techniques can be used. Four of these techniques are depicted in figure 4.5. An expert assessment is used in those cases where data is not available, and it can not be collected due to time constraints. However, personal biases or misjudgments can cause significant errors in these measures, implying that at the very least a mechanism of cross examination should be used. More reliable sources of information are archive material and available data sets, although these needs to be sufficiently up-to-date in order to be useful. If the information is sufficiently important, and an estimate is not secure enough, one might consider to conduct a measurement of sample data in the organization in question.

**Verification of the model system**

Similar to the structural model, a verification of the model system can be performed to ascertain its correctness. The objective of this verification is to determine whether the model system correctly represents what the designer(s) intended (Shannon 1975).
In order to verify the model system, impulses can be given to it, and the behavior that results from this impulse can be examined. In the example of the financial firm, for instance, a single check can be generated in the model system and the different steps in handling that check can be followed through. The correctness of delays, priorities and probabilities for alternative paths can hence be checked.

As part of the model system's verification, possible coding errors need to be found (Hoover and Perry 1989). In our approach, these coding errors can emerge when either the rule-book is changed in comparison with the default rules provided, or when a wide variety of urgency levels is used for the different skills of an actor. Therefore, the behavior of an individual actor in the model system should be inspected closely, to see if the order in which the different jobs on the agenda are treated, correctly represents actual working patterns in reality.

This concludes the description of the specification task as depicted in figure 4.2. The result of this task is a descriptive model system which is certified to be represented correctly. The next step in the process is to check whether the output of the model system accurately resembles reality.

### 4.2.2 Validation

The validation task is performed in order to certify the correctness of the model system's output. In order to attain such model system output, a so-called treatment of the model system should be made.

A treatment is comprised of the following (Ören and Zeigler 1979; Sol 1982):

- a specification of input data
- a collection of input data
- initialization conditions
- run control conditions
- a specification of output data

A run is the combination of a treatment and the collected output data. Replications are runs which are performed under the same treatment. The total set of replications for a specific treatment is called an experiment.

The initialization conditions determine the time it takes for the model to overcome possible start-up effects (Pritsker and Wilson 1979). During this time the output of the model is invalid and should hence be disregarded. An adequate start-up time can best be determined by using a trace of the model in which the value of an output variable or internal state variable is tracked while simulation time proceeds.
By projecting such a traced value against the simulation time, the amount of start-up time required for the variable to stabilize can be determined.

Run control conditions determine the run length, which is the amount of simulation time that is used for measurement of model output. Furthermore, these conditions determine the number of replications required, i.e. the number of independent observations of the model system's output, as well as the way in which these replications are performed. The number of replications required depends on the reliability intervals that are selected for the model outputs. The easiest way to perform different replications is to let each run start with its own start up time, implying that the only difference between runs is their initial random stream values. Alternatively, one long run can be subdivided into several small pieces, where each of these pieces is regarded as one replication. Some disadvantages of this method, caused by autocorrelation effects, are summarized by Sol (1982).

The specification of output data concludes the treatment specification. An important decision to be taken in this respect is whether the output of the model system should already be aggregated during its execution, or should be performed afterwards on the raw data generated. For enabling both, many kinds of standard model output can be made available, and special measuring and reporting devices can be offered which can explicitly be put into the model system in order to attain some special model output.

A task structure for devising a treatment for a model system is depicted in figure 4.6

![Diagram](image)

**Figure 4.6** The task 'Devise treatment'
Due to the fact that the model system will contain various stochastic elements, the output derived from a model system will be different for each run, even if the same treatment is used. Therefore, a confidence level has to be chosen for each of the model outputs. With such a confidence level, the number of replications that is required can be determined using statistical techniques. Often, the model output will pertain to average values, which are distilled from a significant amount of individual independent observations during the simulation run. According to the central limit theorem the distribution of such average values is a close approximation of the normal distribution. For such normal distributed model outputs the $t$ distribution can be used to determine the number of replications required, given a chosen confidence level (Van Soest 1985; Thesen and Travis 1992).

Once a treatment is devised for a model system, validation can proceed. In figure 4.7 a task structure is given for the validation task.

A distinction is made between structural validation and replicative validation. Structural validation is performed in order to check whether the model behavior in special situations is consistent with what the designer would expect. Replicative validation is performed in order to determine the degree of similarity between the numerical output of the model system and the corresponding values in reality.

![Figure 4.7](image)

**Figure 4.7** The task 'Validation'

During the structural validation, special treatments are devised for the model system in order to check if the predicted model behavior is observed. An example of such a treatment could be that no information items are generated, and hence no
actor should be active during the simulation run. Similarly, specific actors could be removed from the model system to check if the work intended for them indeed starts to pile up correctly. The structural validation is intended to increase the confidence in the model system, as it not only projects the behavior of the current situation correctly (this is established during the model system verification) but also has a correct reaction to a number of extraordinary situations.

During the replicative validation the numerical output of the model system is compared with the corresponding values in reality. For this purpose, an experiment is performed with the model system. Also, the appropriate values are determined in the organization itself. These two sources of data are then compared. The extent to which replicative validation can be performed depends on the available data from the actual organization. If this data is limited, a comparison might be made between the model system's output and the expected values of an expert in the organization. This evidently only establishes a certain degree of validity for which the term face validity is often used. If data are available, statistical methods of comparison can be used, such as the t-test for comparison of mean values, and the F-test for comparison of variances. Also, the so-called Turing test can be performed where reports based on the model system's output and reports based on actual data are presented to a team of experts. If these experts can not discriminate the two, the model system is said to be valid.

This concludes the validation task. As a result of this task, a descriptive model system which is proven to be a valid representation of reality is available. This descriptive model system can then be used to analyze some of the problems perceived in the organization (diagnosis) or to evaluate the effect of certain changes in the organization's business processes or in the individual work places of certain actors (solution finding). This use of the descriptive model system as a yard stick during the experimentation with changes represented by prescriptive model systems is discussed below.

4.2.3 Diagnosis

After a valid descriptive model of the organization is made, diagnosis of perceived problems can be performed. In figure 4.8 a task structure is given for the diagnosis task. In short, problem diagnosis is performed by making changes to the model system and by observing the effect of the change on the model system's output. At first a trial-and-error search for change propositions is conducted in order to attain a global understanding of the relations between input and output variables. Later, a more formal experimental design can be set up to determine which combinations of changes need to be analyzed.
4.2.4 Solution finding

The activity of solution finding greatly resembles that of diagnosis, as far as the way of working is concerned. Again, parameter values, entity instances or classes, and task structures in the descriptive model are altered in order to examine the effect on the model output. The difference between diagnosis and solution finding stems from the difference in objective, where the former is focused on improving the understanding of perceived problems by determining the causes, and the latter focuses on finding ways to resolve them. Due to the similarities, the same task structure of problem diagnosis (figure 4.8) is used to denote the solution finding task.

During the first part of the solution finding task, a learning process is established for the participants in the organization in question. Propositions for changes made by these participants are analyzed by incorporating the change in the model system, and by performing a treatment with it. As such a changed model projects a situation which is different from the current one, we use the term *prescriptive model system* to denote these. For a trial-and-error search to be effective, the impact of such changes should be made clear directly, without long delays caused by extensive remodelling or slow simulation speeds. As a result of this rather informal search for alternative changes, a broad set of possible measures will be attained, each of which is already analyzed individually.

For proposing ideas for change, the expertise of an analyst, and the creativity of both the analyst and the participants in the organization are of crucial importance. The expertise of the analyst can contribute to change propositions by drawing a comparison with similar problems that have already been dealt with in previous
experiences. Also, certain sources of knowledge, perhaps from different fields, can be very useful at this stage. With respect to improving information intensive organizations, for instance, incorporation of logistic principles in the solution finding process can yield new effective courses of action which might otherwise have been left out (Groenenboom 1989). Also, theories on organizations and organization design (e.g., Mintzberg 1979, 1988; De Leeuw 1990; De Sitter 1987, 1989; Huber 1984, 1990) can be used to determine potential improvements.

After a set of alternative solutions is established, the next step is to find appropriate combinations of these changes. Due to the fact that the entire set of alternatives may be quite extensive, an experimental design should be made for this purpose. An experimental design technique reduces the number of possibilities that need to be considered. Kleijnen (1974) describes different experimental design techniques that can establish a reduction of required experiments. Sol (1981) even presents automated support for the making of experimental designs. A significant reduction of alternatives can be established by first selecting those alternatives which counter act one another's effect. Evidently, these propositions do not have to be considered in combination. Furthermore, if no interrelation exists between two alternative changes, the effect of their combination will simply be the sum of the individual effects. Hence, only in those cases where interrelated change propositions exist, separate experiments are required to assess their combined effect.

4.2.5 Evaluation

After the experiments for all the alternative changes proposed during the diagnosis and solution finding task have been performed, an evaluation is performed in order to make a selection of the alternatives which are considered to be most effective.

During this evaluation, the required investments for each change alternative are estimated. These investments include both the costs in terms of time and money required to establish the change, as well as qualitative aspects such as the required flexibility and motivation of personnel effected by the change. These investments required to make a change possible can then be compared with its estimated profit. An adequate comparison of the profit and required investments facilitates a way to select the alternative which is most preferable for the organization.

Although during the diagnosis and solution finding step emphasis is placed on the comparison of numerical outputs, the evaluation task specifically focuses in on effects which may not be measured numerically. How important is customer satisfaction, and how important is the work satisfaction of employees, or are these considerations interrelated? Questions like these need to be answered in order to select the most preferable solution. Hence, although the numerical output of the descriptive and prescriptive model systems can help in making a comparison between alternatives, other considerations are equally important.
4.2.6 Implementation and post-evaluation

The final step in the business reengineering process is to implement the selected change in the organization. The fact that the proposed solution is represented in a prescriptive model system can be of great help during this stage. Individual people and departments can see what their role is in the new organization, and how this role may differ from their current position. Also, the model system can be used to make clear what the purpose of the change is, i.e. what gains can be attained for the organization as a whole. Without this awareness of the potential gains, and without a cooperative attitude in place, any implementation effort is doomed to fail (Keen 1981; Joshi 1991).

Dependent on the level of aggregation that is selected during the modelling process, the specifications of the chosen solution may need to be refined. The general idea of the chosen solution, however, is clearly projected by the model system devised. The new working procedures can be directly obtained from the task structures contained in the functional model. If specific computer based support for individuals in their work place is suggested as part of the solution, the general functional specifications for such support can also be derived from the model system. Finally, technical considerations such as the required response times and the quantities involved for each transaction in such a computer based support system can be distilled from the model system easily.

During the implementation of the chosen solution, the new working procedures have to be written down and instructions should be given to the people involved. For possible support systems, detailed specifications are made up, and the systems are realized. Again, training of the people who need to work with the new systems is required before actual implementation of the solution as a whole can take place. Finally, if the chosen solution has consequences for the number of people required in each department, a reallocation of people across the organization is performed.

It is important to keep track of the actual impact of the chosen solution on the organization. Forces which may not have been anticipated during the solution finding stage may cause these effects to differ from those which were expected. The term post-evaluation is used to denote the evaluation of the actual impact after the solution is implemented.

When an adequate evaluation of the attained improvements is made, an organization is able to establish an ongoing process of organizational adaptation and improvement. As Huber (1984, p. 942) states, "One approach for accomplishing more rapid learning in organizations is to reduce the ambiguity of feedback about the relationship between organizational actions and outcomes. Another is to ensure the collection and interpretation of such feedback". By measuring the effects and comparing these to the predictions that were made, adjustments can be made where necessary, and even an entirely new process of business reengineering may be
initiated in this way. The results of the post-evaluation can be used to improve the understanding of the organization. In subsequent change efforts, a significant part of the specified models can be re-used reducing the effort and time required.

4.3 Functional requirements of a DMSE

In the previous section, an analysis of the business reengineering process is performed. As a result of this analysis, a task structure is set up making clear the different subtasks distinguished in this process and the order in which these are performed. In this section we review this task structure in order to establish the requirements for a dynamic modelling support environment (DMSE) which is to support the process.

We start with the specification subtask. As was indicated in figure 4.2 a distinction is made between a conceptual model, a structural model and a model system, which each describe the organization in question, with an increasing level of detail. During the specification of these models, both entity classes and instances as well as tasks are manipulated constantly.

During the conceptualization, entity classes and tasks are identified. This implies that a name has to be selected for the new class. Also, a graphical icon representing the entity class can be devised. Such a graphical representation can be specifically useful when the entity model is discussed with people who are not specifically trained in modelling activities (Verrijn-Stuart and Anzenhofer 1988), as will be the case for most of the participants in the organization. Any naming or drawing conventions that might need to be followed should be enforced by the DMSE. During the conceptualization the designer should be able to make initial specifications of entity classes and tasks without being confronted with any consistency rules that result from the properties that were set up for the entity model and functional model in the previous chapter.

During the specification of the structural model, existing entity class and task specifications should be available for the designer. Fast and easy access should be provided to these existing definitions. For the entity model, the designer should be able to specify the attributes and possible skills for an entity class. Also instances of such a class can be made, where the appropriate attribute values are assigned to them. For the functional model, the designer should be able to specify the task structure for compound tasks. The DMSE should maintain the hierarchy of defined tasks and should provide easy access to this hierarchy. Again any naming or drawing conventions should be enforced by the DMSE. During the specification, the designer should be able to check the structural model for possible inconsistencies in light of the properties that were specified in the previous chapter.

For the verification of the structural model, different projections of both entity classes as well as task structures should be provided. For this purpose the DMSE

69
should offer text processing facilities which can combine free text with the projections of the structural model.

During the specification of the model system, support should be available for data analysis. If a decision is made to collect data specifically for the specification purposes, the DMSE should provide an quick and easy to use data base facility by which the data can be entered and analyzed. Also, support for statistical analysis should be available in order to derive the desired quantities from the raw data collected. Finally, the designer should be able to make different projections and charts of the collected data using the DMSE.

The final subtask within the specification task is that of model system verification. For this task, an adequate representation of model behavior is called for, which provides a clear view of the sequence of events that takes place when the model system is executed. We agree with Kramer and Ng (1988), Dachouffe and Lesuisse (1989) and Novels and Weigl (1989) who claim that a graphical animation of model behavior significantly improves the effectiveness of model system verification. If a picture-like projection of the organization is provided by such an animation, people will have no difficulty in understanding the model behavior and will be able to pin point possible errors easily.

After the specification task, a validation of the model system is performed. Before this can be done, a treatment has to be devised for the model system, as indicated in figure 4.7. To devise a treatment, support for statistical analysis is again called for. Based on a selected confidence level, the number of replications can be calculated using such support. In addition to this, the DMSE should again provide facilities for making graphical projections of raw data. With such a facility, traced variables can be projected against simulation time in order to determine an appropriate start-up time.

For the structural validation, animation of model behavior is again very useful. The model behavior in extreme situations can be inspected without having the burden of going through excessive lists of numerical output.

For the replicative validation, statistical test for a difference between two distributions, two mean values or two variances should be provided. Due to the fact that many replications may be required in order to attain valuable model output, the DMSE should make it possible to exclude any animation facilities from the model system in order to increase the model system’s execution speed.

After the descriptive model is specified and validated, the diagnosis and solution finding task can begin. For this task, manipulation of the entity and functional model are again important. In order to construct a prescriptive structural model and model system, changes can be made to existing definitions of entity classes and tasks. Therefore, an easy access should again be provided for the existing definitions, and the designer should be able to specify the required changes in these. Due to the fact
that often only minor changes are made to existing specifications, the DMSE should provide the making of a copy of the descriptive model. Also, version management for the both entity class specifications and for task structures should be available, where a textual comment can be attached to specifications stating the nature of and the reason for the changes made.

After a prescriptive model system is specified, an experiment similar to the one performed on the descriptive model system should be provided. Again, both the graphical animated output as well as the numerical output should be made available, where the designer can choose to turn the animation on or off.

For the final tasks within the business reengineering process, i.e. 'evaluation' and 'implementation and post-evaluation', support similar to that specified above is required. The animated behavior of the prescriptive model system can be used to demonstrate the pursues improvements to those who are involved in the change process. Throughout the evaluation and implementation tasks, the text processing facility is used to produce documents within which the chosen solution is explained and possibly specified in further detail. Finally, data management and statistical analysis are required for the post-evaluation of the actual results obtained after the change is implemented.

Based on these considerations, seven distinct support tools can be identified which need to be incorporated in the dynamic modelling support environment. A global architecture of such an environment is shown in figure 4.9, where the seven distinguished tools, as well as the types of information they manipulate, are depicted.

As is clear from the architecture, different types of information are maintained in the environment:

1. The entity model base contains the different entity models, both descriptive and prescriptive, that are devised during the modelling process. For each entity model, all the entity class specifications as well as the instances created for these classes are stored in the entity model base.

2. The functional model base contains the different functional models which are devised during the modelling process. For each of these models, the total set of specified tasks and task structures is maintained.

3. The program base is used to store executable model systems which can be exposed to a treatment. We shall explain this in more detail when the model system translator is discussed.
4. The *data set base* contains all the data sets that are created and analyzed during the modelling process. Hence, these data sets may be created by the designer, be imported into the environment from sources elsewhere in the organization, or created by a model system as numerical output.

5. Finally, the *document base* contains the total set of documents that is created during the modelling process.

These different types of information are created, maintained and manipulated by the designer by means of seven support tools. We discuss the functional requirements for each of these tools separately.

![Diagram of the dynamic modelling support environment](image)

*Figure 4.9* Global architecture of the dynamic modelling support environment
Entity modeller

With the entity modeller, an impression of which is given in figure 4.10, class definitions as well as entity instances can be created, changed and disposed. Specific conventions that have been set up for entity models are either enforced by the entity modeller, or checked at the designers request.

![Entity class editor](image)

**Figure 4.10** The entity modeller

An entity modeller provides browsing facilities for entity classes and instances, which enable a designer to walk through existing class definitions and instances quickly, and to select the one that needs to be inspected or changed.

In addition to this, the entity modeller provides an editor for entity classes. With this editor, attributes and skills of existing or new entity classes can be specified along with graphical or textual representations for that class. For an entity class, different representations can be made for each state this entity can have. An actor, for instance, could have three possible states: *Active*, *Passive* and *Not Available*. The different representations devised for these states can then be used during the model system animation at a later point in time.

As indicated in the previous section, entity class definitions can change during the modelling process. Also, different descriptions of the same class can exist simultaneously if differences of opinion exist within the organization. Therefore, version management is an important feature of the entity modeller. Changes to a
class description should be time stamped and the designer should be able to attach comments to such changes for documentation purposes. Finally, as changes to an entity class description are usually minor, it should be possible to make a copy of an existing class instead of having to specify attributes or skills redundantly.

An entity modeller is the interface between the designer and the entity model base. The entity modeller also has access to the functional model base in order to be able to check the skills that are specified for an actor, against the tasks assigned to this actor in the various task structures comprised in the functional model. Hence, the consistency between these two model types can be checked and maintained by the entity modeller.

Task modeller

Many similarities exist between the entity modeller and the task modeller. The task modeller is used to create, change and dispose task specifications, and is hence the main interface between the designer and the functional model base. Again, a distinction is made between a browser, which facilitates a convenient way to search through existing task specifications and select the task that needs to be inspected or changed, and an editor, which supports the actual editing of task structures and simple task specifications. An impression of the task modeller is given in figure 4.11.

![Task Modeller](image)

**Figure 4.11** The task modeller
In the editor the different attributes of components and transitions can be specified. For a component that refers to a simple task, the values of attributes like \textit{actor}, \textit{duration} and \textit{code} should initially be copied from this simple task. In the editor, however, a distinction should be made between attributes that are assigned to components, and attributes that are assigned to a subtask referred to by a component. This way, the default attribute values can be specified in the task itself, whereas these default values can be overruled by specifying different values in the component referring to that task.

Similar to the entity modeller, the task modeller has access to both the functional model base and the entity model base. When an actor is specified for a certain task in a task structure, the task modeller can hence determine whether or not this task belongs to the set of skills for that actor. Obviously, the task modeller can not update class specifications, as this is allowed for the entity modeller only.

Finally, version management for the task base should be provided by the task modeller. Therefore, a textual description for each structure should be provided, stating the difference in comparison with the preceding versions of the same task structure. By default, the task modeller should provide access to the most recent versions of the tasks specified in the task base, although preceding versions should also be made available on request.

\textbf{Translator and simulator}

Up till now, we have used phrases such as 'The model system is \textit{executed}' or 'The model system is \textit{run}'. In order to do so, however, the model system should be translated into a form which can be processed by a computer system. For this translation from a model system to a simulation \textit{program}, we propose to use a model system translator in the \textit{DMSE}. Such a translator links the specifications contained in the model system, to a set of run time simulation routines. These routines include a variety of distribution functions, as well as the agenda and sequence control mechanisms which are required for the model system simulation. The result is stored in the \textit{program base}.

Alternatively, an interpreter could have been used to interpret task structures directly (e.g., Tanenbaum 1984, ch. 8), without a preceding translation. The advantage of such an approach is that the model system can be altered \textit{during} simulation, and proceed from the state it was in before the change was made. However, for reasons of execution speed and efficient usage of disk and memory resources, a translator is preferred.

Following its translation, a simulation program can be executed, i.e. experiments with the model system can be performed. In our review of required support, the need for an animation of model behavior on a graphical display was made clear. In some cases, however, the designer is not interested in such an animation, but wants to obtain the numerical output for a number of replications as quickly as possible. Within the \textit{DMSE}, both of these types of simulation should be
provided. Hence, the simulator which is used to execute the simulation programs should offer the possibility of switching the animation facilities on and off at the user's will.

The model system can contain one or more views of the entity model which is displayed during the animation session. The views that are required need to be specified in advance during model system translation, as the required animation features need to be linked into the simulation program at that time. In the case that more than one view is specified, switching between these views during the simulation should be facilitated by the animated simulator. An example of a model system animation is given in figure 4.12. Such an animation consists of (graphical) icons for entity instances, and perhaps, if this increases the comprehensibility of the display, a background resembling the actual situation in the organization modelled. In the example shown in figure 4.12 the different actors specified within the model system are depicted on the display with the name of the task they are working on written underneath.

![A graphical display produced for an animated simulation](Image)

**Figure 4.12** A graphical display produced for an animated simulation

In addition to providing graphical animated output, the simulator has to guide a model system through an experiment predefined by the designer. For such an experiment, the initialization and run conditions can be specified in advance, and the experiment can be conducted without any required intermediate input. If more than
one model system needs to be analyzed, or different experiments need to be performed for a model system, it should also be possible to specify the entire set of required experiments in advance, where the simulator conducts each of these experiments consecutively and accumulates the different sets of numerical output in the data set base in order to be analyzed at a later time.

Data manager

The data manager can manipulate data sets, either imported from external sources, set up specifically for the modelling purpose, or generated during an experiment with a model system. In such a data manager the structure of a data item can be specified where no restrictions on field lengths or formats should be imposed, and the set of data can be inspected according to such a structure. Again, a browser and an editor are comprised in the data manager, where the former provides an overview of all the data items, ordered according to some specified key, and the latter is used to create, change or delete data items.

The support for the analysis of data should cover calculations either within data items, i.e. a calculation of the number of days between two dates, or over data items, i.e. the mean value of an attribute over all the items in a set. Different kinds of date and time conversions should be provided as these are most always required for analyzing existing data sets during the specification of the model system.

Figure 4.13 The data manager
The data manager should also facilitate different graphical representations of data sets or parts thereof. These are used for several purposes. For determining the start-up time of a model for instance, certain state or output variables can be traced during the model execution. This traced value can then be projected against the simulation time, and the steady-state for that variable can be determined. Furthermore, graphical representations of data are used to present the outcome of the model systems. For this purpose, bar charts, pie charts, and different line graphs should be provided.

**Statistical analyzer**

The statistical analyzer is a support tool which provides statistical tests that can be performed on data sets. These tests are used during the specification of the model system, where the values of certain model variables have to be derived from existing data. Also, it is used to determine the number of replications based on a chosen confidence level for a model system's output. Finally, different tests are used to compare a model system's output with corresponding values in reality, or with the results from other model systems.

The statistical analyzer should provide calculations of mean and variance values for data sets, as well as statistical tests like the Mann-Whitney test, the $t$-test or an ANOVA test to determine the difference in mean values between data sets based on a confidence level and interval width, and the $\chi^2$-test or the $F$-test to determine differences in variance between data sets.

**Document processor**

The last support tool distinguished in the global architecture for the *DMSE* is the document processor. Documents made with this processor can include free text, but can also include different projections of the entity models and functional models, as well as any graphical projections of data sets. The document processor is used to construct the documentation by which the models and the results of the experiments are presented to the participants involved or to other people in the organization. Documents are also used to record design decisions made during the process in order for the models to be accessible by others at a later time.

As a summary of the analysis of the business reengineering process and the functional requirements of the *DMSE* that were derived from this analysis, a cross reference table is given in figure 4.14 showing which support tool is used during which task, and vice versa.
Figure 4.14 Cross reference table between support tools and modelling tasks
4.3.1 **Implementation of a DMSE**

The overview of functional requirements of the different support tools discussed above is kept short in order to provide a general outline of the **DMSE**. In this section some considerations related to an implementation of the **DMSE** which is constructed during the research are discussed.

Our implementation of the **DMSE** is devised in order to be able to test the feasibility of our dynamic modelling approach. For this objective, the ease of modelling and the possibility of using the environment to construct and revise models interactively with the involvement of the organization's participants is emphasized. Therefore, specific attention is paid to the user interface of the modelling tools and the ease with which an executable model can be obtained.

Our implementation of the **DMSE** encompasses the first four support tools distinguished in the global architecture in figure 4.9 which are used to construct and analyze models. For the other support tools, standard packages for data base management, spreadsheets and word processing covered our requirements implying that only the appropriate interfaces to these existing packages had to be made.

**Programming language**

For the selection of a programming language, two distinct areas have to be taken into consideration: (1) the implementation of interactive modelling tools, and (2) the representation of an executable model system.

For the first area it is important that adequate screen handling procedures can be devised which provide an easy to use graphical interface for the user of the **DMSE**. Other important facilities are those for memory management and disk usage. Due to these requirements, a powerful third generation programming language is used for this area.

For the second area, an existing simulation language or package should be preferred. Such a language already provides the sequence control mechanism which we use to enable concurrent process execution. A great number of simulation languages and packages exist, such as GPSS (Schriber 1974; Gordon 1975), SLAM (Pritsker and Pegden 1979), SIMSCRIPT (Russel 1983) and Siman/Cinema (Pegden et al 1990) and many small domain specific packages (Banks 1991). Languages like these were applied successfully for the purpose of problem solving in general (e.g., Burns and Morgeson 1988; Szymankiewicz et al. 1988; Hoover and Perry 1989), as well as for information systems development specifically (e.g., Sol 1982; Wierda 1991). However, it is quite difficult to implement the mechanism of an agenda with the associated rule-book in these existing languages. This fact, combined with the complexity of the interface that would be required when two distinct languages are used, has made us decide to implement the required simulation features in the same programming language used for the first area. Similar implementations of simulation features in third generation languages have been
reported for Pascal (Kriz and Sandmayr 1980; Raczynski 1986; Stokking 1989) and Modula-2 (Bolckow et al 1989).

For our implementation we choose to use an object-oriented derivative of the programming language Pascal (Borland 1988, 1989). In this language, so-called UNITS are used in which the data types or object class definitions, procedures and functions pertaining to a certain subject can be combined and encapsulated by an INTERFACE. These UNITS provide a way to divide the program code into small chunks.

![Interactive Modelling Tools Diagram](image)

**Figure 4.15** Subdivision of areas for DMSE implementation

For our implementation, a number of UNITS is devised for making the interactive modelling tools. As depicted in figure 4.15, these units are split into three areas: (1) user interface, (2) memory management, and (3) printer driver. In addition to these, a number of UNITS is made which comprise the simulation features required for making an executable model system. We discuss each of these areas separately below. For a more detailed specification of the units and modelling tools discussed below we refer to Dur and Bots (1992b) and Dur (1992).

**Graphical user interface**

In the global architecture of the modelling support environment, depicted in figure 4.9, the different support elements are integrated by the graphical user interface. This interface, which can itself be defined in terms of a set of classes and instances (Dur and Versendaal 1989), provides a universal access to all the support elements mentioned. In our prototype a direct manipulation user interface style (Shneiderman 1983, 1987; Zeigler and Fährich 1988) is implemented which facilitates rapid incremental reversible operations whose impact on the object of interest are immediately visible. The graphical representation of an entity or task can be moved by dragging it physically across the screen using the mouse device, and its attribute values can be changed by filling out a form which pops up when the object is selected. For the implementation of this user interface, five units are constructed, as shown in figure 4.16.
**Scanner** is the basic unit which provides the access to user input for the other units. The most important procedure is ScanEvent which translates the user input from the keyboard and the mouse into a set of globally defined variables. Examples of such variables are PressedKey and FunctionKey. The ScanEvent procedure also maintains an object called Mouse, within which the screen coordinates of the mouse device and the status of its buttons is stored. In addition to these global variables for user input, the unit Scanner also provides a number of status lines which can be used to project the time, disk capacity or available core memory on the graphical screen. Due to the fact that the ScanEvent procedure has to be called on a continual basis, these status lines are refreshed often enough to guarantee their contents is up-to-date.

On top of the Scanner unit, a unit named Lists is built with a general object class named List which can be used for a variety of purposes such as menus and pick-lists. An example of a menu which is made using the Lists unit is given in figure 4.17.
For each item in a list, the following features can be specified:

- A *representation* which will often be a text but can also be a graphical icon,
- A *data structure pointer* which can be used to maintain a reference to any data structure that is associated with the list option,
- A *condition* which is a handle to a function which accepts a pointer as argument and returns a boolean value. The data structure pointer is passed to this condition function which can then determine whether or not the list option can be selected at this point, and
- An *action* which is a handle to a procedure which accepts a pointer as argument and which performs the desired action. The data structure pointer is passed to this action procedure when the list option is selected by the user.

When the user wants to select an item from a list, its condition is evaluated and if this condition returns a true value, the action is performed.

On top of the List unit, two additional units are built.

The **Windows** unit facilitates the use of multiple overlapping windows on the graphical display. Such a window is made up of a working area and a title bar. At the left hand side of the working area, a button list can be included to provide fast access to a number of operations that are provided for the user in the specific window. Furthermore, a horizontal and/or vertical ScrollBar can be added to the working area in order to enlarge this area without requiring additional space on the display. In figure 4.18 an example of a window is given which could be used for making a drawing.

---

**Figure 4.18** Example of a window
Again, the object Window does not have to know what the contents of its working area is, due to the fact that handles are used to procedures that display the windows content, as well as to procedures that take care of operations like Scroll, Reshape and Close.

For the working area of a window, screen coordinates are automatically translated into centimetres, where the resolution of the graphical screen is taken into consideration.

![Figure 4.19 Example of a form](image)

The **Forms** unit provides a general object class Form which can be used to construct an area on the graphical display in which a number of data entry fields and select buttons can be made. By a single ReadForm operation, the user input into the fields and buttons is handled by the object Form until the user presses the Accept or Cancel button at the bottom of the form. An example of such a form is given in figure 4.19. With the set of building blocks provided by Lists, Forms and Windows, applications with a graphical user interface can be constructed.

**Memory management**

For the construction of the interactive modelling tools, adequate facilities for memory management are very important. To provide such facilities, two units are devised. The first unit is called **Storage** and provides access to the disk of the computer system. The second units is called **Core** and provides access to the computer’s memory.

With the unit Storage a large model can be written to the background memory, where an index to its different elements is maintained to provide a fast access. For this purpose, each data item in a model is assigned a Key, which can be used when retrieving the data item from disk. To reduce the amount of write and read operations required, Storage also maintains a cache memory to store those data items which have most recently been accessed. Only if a data item is not available in this cache memory will a read and/or write operation be performed on the computer disk.
The unit Core facilitates a number of routines for allocating pieces of memory. This unit is not only important for the construction of the interactive modelling tools, but is also an important unit for the simulation of a model system because a significant amount of entity instances will be created during such a simulation, and memory has to be allocated for all of these entities. The two procedures in the Core unit are Allocate(pointer, size) and Release(pointer, size). When a piece of memory is released, it is added to a free list. When a piece of memory is allocated, this free list is first inspected to determine whether a block of the desired size can be found. If this is the case, the block is taken from the free-list and the address of this block is returned. Hence, a new allocation of memory is only performed if the appropriate size can not be found in the free list. With this unit, the number of small holes in the memory which can not be used any more is reduced significantly.

Version handling is implemented in the prototype in a rather simplistic fashion. All the data files pertaining to the same model are located in a single directory. In this directory, a text file is used to describe the purpose of the specific model, possibly the source model it is copied from, and the changes that have been made to it. Given our objective of testing the feasibility of our modelling approach, this version handling procedure is adequate.

**Printer driver**

In a separate unit, a printer driver for a PostScript device is made which perform all the graphical operations which are also provided for the graphical display. Again, all measures are taken to be in centimetres implying that no conversion is required between the statements that are used to display model components on screen or on the printer.

In addition to the regular graphical operations, the printer driver also provides an easy way to make charts and graphs on the printer.

**Interactive modelling tools**

Although a modelling support environment can help to solve some problems that exist with a manual modelling procedure, it also reduces the flexibility of the designer as it imposes restrictions regarding visibility of a design, i.e. small screens, and of modelling activities, i.e. everything should be specified through the keyboard and a mouse. In our implementation of the entity modeller and task modeller tools we therefore emphasized the ease of modelling. Using the units discussed above, the modelling tools are implemented in such a way that, where possible, model manipulation can be performed by simple and fast mouse operations (Dur 1992).

As an example, we discuss the task modeller (figure 4.11) in some further detail. In the ideal situation, the analyst can, when communicating with the people in the organization, create and change task descriptions on the spot. The problem
with this, however, is the fact that the graphical display is most often too small to do so effectively. The advantages of using flip overs and large A3 sheets of paper can hardly be met by these displays. By providing adequate zoom facilities, however, some disadvantages can be surmounted. During the actual entry of the model, a detailed view of one part of a task structure might be required, where everyone is able to read all the texts clearly. By using the 'zoom out' (zoom out) button, perhaps several times, the view can be transposed to an overview, in which the general layout of the task structure becomes visible. Afterwards, the detailed view can be retained by using the 'zoom in' (zoom in) button. Although this will still not be compatible with the modelling freedom that can be attained by using paper and pencil, such facilities can certainly compensate the disadvantages to a large degree.

A similar example pertains to the hierarchy of specified tasks which is maintained by the task modeller. In the editor one of the subtasks in a structure can be selected and its structure can be opened by selecting the 'zoom subtask' (zoom subtask) button displayed left in the editors window. Similarly, the structure of the supertask, i.e. the task of which the task currently edited is a subtask, can be opened by selecting the 'zoom supertask' (zoom supertask) button. In the case that the current task is a subtask of more that one other task, a pick-list is presented from which the desired supertask can be selected. Using these buttons, the designer can wander around through the task hierarchy by a sequence of simple button presses.

**Run-time simulation library**

The final category of units pertains to the library of simulation features required to execute a model system. For this purpose, four units are used, as depicted in figure 4.20. In addition to these four units specially devised for simulation purposes, the units Core and Scanner are also used.

![Figure 4.20 Units devised for model system execution](image)
The unit Core is used to allocate the required memory for generated entity instances during the simulation. The unit Scanner is used to enable user input during simulation. An example of such a user input is when the simulation speed can be altered by pressing certain keys or by using the mouse device. Scanner is also used to project possible status lines during simulation. Especially the available memory and disk capacity can be of interest at this time.

The bottom three units mentioned in figure 4.20 are Stoch, Events and Stats. Stoch is a unit which has a number of distribution functions. Also, different algorithms for generating random numbers are provided by Stoch. These generators are again represented as object classes. If two independent streams of random numbers are required, two instances of this class can be created and a separate random seed can be assigned to them.

The unit Events offers the sequence control mechanism, or event list, which is required during simulation. Due to the fact that each event has a time stamp attached to it, several time and date conversion functions are also provided by the unit.

Finally, the unit Stats provides the object class Statistics by which statistics can be maintained during simulation. Observations can be added to the statistics during the simulation. The number of observations and the mean and variance can be determined at any time.

The Agenda unit contains the implementation of the agenda mechanism for actors during simulation. The actor behavior on specific events, as was explained at the end of the previous chapter, is represented in this unit. For this purpose the procedures AddWorkToAgenda and FinishCurrentJob are implemented. For a thorough explanation of the Agenda unit we refer to Dur (1992).

With the Agenda unit, an animated simulation of a model system can be generated. To limit the implementation efforts required, the number of views that can be used in such an animation is limited to one. In this view, an actor can be visualized by its name and the name of the task he is currently working on (see figure 4.12). It is also possible to represent the actor by a graphical icon, which can change according to the status of the actor. In practical use, however, these icons may distract the attention of the participants and hence are not preferred on all occasions. Also, the simulation speed is significantly lower when such icons are used.

Within our implementation of the DMSE, an executable model system is attained in the following way. First, the entity modeller and task modeller generate the appropriate source code for the simulation program. This code is then linked to the Agenda library and an executable model system is generated. Treatment conditions can be passed to the model system by means of parameters. Hence, the simulator in our implementation is merely a shell which translates an experimental design, into a call of the executable model system with the right parameter values. The output of the model system is stored in a data file which can be imported in many existing data base management or spreadsheet packages.
4.4 Summary and Conclusions

In this chapter, the specifications of a support environment (DMSE) for the dynamic modelling approach outlined in the previous chapter have been given. To attain these specifications, three major activities were performed:

1. First, an analysis of the business reengineering process was made. A thorough understanding of this process is crucial for deriving the correct requirements for an environment that is to support this process. The analysis was represented in a task structure, identifying all the different modelling activities, and their ordering.

2. Secondly, the functional requirements for the support environment were derived from the analysis of the modelling process. For each task identified in the task structures, the required support was assessed. For the process as a whole, a total of seven support tools were identified. For each of these tools, the functional specifications were given.

3. Finally, a number of implementation issues were discussed. For the most important support tools, a detailed description was given concerning the way in which their functional requirements can be met. Special attention was given to the user interface of the modelling tools and the ease with which an executable model system can be attained.

As a result of this analysis of the dynamic modelling support environment, a number of conclusions can be drawn.

The use of task structures to represent the analysis of the modelling process, can be seen as a first field test to its useability for representing processes in general. Features which were regarded specifically useful are the different levels of abstraction that can be distinguished, and the illustration of the possible flows and iterations in the processes modelled.

The task structures devised for the modelling process can be seen as an integral part of the support environment. These structures help to determine which activity should be performed when. This, combined with the cross reference table depicted in figure 4.14, helps the analyst to determine at what time, and for what purpose each support tool can be used.

Finally, the implementation of the dynamic modelling environment which was described in the last section demonstrates the feasibility of the support outlined in this chapter. As such, this chapter provides the first preliminary evidence for our fourth research hypotheses stated at the end of chapter 2.
5. Case 1: A governmental department

5.1 Introduction

The first case study was performed at a department within the Dutch Ministry of the Interior. As indicated in our research approach in chapter 1, this first experiment was performed in parallel to the development of the modelling techniques and the modelling support environment. During this first case study different external representations and different prototypes of the DMSE have been used (Dur and Sol 1989, 1991) and evaluated, resulting in their final form as described in the previous two chapters.

The department concerned is the Department for Unemployment Benefits (D.U.B.) which administers regulations set up to provide compensation payments (a percentage of last earned wages) to former civil servants who have involuntarily become unemployed. The department's main activity is to judge new benefit applications and to maintain and update all the information that is required to determine the correct amount for each monthly payment to existing clients. Essential other activities include a variety of contacts with other organizations with respect to social security and tax payment issues. Furthermore, the monthly expenditures are charged to the former employers, i.e. the different ministries. Finally, for every update in legislation concerning one of the benefit regulations, measures have to be taken by the department to make sure that the proper changes are made to both the working procedures and to the system that calculates the monthly payments.
The procedures that are followed by the department have been in effect since the late 1970s and can be characterized by a number of observations:

1. The department has a strong hierarchical structure. For each regulation a separate office exists comprised of a head and several subordinates. The work is divided amongst these subordinates by the head of office, who also checks their work. Furthermore, due to the large sums of money involved (the department has a turnover of DFL 400 million on an annual basis), a thorough check is performed on all the activities by a separate verification group.

2. The computer system used by the department is a batch oriented system with a production cycle of one month. All the updates in information are sent to a central computing department, which calculates the monthly payments and initiates these. When a change in the information of a client becomes available at the end of the month, and can not be processed in time by the computing department, a manual payment needs to be performed by the regulation offices. Such manual payments are reported to the system in the next month.

3. Normally, the computing center calculates the monthly payments for each client automatically, based on the available client data. On some occasions, however, such calculations have to be performed manually. These calculation are required for the manual payments mentioned under (2), as well as for clients who make inquiries concerning their benefit prior to the monthly system’s run. Also, any changes in a client’s data which are made retroactive can not be dealt with by the computer system, implying that manual calculations are required.

The rather unwieldy way of working at the department, which is a result of the thorough checking procedures and the strict delivery deadlines imposed by the computer system, has invoked problems throughout the 1980s. To an increasing extent the monthly payments were performed late, the processing of benefit applications was postponed, and secondary activities like an annual report for the Dutch treasury were neglected. Although the problems were indicated by the department as early as 1981, it was not until 1988 (O&I 1988) that action was taken to improve the situation. By this time the number of clients had increased significantly as a result of the privatization and curtailment of governmental departments throughout the 1980s. As a result of this increase serious congestions in the department’s processes developed, making action inevitable.

In 1988 it was estimated that 300 payments (5%) were performed manually, and hence late, each month. Manual calculations were required for every 1 out of 13 clients. Such calculations can take anywhere from 20 minutes to an entire day, implying that their claim on the working capacity was intolerably large.
Indicative of the problems within the department is the fact that questions and complaints were raised by clients in some 8000 phone calls each month. Besides late payments, these phone calls were also caused by the lack of information provided to clients. For instance, when a benefit application is approved, a letter is sent to the client. This letter is made up at an interdepartmental typing room, which can have a backlog of up to 8 weeks. Hence, clients receive payments before they even have any information concerning their attributed benefit, invoking them to call for information.

In 1988 a project was started with the objective of increasing the productivity of the regulation offices. It was estimated that an increase of at least 25% was required in order for these offices to be able to properly cope with the amount of work assigned to them. In the project a new computer system was proposed as a solution to the problems perceived. For this new system an extra budget was claimed of DFL 4.8 million for 1988 and 1989, on top of the department’s regular annual I.T. expenditures of DFL 2.5 million (O&I 1988).

To evaluate the effect of the proposed system a model of the existing situation within the D.U.B. is devised and analyzed. Apart from enabling the analysis of the solution proposed by the department itself, alternative changes can also be designed and analyzed using such a descriptive model. In subsequent sections we discuss the specification of the descriptive model of the D.U.B., the validation of this model, diagnosis and solution finding and the evaluation of solution alternatives.

5.2 Specification

As was indicated above the primary concern of the D.U.B. is to acquire and maintain the information of clients which is necessary to administer the regulations regarding unemployment benefits for former civil servants. In this section entity class definitions and task structures are specified which make explicit how the D.U.B. tends to this responsibility.

Before we turn to the specification of the model itself, a demarcation of the area of interest is required. The essential problem indicated in the introduction is the fact that regulation offices can not adequately process their work within the time and quality constraints imposed on them. Hence, these regulation offices will be the main focus of the analysis. In order to understand such offices correctly, however, their communication with other departments and organizations also needs to be made clear. For such related departments and organizations a specification is only given for those aspects that enhance our understanding of the regulation offices. For the regulation offices themselves and for the verification groups associated with them a more detailed specification is given.
In our specification we start with a description of the department’s clients which trigger the different processes within the D.U.B. by inputting various kinds of information.

Information can be obtained from a client by phone, but is in most cases exchanged by written correspondence. All correspondence which is sent to the D.U.B. is recorded and sorted out per office by Registration. Such written correspondence includes benefit applications and a variety of forms by which changes in a client’s information are relayed. In addition to this, many clients are obliged to return an account of their current income each month. For this purpose so-called income statements are used. Based on the above, the following entity classes can be identified:

\[\text{class Client} \leftarrow \text{simple actor}\]
\[\text{attributes}\]
\[\text{personal data}\]
\[\text{wages earned elsewhere}\]
\[\text{attributed benefit}\]
\[\text{skills}\]
\[\text{apply for unemployment benefit}\]
\[\text{send income statement}\]
\[\text{send correspondence}\]
\[\text{call for information}\]
\[\text{end class}\]

\[\text{class Registration} \leftarrow \text{simple actor}\]
\[\text{attributes}\]
\[\text{skills}\]
\[\text{register mail and sort per office}\]
\[\text{end class}\]

\[\text{class Benefit application} \leftarrow \text{simple item}\]
\[\text{attributes}\]
\[\text{date}\]
\[\text{client}\]
\[\text{regulation}\]
\[\text{former employer}\]
\[\text{additional information}\]
\[\text{end class}\]

\[\text{class Income statement} \leftarrow \text{simple item}\]
\[\text{attributes}\]
\[\text{client}\]
\[\text{month concerned}\]
\[\text{earned wages}\]
\[\text{end class}\]

\[\text{class Correspondence} \leftarrow \text{simple item}\]
\[\text{attributes}\]
\[\text{client}\]
\[\text{date}\]
\[\text{subject}\]
\[\text{contents}\]
\[\text{end class}\]

\[\text{class Phone call} \leftarrow \text{simple item}\]
\[\text{attributes}\]
\[\text{client}\]
\[\text{date and time}\]
\[\text{subject}\]
\[\text{end class}\]

Registration forwards all the correspondence to so-called regulation offices. These offices serve all the clients with respect to one specific regulation. In each

92
office, some ten regulation clerks process the incoming correspondence and phone calls. Each of these clerks can have up to 900 clients assigned to him, depending on the clerk’s skill level.

The work of the regulation clerks is checked by a separate verification group. If any errors are found, these are written down on an error statement which is handed over to the head of office who signs it and sends it on to the accountable clerk. In addition to a thorough check by the verification group, the final decision with respect to a new benefit application is also checked by the head of office.

The entity classes that are associated with the regulation office can now be specified:

```plaintext
class Regulation office ← compound actor
  attributes
    name of regulation
end class

class Regulation clerk ← simple actor
  attributes
    name
    skill level
  skills
    process benefit application
    process income statement
    process other correspondence
    answer telephone call
end class

class Head of office ← simple actor
  attributes
    name
  skills
    sign decision on attributed benefit
    sign error statement
end class
```

If a change is required in the information of a client, regulation clerks specify this on a transaction form which can be processed by the computing center. The form, along with the client’s dossier, is sent to the verification group which checks its correctness. As indicated above, any error that is found is reported back to the head of office. Correct transaction forms are sent to the computing center, where they are processed. The computer output is sent back to the D.U.B., where the verification group takes care of sorting it and distributing it to the different offices.
Again, a number of entity classes can be specified based on the above. We note that members of the verification group are called verifiers:

```plaintext
class Verification group ← compound actor
  attributes
    Name of regulation(s)
end class

class Verifier ← simple actor
  attributes
    name
    skills
      check correctness of work
      receive and sort computer output
end class

class Computing center ← simple actor
  attributes
    skills
      process transaction forms
end class

class Transaction form ← simple item
  attributes
    client
    date of change
    content of change
end class

class Batch of forms ← compound item
  attributes
    month
    number of forms
end class
```

The batch of forms serves as a compound item in which the transaction forms can be stored until the 15th day of the month, at which time the entire batch is transferred to the computing center in order to be processed.

In addition to the departments mentioned so far, the regulation offices also communicate with the investigation department and with finance.
The investigation department is primarily involved with research pertaining to abuse of benefit regulations by clients, but also administers clients who have a so-called inactive account. Such clients do have a benefit attributed to them, but the amount of the monthly payments is DFL 0.00. This may be the case for a number of reasons the most important of which is the height of their current income. As soon as a regulation clerk finds out that one of his clients has become an inactive account, this client is delegated to the investigation department.

Finance, finally, is involved with the manual payments that are performed by regulation clerks in situations where the information could not be made available to the computing center in time. Such payments are sent to finance which takes care of the fund transfers and notifies the computing center in the next month.

For both departments additional entity classes are specified. We will only specify the skills that are required for describing the work of the regulation offices, as this is the scope we are interested in.

```plaintext
class Investigation dept. ← simple actor
  attributes
  skills
    administer inactive account
end class

class Finance ← simple actor
  attributes
  skills
    make manual payment and dispatch
      transaction form for run next month
end class

class Manual payment ← simple item
  attributes
    client
    amount
end class
```

With these additional entity class definitions the entity model is complete. In figure 5.1 a summary of the model is given, where a graphical representation of entity classes is used, and the physical information flows are depicted.

The regulation office and the verification group are located in the box in the center of the figure. The forms that are transferred within this box illustrate the massive flows of documents that exist between these departments. The regulation offices and the verification groups are located on different floors in the same
building, so one can imagine the effort that is required for retrieving, transporting and filing the different documents each month.

![Diagram of information flow]

**Figure 5.1** Overview of defined entity classes and information flows

Although a preliminary overview of information flows is already contained in figure 5.1, their nature will be made clear in more detail by the task structures which
represent the main processes within the D.U.B.. These processes are made up of subtasks which have already been identified as skills in the entity model presented above. Again, a distinction is made between the different kinds of correspondence that are received by the D.U.B. as indicated in figure 5.2. The incoming mail is registered and sorted per office. The mail is then sent on to the responsible regulation clerk, who can choose the appropriate task for dealing with the mail, depending on its type.

![Figure 5.2 The task 'Receive and process correspondence'](image)

In addition to the written correspondence, regulation clerks also answer phone calls made by clients. Hence, in total four types of work are identified for regulation clerks. Below, each of these types is discussed separately.

1. **New benefit applications**

Benefit applications are checked by the regulation clerk for completeness. All the appropriate forms have to be filled out before any further action can be taken. If additional information is required the client will be asked to provide this. In most cases the information has to be sent in by mail, implying that further treatment is delayed until the additional information has arrived.

Based on all the information that is provided the amount of benefit can be determined and a scheme can be set up according to which the payments will be performed. The final decision concerning the ascribed benefit is reviewed by the verification group. If errors were made, an error statement is used to describe these. The client’s file and the error statement are then passed on to the head of office, who signs the error statement and instructs the regulation clerk on how to correct it.

If the payment scheme is correct, the regulation clerk archives the file. The head of office checks the final payment scheme and signs it for his consent. Next, a letter is written by the regulation clerk to inform the client about the decision that was made.
Figure 5.3 The task 'Process benefit application', actor: Regulation clerk

If the application is accepted the amount of benefit determines the appropriate action. If the benefit equals DFL 0.00, implying that the client has an inactive account, the client is transferred to the investigation department which then takes over the responsibility. Otherwise, the client’s information is sent to the computing center which will take care of the appropriate payments. During the month, the forms are piled up at the D.U.B. and transferred to the computing center on the 15th day of the month. As can be seen in figure 5.3 the join until symbol is used to denote this procedure.

The task structure for processing new benefit applications is depicted in figure 5.3. Although the subtask 'make monthly update' is a compound task, we will not discuss it at this time because it is also contained in the structures for processing income statements and other correspondence.
2. Income statements

The second type of incoming mail distinguished in figure 5.2 is that of income statements. We call to mind that these statements are used by clients to inform the D.U.B. of their earned wages each month. The amount that is payed each month depends on the client’s income, implying that these income statements are crucial for making the correct payments.

Before we discuss the way in which income statements are processed, some specifics about the computer system have to be explained. As we stated in the introduction the computer system of the D.U.B. is a batch oriented system with a production cycle of one month. The system is intended to calculate the correct amount for each client each month, and to initiate the appropriate payments. These payments are sent by tape on the 24th day of the month. In order to make this tape ready, the computing center has to have the up-to-date information available by the 18th day of the month. All the transaction forms that are made during the month are sent to the computing center on the 15th day. The computing center then enters these forms and checks their correct coding. On the 18th day the verification group has the ability to make on-line corrections for errors that may have been found in the transactions. From that time on, the computing center will no longer accept changes. Transactions which still contain errors are rejected and sent back to the D.U.B..

The delivery dates imposed by the computing center play an important role in the processing of income statements. For all the clients who did not send in their income statement before the 15th of the month, an instruction is given to the computing center to block their monthly payment. For this purpose, special blocking cards are already available for each client which only have to be sent to the computing center on time.

If income statements are received in the next few days, before the final closing date on the 18th, the order to block a payment can be cancelled by means of an unblocking card. If the income statement contains changes, however, these unblocking cards are of no use because changes can not be made with these cards. Therefore, any change in the client’s information implies that a manual payment needs to be performed.

For income statements that are received on the 18th day or later, a manual calculation has to be made, regardless of any changes being made or not. For all these income statements the monthly payment is performed by means of a manual payment by the finance department.

The process for income statements is represented by the task structure in figure 5.4. As can be seen in this task structure, the regular procedure for creating a transaction form is followed if the income statement is received before the 15th day of the month. For income statements that are received before the 18th day, a manual calculation is only required for those income statements that report a change. After the 18th day every income statement requires a manual calculation and a payment by finance.
Figure 5.4 The task 'Process income statement', actor: Regulation clerk

All the manual calculations that are made by a regulation clerk are checked by the verification group. Again, any errors are reported back to the head of office who will instruct the regulation clerk on how to revise his work.

The subtask 'create and process transaction form' will again be discussed later, as it will also be contained in the task structure for processing other correspondence.

3. Other correspondence

The term 'other correspondence' is used to indicate a broad range of forms and letters that are received by the D.U.B.. Such correspondence may be used to indicate a change in address, but may also imply changes which have an impact on the amount, or even the right to an unemployment benefit.

The task structure for processing other correspondence is depicted in figure 5.5. The regulation clerk starts to determine the changes that are reported in the client's information. If these changes imply that the monthly payments will be reduced to
DFL 0.00 the account is transferred to the investigation department. Otherwise, the procedure for making a transaction form is followed.

![Diagram](image)

**Figure 5.5** The task 'Process other correspondence', actor: Regulation clerk

4. Telephone calls

Besides dealing with the incoming correspondence, regulation clerks also answer telephone calls that are made by clients. As indicated in the introduction, the number of calls received by the D.U.B. has grown throughout the 1980s, and was estimated to be 8000 per month in 1988. At the regulation offices, special consulting-hours are set up to prevent the regulation clerks from being entirely occupied by these calls.

In most cases clients call to make inquiries about their payments. On some occasions, however, clients may inform the D.U.B. of changes that occurred, and hence a procedure similar to the one projected for 'Process other correspondence' is required. The task structure for answering phone calls is depicted in figure 5.6.

![Diagram](image)

**Figure 5.6** The task 'Answer telephone call', actor: Regulation clerk

101
Having discussed the task structures for handling the three types of correspondence and telephone calls, the subtasks for making a transaction form and for making the monthly updates can be described.

In the process of making a transaction form, the delivery dates of the computing center are again important. After the regulation clerk has devised a transaction form, it is sent to the verification group. If the transaction form is received on the 13th or 14th day of the month, i.e. if they can not be checked before the 15th, and if it includes changes that need to be reported to the computing center, the verification group will send these on to the computing center without checking them. These transactions are checked and possibly corrected during the on-line session on the 18th day.

If the transaction forms are received in time, however, they will be checked first by the verification group, before they are sent to the computing center. Again, any errors found are reported back to the head of office, and corrected by the responsible regulation clerk. If no errors are found, the file is sent back to the regulation clerk who archives it. If the transaction form includes changes for the computing center, the form is put on the stack of forms and transferred to the computing center on the 15th day of the month. As can be seen in figure 5.7, the join until symbol is again used to denote this procedure.

**Figure 5.7** The task 'Create and process transaction form', actor: Regulation clerk
The final subtask that needs to be explained is 'make monthly update' which is performed by the computing center. The stack of forms is entered and checked. On the 18th day the verification group has the opportunity to check and correct some of these transactions during an on-line session. Next, the transactions are actually put into effect and an update of the data base is made. The monthly payments to clients are collected on a tape. The refused transaction forms are sent back to the D.U.B., where the verification group sorts and distributes them. For each incorrect form a correction is made for the run in the next month. The payment for the current month is calculated manually and an order for a manual payment is issued to finance.

![Diagram](image)

**Figure 5.8** The task 'Make monthly update', actor: Computing center

In figure 5.8, the split symbol is used to indicate that the stack of rejected transactions is subdivided into individual pieces, and that for each of these individual transaction forms a task is performed by the regulation clerk and by finance.

To construct the task structures discussed above, a two phase approach was used. First, existing documentation was used to specify the current procedures within the D.U.B. in order to minimize the required time spent by D.U.B. employees on interviews. The constructed model was then presented to a group of 10 individuals. These individuals were members of different regulation offices and verification groups and represented the work force of the department as a whole.

In a two day session misinterpretations were pointed out by the group and the model was improved on a wide range of aspects. The session demonstrated the ease of using task structures, even for people who are not trained in using abstract
models. Without extensive prior explanation the group members were able to point out errors easily.

With the improvements proposed during the model verification session, an adequate static model of the D.U.B. is available. Based on this description a number of conclusions can already be drawn:

1. The deadlines imposed by the computing center have a significant impact on the way in which transaction forms are dealt with by the D.U.B.. With the closing date on the 15th day the regulation offices actually have less than two weeks to perform their work. In these two weeks new applications are neglected due to the large amount of time they require. Income statements and other correspondence are treated with higher priority in order to preclude the manual calculations that are required when these are not finished in time.

2. Thorough checking procedures are followed not only for new applications but also for all the other work performed by the regulation offices. These procedures further decrease the amount of time that is available for the regulation offices to perform their work. It is striking to see that the verification of transaction forms is abandoned after the 13th day of the month (left hand side of figure 5.7) and is replaced by a quick check during an on-line session on the 18th day. Errors that are found during this session are immediately corrected by the verification group itself without any feed back to the regulation offices.

Apart from these findings the task structures demonstrate the fact that many different people and departments are involved in each process implying that clients' files need to be transferred quite often. However, to deepen our understanding of the department, its problems and their causes, more needs to be known about the way in which applications, forms and phone calls are dealt with through time. What work load is common for each department and how do these departments attempt to finish these within the quality and time constraints imposed on them? To answer these and other questions, a dynamic representation of the D.U.B. is called for.

A dynamic model system can be attained from the static model by specifying the appropriate input data such as priorities and delays for the identified tasks. Up till now we were able to use a single model to represent the different regulation offices within the D.U.B. due to the fact that no differences exist in the procedures they use. As far as priorities and time delays are concerned, however, differences exist between different offices. Therefore, the specification of a model system is performed for one regulation office specifically. The office that is selected for this purpose is called UR '66 which is responsible for one of the major regulations administered by the D.U.B..
Apart from selecting a single regulation office as our scope of analysis, we also focus our attention on one process in particular: the processing of income statements. This decision is prompted by the fact that the process for handling income statements is very time critical, and the monthly volume of such statements is very large. For the tasks related to treating new applications, other correspondence and phone calls, estimated data will only be incorporated in the model system in order to obtain an accurate level of occupation for the actors involved.

In table 5.1 the amount of correspondence is listed for the D.U.B. as a whole and for the UR '66 office separately. As indicated by this table, more than 30% of all the correspondence and phone calls received by the D.U.B. is intended for the UR '66 office. To process these forms and calls, the office has 10 regulation clerks who are each responsible for an average of 700 clients.

<table>
<thead>
<tr>
<th></th>
<th>total D.U.B.</th>
<th>U.R. '66</th>
</tr>
</thead>
<tbody>
<tr>
<td>new applications</td>
<td>800</td>
<td>300</td>
</tr>
<tr>
<td>income statements</td>
<td>4100</td>
<td>1360</td>
</tr>
<tr>
<td>other correspondence</td>
<td>1700</td>
<td>500</td>
</tr>
<tr>
<td>phone calls</td>
<td>8000</td>
<td>2800</td>
</tr>
</tbody>
</table>

Table 5.1 Monthly average amount of incoming correspondence and phone calls

Benefit applications and other correspondence arrive at a steady rate throughout the month. This is also true for the phone calls. Income statements, however, have a special frequency distribution. Most of these statements are received between the 3rd and 6th working day of the month. From that time on, the number of forms received each day decreases gradually. A very small portion of the income statements is received on the first days of the next month. In figure 5.9 this frequency distribution is depicted for the UR '66 office. In this figure the number of working days per month is set to 20.

Due to the fact that the transaction forms that may result from processing an income statement have to be sent to the computing center by the 15th day of the month (i.e. the 12th working day of the month) the regulation office spends most of its time during the first two weeks on processing these income statements. Benefit applications and other correspondence are postponed, and mainly dealt with in the last two weeks.
Figure 5.9 Distribution of received income statements for the UR '66 office

In a number of interviews with regulation clerks and verification group members, the delays and priorities for their different skills were assessed. As illustrated by the income statements, these priorities shift throughout the month due to the strict delivery dates set by the computing center. By using several independent assessments a reasonable accuracy was established.

In addition to the delays and priorities, the probability of some of the alternative routings in the task structures were determined. For example, when the verification group checks the work of a regulation clerk, the probability of finding an error needs to be determined. Some of these probabilities which were incorporated in the model system are listed in table 5.2.

<table>
<thead>
<tr>
<th>Event</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>accepted benefit applications</td>
<td>96 %</td>
</tr>
<tr>
<td>errors found by verification group for applications</td>
<td>15 %</td>
</tr>
<tr>
<td>errors found by verification group for other work</td>
<td>7 %</td>
</tr>
<tr>
<td>transaction forms refused by computing department</td>
<td>6 %</td>
</tr>
<tr>
<td>portion of inactive accounts</td>
<td>1 %</td>
</tr>
</tbody>
</table>

Table 5.2 Probabilities for some events in the UR '66 office
With the delays, priorities and probabilities estimated during interviews, the
model system for the UR '66 office has been constructed. With this model system
several experiments were performed for verification purposes. These experiments
were used to ascertain the correct specification of the generators for the income
statements as well as for the other types of work. Also, tests were performed to
ascertain that delays and priorities specified in the model were consistent with the
designer's intentions.

In addition to the test regarding the correct specification by the designer,
experiments were also performed to demonstrate the behavior of the model to the
participants in the organization. In one of these tests a single income statement was
generated and followed in its course through the model. For this purpose an
animated simulation was used, examples of which will be given when we discuss the
structural validation below.

With the specification and verification of both the static model and the dynamic
model system, the first step in the modelling process is finished. The validity of the
resulting model system still needs to be checked, however. This validation step is
discussed in the next section.

5.3 Validation

As discussed in chapter 4, the correctness of a model system can be determined by
means of a structural and a replicative validation. In this section, both are discussed
for the model system specified above.

The objective of a structural validation is to determine whether the behavior of
the specified model system in a certain situation is consistent with what one would
expect it to be. For this purpose, treatments have been devised for the model system,
and a series of experiments has been performed. For instance, an experiment was
performed to see what would happen if no regulation clerk were available. Evidently,
one would expect the work to pile up at the regulation office. Similarly, the behavior
of the model system has been checked when 100 regulation clerks were available in
stead of the regular 10. In this situation, the work load at the regulation office should
decrease significantly in comparison with the current situation. Checks like these
have been performed to increase the confidence in the model system devised.

For the structural validation, an animated simulation of the model system was
used which is generated by the prototype support environment used in the project
(Dur 1991). Such an animation consists of an overview of all the subsequent tasks
which are linked by intermediary work loads. The work loads are depicted by a bin
symbol (□) which is filled with the correct amount of items. As simulation time
proceeds, the up and down movements in each work load can be inspected. In figure
5.10 a projection of such an animation screen is given.
For each test in the structural validation, the fluctuation of the work loads during simulation was discussed with the participants. Each time, the behavior of the model was compared with the expectations the group had beforehand. In addition to the animation itself, a report of each work load for the simulation time which has passed so far can be obtained. An example of such a report is given in figure 5.11 where the work load for the verification group is depicted during a period of 7 months. The report also shows some additional statistics in a separate panel on the right, as well as projections of the walking averages for the waiting time in the work load and for the life span of income statements so far.

Following the structural validation session, a replicative validation has been performed for the model system. Similar to the structural validation, a treatment for the model system was first devised.

To begin with, an estimate of the model system’s start-up time is required. To determine this start-up time a trace of an output or state variable can be made. By projecting such a traced value against the simulation time, one is able to determine when such a value starts to stabilize, implying that the models transient period has passed.
In figure 5.12 the average life span for income statements that were sent to the computing center on time, as well as for those that were finished late, is projected. Also, a projection is given of the average waiting time for income statements that arrive at the UR '66 office. In the model system of the UR '66 office the number of working days per month is set to 20, implying that the total simulation time depicted in figure 5.12 is equal to 5 months. The line charts show the moving average of the variable as simulation time proceeds. The dots associated with these line charts depict the average values for the forms that were processed in that month, i.e. where the samples for previous months are neglected. Based on the figure we determined that between 60 and 80 days simulation time, a stable situation is obtained. We choose a start-up time of 80 days to be on the safe side.

The next value that is required is the run length. This is the amount of simulation time during which the output of the model system is measured. For each replication this run length is added to the start-up time in order to determine the total amount of simulation time.

A rule of thumb for determining the run length is to take three times the length of the longest cycle in the model system. For the D.U.B. model this cycle length would be one month. Hence, the run length is set to 3 months, which equals 60 days.
Case 1: A governmental department

![Graph showing traced variables projection to determine start-up time](image)

**Figure 5.12** Projection of traced variables to determine start-up time

The output that we require from the model system are average life spans for income statements that are finished in time, and for those which are finished late. Also, the number of forms in each of these categories is of interest to us as the percentage of forms finished late is a good measure for the overall performance of the regulation office as a whole. In addition to these outputs, average waiting times at intermediary departments in the process are of interest when the causes for perceived problems are sought.

With the start-up time, run length and model output determined so far, the number of required replications can be determined. Such a calculation can be made by choosing the desired risk \( \alpha \), and the width of the \((1-\alpha)\) confidence interval for the output variables selected (Van Soest 1985; Thesen and Travis 1992). The number of replications is determined by the following procedure:

1. Establish an acceptable level of risk (say: \( \alpha = 0.05 \))

2. Draw a sample of \( N_0 \) observations of the output variables by performing \( N_0 \) replications
3. Compute the sample variance $s^2$ for each output variable.

4. Establish an acceptable width $\delta$ of the $(1-\alpha)$ confidence interval for each output variable (say: $\delta$ equals $\frac{1}{3}$ of the interval between the lowest and highest value in the sample of $N_0$ observations).

5. Determine the required number of replications $N$ with:

$$N \geq \frac{4 t^2_{\alpha, N_0-1} s^2}{\delta^2}$$

where $t_{\alpha, N_0-1}$ is the appropriate value taken from a table of the $t$ distribution.

For each of the output variables an additional sample of $N-N_0$ observations should now be drawn. With a confidence of $1-\alpha$ the actual mean value $\mu$ of the output variable can be found in the interval:

$$\bar{x} - \frac{1}{2}\delta \leq \mu \leq \bar{x} + \frac{1}{2}\delta$$

where $\bar{x}$ is the sample mean for the $N$ observations.

<table>
<thead>
<tr>
<th></th>
<th>obs</th>
<th>min</th>
<th>max</th>
<th>$\delta$</th>
<th>stand. dev.</th>
<th>replications required</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean waiting time upon entry</td>
<td>10</td>
<td>1.97</td>
<td>2.51</td>
<td>0.18</td>
<td>0.186</td>
<td>22</td>
</tr>
<tr>
<td>mean waiting time at verification</td>
<td>10</td>
<td>1.29</td>
<td>1.40</td>
<td>0.04</td>
<td>0.035</td>
<td>19</td>
</tr>
<tr>
<td>mean waiting time for correction</td>
<td>10</td>
<td>1.04</td>
<td>1.13</td>
<td>0.03</td>
<td>0.028</td>
<td>18</td>
</tr>
<tr>
<td>mean life span forms on time</td>
<td>10</td>
<td>9.86</td>
<td>10.0</td>
<td>0.05</td>
<td>0.059</td>
<td>25</td>
</tr>
<tr>
<td>mean life span forms too late</td>
<td>10</td>
<td>5.31</td>
<td>6.55</td>
<td>0.41</td>
<td>0.357</td>
<td>16</td>
</tr>
<tr>
<td>number of forms on time</td>
<td>10</td>
<td>2666</td>
<td>2822</td>
<td>52</td>
<td>50.5</td>
<td>22</td>
</tr>
<tr>
<td>number of forms too late</td>
<td>10</td>
<td>1313</td>
<td>1458</td>
<td>48</td>
<td>1.15</td>
<td>24</td>
</tr>
<tr>
<td>percentage of forms too late</td>
<td>10</td>
<td>31.8</td>
<td>35.1</td>
<td>1.1</td>
<td>1.1</td>
<td>21</td>
</tr>
</tbody>
</table>

$\alpha = 0.05 \quad t_{0.025, 9} = 2.262$  \quad \text{Max: 25}$

Table 5.3 Calculation of the number of replications required
Case 1: A governmental department

For the model of the UR '66 office, the procedure described above is performed with a significance level $\alpha=0.05$, an interval width of $\frac{1}{3}$ of the difference between the minimum and maximum value found in the $N_0$ replications, and an initial number of replications $N_0=10$. In table 5.3 the results for this experiment are listed. The waiting time and life span of income statements are listed in units of days.

The number of replication required for this model is 25, implying that an additional set of 15 replications is required. For each of the output variables, the sample mean, standard deviation and 95% confidence interval are listed in table 5.4.

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>s.d.</th>
<th>95 % confidence interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean waiting time upon entry</td>
<td>2.23 days</td>
<td>0.205</td>
<td>[2.14, 2.32]</td>
</tr>
<tr>
<td>mean waiting time at verification</td>
<td>1.35 days</td>
<td>0.028</td>
<td>[1.33, 1.37]</td>
</tr>
<tr>
<td>mean waiting time for correction</td>
<td>1.10 days</td>
<td>0.041</td>
<td>[1.08, 1.12]</td>
</tr>
<tr>
<td>mean life span forms on time</td>
<td>9.91 days</td>
<td>0.075</td>
<td>[9.88, 9.94]</td>
</tr>
<tr>
<td>mean life span forms too late</td>
<td>5.87 days</td>
<td>0.403</td>
<td>[5.66, 6.08]</td>
</tr>
<tr>
<td>number of forms on time</td>
<td>2733</td>
<td>45.3</td>
<td>[2707, 2759]</td>
</tr>
<tr>
<td>number of forms too late</td>
<td>1395</td>
<td>67.5</td>
<td>[1371, 1419]</td>
</tr>
<tr>
<td>percentage of forms too late</td>
<td>33.8 %</td>
<td>1.25</td>
<td>[33.2, 34.4]</td>
</tr>
</tbody>
</table>

Table 5.4 Results derived from the descriptive model system

In the current situation, more than 30% of the income statements are finished after the closing date for the computing center. Forms that are finished in time have an average life span of about 10 days, while statements that are finished late take an average of 6 days to be processed. For this last category the time it takes for finance to perform the manual payments is not incorporated in the statistics.

For a replicative validation of a model system its numerical output should be compared with the corresponding values in reality. Due to the lack of data concerning the actual process, however, such a comparison could not be made using formal statistical techniques. The collection of the required data specifically for our purposes was also difficult due to the limited amount of time that was available for the departments employees to participate in such an effort.

As a consequence, establishing face validity was the best we could do for the UR '66 office model. The model output was presented to the same group of 10 individuals which had already participated in the structural validation session. The verifiers were surprised by the fact that more than 30% of all the income statements
were finished late. Although some initial discussion between regulation clerks and verifiers came about regarding this fact, enough anecdotal evidence was found within half an hour which supported this outcome.

5.4 Diagnosis, solution finding and evaluation

After the validity of the descriptive model system is established it can be used to determine the main causes for problems perceived in the existing organization, and it can also be used as a yard-stick for the significance of these problems. For proposed solution alternatives, the results of prescriptive model systems can be compared with this yard-stick to ascertain the improvements that can be achieved.

The first proposal for a change was submitted by the project team, which had continued its endeavour to design a new computer system in parallel to our specification activities. According to this project team, the major cause for the fact that income statements could not be finished in time was the fact that a significant number of these statements arrive at the D.U.B. after the 14th day (11th working day) of the month.

![Graph](image)

**Figure 5.13** Distribution of income statements in the current and new situation
As a solution to the late arrival of income statements, the project team proposed an automatic reminding system which would send a reminder to a client if the income statement had not arrived on the fourth working day of the month. With such a system, the distribution of income statement over the month should be altered as depicted in figure 5.13.

The effect that is expected from the earlier arrival of income statements is a significant increase in the number of forms that are finished in time. This increase reduces the time required for manual calculations and payments and would hence resolve many of the department's problems. The average waiting time for income statements during the process may increase slightly due to the fact that a larger number of forms is received during the first four working days and these forms cannot be dealt with at once. The life span, however, is not expected to increase much due to the strict delivery dates which preclude life spans longer than 10 working days.

Using a prescriptive model system where the arrival of income statements is altered according to figure 5.13, an experiment similar to the one performed with the descriptive model system is executed. The results of this experiment are summarized in table 5.5 where the mean and standard deviation for the new model output, as well as those determined for the current situation are listed. To compare the current and proposed situations, a \( t \)-test is performed. In the table, the \( t \) value for this test is listed. Finally, the table shows whether or not the difference is significant for significance levels \( \alpha = 0.05 \) and \( \alpha = 0.10 \).

<table>
<thead>
<tr>
<th>Current situation (n = 25)</th>
<th>Statements earlier (n = 25)</th>
<th>( t )</th>
<th>Significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
<td>s.d.</td>
</tr>
<tr>
<td>wait upon entry</td>
<td>2.23</td>
<td>0.205</td>
<td>2.75</td>
</tr>
<tr>
<td>wait at verification</td>
<td>1.35</td>
<td>0.028</td>
<td>1.34</td>
</tr>
<tr>
<td>wait for correction</td>
<td>1.10</td>
<td>0.041</td>
<td>1.09</td>
</tr>
<tr>
<td>life span on time</td>
<td>9.91</td>
<td>0.075</td>
<td>9.95</td>
</tr>
<tr>
<td>life span too late</td>
<td>5.87</td>
<td>0.403</td>
<td>7.25</td>
</tr>
<tr>
<td>forms on time</td>
<td>2733</td>
<td>45.3</td>
<td>2713</td>
</tr>
<tr>
<td>forms too late</td>
<td>1395</td>
<td>67.5</td>
<td>1401</td>
</tr>
<tr>
<td>percentage too late</td>
<td>33.8</td>
<td>1.25</td>
<td>34.0</td>
</tr>
</tbody>
</table>

\( t_{0.025, 48} = 2.011 \)
\( t_{0.05, 48} = 1.678 \)

n.s.: not significant

**Table 5.5** Evaluation of improvements when statement arrive earlier
From the results it is clear that the effect of bringing the income statements in earlier is quite different from the effect that was expected. There is no significant change in the number of forms that is finished in time for the computing center. Evidently, the fact that regulation clerks are fully occupied with processing income statements during the first two weeks makes it impossible for them to augment their production, regardless of the number of statements that is available to them. In fact, the early arrival of forms only increases the waiting time for income statements that arrive at the D.U.B.. On the average, this time increases by half a day.

The results presented above are rather discouraging for the solution that is proposed by the project team so far. The automatic reminder system, which causes the forms to arrive earlier, will not solve the problems perceived by the D.U.B.. Hence, alternative proposals need to be devised.

To come up with alternative proposals, the group of 10 participants spent an afternoon session with the model system of the UR '66 office to determine the main causes for the perceived problems. The animated simulation where the intermediary work loads are visible as simulation time proceeds was very effective for this purpose.

In the simulation of the model system the most striking aspect of the working procedures was the great number of files and forms that are passed on between the regulation clerks, their head of office and the verification group. Regardless of the kind of change that is made in the client’s information, all steps in the process are kept the same. Hence, about 1400 files are sent around for the UR '66 office each month. Each of these files is passed on at least 5 times (each month), but if errors are made this number can increase to as many as 20!

A solution which was proposed by the participants when they saw the model system’s behaviour, i.e. when they were confronted with their own working procedures, was to select those income statements which do not imply a change in the client’s information, and to archive these statements directly without any further processing. In a first estimate, the participants thought that at least 50% of the income statements could be eliminated by such a measure. After a closer investigation it even turned out to be 65% of all income statement, i.e. only 35% of these statements contain changes.

For this second proposal, an alternative model system is made in which the proposed change is incorporated. The only change is an additional evaluation task in the task structure for the task 'Receive and process correspondence'. The new structure is depicted in figure 5.14.

In this structure changes are determined by the regulation clerk. Alternatively, this could be performed by registration which would further decrease the occupation level for regulation clerks. Again, an experiment is performed with the prescriptive model system. The results are summarized in table 5.6, which is organized similar to the previous table.
Case 1: A governmental department

Figure 5.14 New structure for 'Receive and process correspondence'

This time, the effect of the proposed solution is eminent. The percentage of forms finished late is reduced from ±35 to 12%. For the UR '66 this implies that on the average 900 manual calculations and payments are eliminated each month.

<table>
<thead>
<tr>
<th>current situation (n = 25)</th>
<th>treat changes only (n = 25)</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
</tr>
<tr>
<td>wait upon entry</td>
<td>2.23</td>
<td>0.205</td>
</tr>
<tr>
<td>wait at verification</td>
<td>1.35</td>
<td>0.028</td>
</tr>
<tr>
<td>wait for correction</td>
<td>1.10</td>
<td>0.041</td>
</tr>
<tr>
<td>life span on time</td>
<td>9.91</td>
<td>0.075</td>
</tr>
<tr>
<td>life span too late</td>
<td>5.87</td>
<td>0.403</td>
</tr>
<tr>
<td>forms on time</td>
<td>2733</td>
<td>45.3</td>
</tr>
<tr>
<td>forms too late</td>
<td>1395</td>
<td>67.5</td>
</tr>
<tr>
<td>percentage too late</td>
<td>33.8</td>
<td>1.25</td>
</tr>
</tbody>
</table>

\[ t_{0.025, 48} = 2.011 \quad t_{0.05, 48} = 1.678 \]

\textbf{Table 5.6} Evaluation of effects when selecting changed statements (35%) only

116
Hence, the time that was spent on these calculations is now available for processing new applications, and for performing tasks which were neglected before.

Due to the fact that income statements which do not contain changes are immediately stored away the average life span of these statements is reduced significantly with the proposed solution. The life span is further decreased by the reduction of the waiting time at the verification group, which is caused by the reduction of the number of files that are passed to this group.

In table 5.6 the results of the model system are shown when only 35% of the income statement contains changes. Although this was estimated using actual data, it might be rather optimistic. Therefore, a similar experiment is performed for the situation where 50% of all the income statements contains changes. The results of this experiment are summarized in table 5.7.

<table>
<thead>
<tr>
<th></th>
<th>current situation</th>
<th>current situation</th>
<th>treat changes only</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(n = 25)</td>
<td>(n = 25)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
<td>s.d.</td>
</tr>
<tr>
<td>wait upon entry</td>
<td>2.23</td>
<td>0.205</td>
<td>2.24</td>
<td>0.177</td>
</tr>
<tr>
<td>wait at verification</td>
<td>1.35</td>
<td>0.028</td>
<td>1.15</td>
<td>0.018</td>
</tr>
<tr>
<td>wait for correction</td>
<td>1.10</td>
<td>0.041</td>
<td>1.10</td>
<td>0.031</td>
</tr>
<tr>
<td>life span on time</td>
<td>9.91</td>
<td>0.075</td>
<td>5.16</td>
<td>0.115</td>
</tr>
<tr>
<td>life span too late</td>
<td>5.87</td>
<td>0.403</td>
<td>5.71</td>
<td>0.348</td>
</tr>
<tr>
<td>forms on time</td>
<td>2733</td>
<td>45.3</td>
<td>3433</td>
<td>64.0</td>
</tr>
<tr>
<td>forms too late</td>
<td>1395</td>
<td>67.5</td>
<td>709</td>
<td>42.3</td>
</tr>
<tr>
<td>percentage too late</td>
<td>33.8</td>
<td>1.25</td>
<td>17.1</td>
<td>0.83</td>
</tr>
</tbody>
</table>

|                        | t_{0.025, 48}= 2.011 | t_{0.05, 48}= 1.678 | n.s.: not significant |

**Table 5.7** Evaluation of effects when selecting changed statements (50%) only

As is clear from the table the improvements are still significant with a reduction of 50% of the income statements. In this case, nearly 700 manual calculations and payments are eliminated each month.
The final experiment performed with the UR '66 office model combines the two proposed changes. Hence, income statements are received earlier, and 65% of these forms are stored away immediately without any further processing. As is clear from Table 5.8 the results are very similar to the situation where the statements are not received earlier, although a small additional improvement is obtained.

<table>
<thead>
<tr>
<th></th>
<th>current situation (n = 25)</th>
<th>combination (n = 25)</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
</tr>
<tr>
<td>wait upon entry</td>
<td>2.23</td>
<td>0.205</td>
<td>2.68</td>
</tr>
<tr>
<td>wait at verification</td>
<td>1.35</td>
<td>0.028</td>
<td>1.12</td>
</tr>
<tr>
<td>wait for correction</td>
<td>1.10</td>
<td>0.041</td>
<td>1.12</td>
</tr>
<tr>
<td>life span on time</td>
<td>9.91</td>
<td>0.075</td>
<td>4.51</td>
</tr>
<tr>
<td>life span too late</td>
<td>5.87</td>
<td>0.403</td>
<td>6.92</td>
</tr>
<tr>
<td>forms on time</td>
<td>2733</td>
<td>45.3</td>
<td>3635</td>
</tr>
<tr>
<td>forms too late</td>
<td>1395</td>
<td>67.5</td>
<td>489</td>
</tr>
<tr>
<td>percentage too late</td>
<td>33.8</td>
<td>1.25</td>
<td>11.8</td>
</tr>
</tbody>
</table>

\[ t_{0.025, 48} = 2.011 \quad t_{0.05, 48} = 1.678 \quad \text{n.s.: not significant} \]

**Table 5.8** Evaluation of effects of combined changes

The results proved to be very convincing for both the project team and the participants. Based on the results the specifications of the automatic reminder system which was proposed by the project team were altered in order to be able to determine whether or not an income statement has changed in comparison with the previous month. In the mean time, the working procedures are altered by the participants immediately to improve their working situation ahead of the new system which was under development. The progress in the specifications of a new central computer system was already too far, however, to reconsider the investments of DFL 4.8 million. Therefore, a further analysis of the business processes, which might have reduced this investment or increased its effect, was not requested by the project team.
5.5 Summary and conclusions

In this chapter a business reengineering process was discussed for a department of central government within the Dutch Ministry of the Interior. First a detailed description of the current structure and working procedures was made. We demonstrated the use of such a descriptive model for problem analysis. In the solution finding stage, an existing proposal was first evaluated. The effect of the proposed change did not resolve the department’s problem, implying that alternative changes needed to be generated. Based on the understanding that the participants had obtained about the problems and their causes, an effective solution was found.

The conclusions that we draw from this chapter fall into two categories. The first pertains to our dynamic modelling approach and the effectiveness of its use during a business reengineering process. A second set of conclusions relates to the department which is analyzed, and the solutions that were found to its problems.

1. The dynamic modelling approach

Our first and most important conclusion is that the use of entity class descriptions and task structures, supplemented by a mechanism which facilitates the analysis of behavioral aspects of a model, facilitates an effective way to analyze and resolve problems in an existing organization. By using the approach, an effective solution strategy was developed which had not been thought of prior to the modelling effort. This, despite the fact that the project team had already spent half a year on problem analysis activities.

The second conclusion is that the analysis described above demonstrates that a specific organization can indeed be described using the modelling concepts distinguished in our approach. The entity classes provide a clear demarcation of the scope of interest. Task structures have proven to be a very useful and easy to comprehend modelling technique to represent working procedures.

Finally, we found that the alternative models, envisioning the solution strategies, can be regarded as a 'blue print' of the new organization. Its ability to demonstrate the effect of a proposed change is indicative for its value during the implementation of a chosen solution strategy.

2. The department for unemployment benefits

The working procedures at the department are characterized by a high degree of control and a monthly production cycle. It is shown that significant improvements in the productivity can be obtained within the current monthly production cycle. The solution strategy which is the result of the modelling effort is effective and yet easy to implement in the existing organization. Doubts were raised concerning the new computer system which was proposed by the project team. A new system could be
made more effective if its specifications would be derived from the causes for problems perceived in the current situation. A further analysis of the processes for handling new applications as well as other correspondence and phone calls would yield such causes, and hence would result in the specifications for such an effective new system.
6.1 Introduction

The second case study was performed at the Goudse Insurances, which is a medium-sized Dutch insurance firm. Hence, our second case study concerns the private sector. Although many similarities exist between the private and public sector as far as information processing and information policies are concerned (see e.g., Brussaard 1983), the decision making on changes in the organization’s strategies and business processes can differ significantly (Schwenk 1990). This second case study will help to make clear possible similarities and/or differences between both sectors.

With the approaching European common marketplace, Goudse Insurances has chosen to use what Porter (1980) would call a 'market focus' strategy. They have selected insurance intermediaries as their main buyer group and attempt to serve this buyer group better than its competitors can. The main demand imposed by the selected buyer group is a great flexibility in the services provided. Knowing that the large insurance companies will try to conquer a market share merely by dropping their prices due to their advantages of economy of scale, the Goudse attends to that part of the market where special demands are made which can not be met by standard products.

Apart from this required flexibility, however, timeliness of services offered is also essential. Intermediaries face the problem of having to compete with large insurance firms which sell their products directly to the public (so-called 'direct writers'), and, hence, have the ability of offering their services with almost no time delay. As a consequence, intermediaries want to shorten their delivery time of services as well, implying that insurance firms like Goudse need to respond quickly to the requests made by these intermediaries.
In their 1990 annual report it was pointed out that improvements in the service levels for clients and intermediaries is the most important objective for Goudse insurances. A number of areas has been identified where such improvements are possible:

- Reduction of the number of errors made, especially those which can be observed by clients and/or intermediaries.
- Augmentation of specific knowledge and expertise throughout the organization by means of special training programmes.
- Reduction of the time spans for all the services offered to existing or prospect clients.

One of the facilities that can help Goudse Insurances to obtain these goals is the new Assurance Data Net (ADN) which provides fast information exchange between different insurance firms and intermediaries. Since 1990, Goudse has been connected to this network implying that insurance applications and policy mutations can be sent in by intermediaries electronically. This electronic communication not only saves time, but also reduces the number of errors made when interpreting written correspondence, standardizes communication and eliminates redundant data entry. Although the frequency of use is rather low for the ADN network at the moment, its potential should certainly be assessed.

Historically, Goudse Insurances has had a product oriented organization structure. Each specific product has its own department with its own procedures and technical support systems. The need for flexible insurance policies, perhaps custom-made for a single client, implies that a great amount of specific product expertise is required. Hence, the product oriented organization structure can hardly be replaced by a client oriented one, implying that the main question with respect to improving the service level offered to its clients is 'How can the quality and timeliness of the services be improved, given the current departmental structure?'.

For the analysis, a single department is selected. The department concerned is responsible for fire insurances. Its selection is prompted by the fact that fire insurance policies have many similarities with other types of insurances. Hence, the findings can probably be used for the other departments with relative ease, whereas products such as a car, health or life insurance have some very specific characteristics which may disrupt the translation of the analysis outcomes to other departments.

Similar to our first case study, a detailed specification is made of the current situation within the department. This descriptive model is then used to determine potential improvements, and to assess the feasibility and effectiveness of these improvements. In subsequent sections, the specification, validation and diagnosis and solution finding are discussed.
6.2 Specification

The objective of the analysis is to increase our understanding of the way in which the business processes within the department are organized at the moment. The analysis focuses on the different individuals and departments which are involved in these processes. The main purpose of the analysis is to learn how these individuals and departments cooperate in order to provide the services to the intermediaries. Hence, the level of detail at which the analysis is performed will make explicit what the exact contribution of all those involved is. It will not make clear in detail how an assessment regarding an new insurance application or damage claim is made, nor will it describe the expertise which is required for these tasks. The reason for this demarcation of the analysis is the fact that a greater improvement can be expected from changes made to the sequence and priorities of tasks in the process than from changes made to the way in which specific tasks are performed or supported by new technologies. An earlier investigation concerning the use of knowledge-based systems within the fire insurance department has pointed this out (Van Dort 1990).

Having made clear the objective and demarcation of our analysis, the specification phase can start. We begin with the specification of the entity classes which are used to describe the department and its environment. Next, the task structures for the main business processes in which the department is involved are given.

We start with clients and intermediaries which are the main external entities. New insurance applications, damage claims and mutations on existing policies are in most cases sent in by an intermediary. The correspondence about these policies and claims can either be sent back to the intermediary, or it can be sent back directly to the client. During the process, additional information may be required, either from the client or the intermediary. Apart from this, however, the intervention of both client and intermediary in the process is negligible.

Based on the above, the following entity classes can be defined:

class Client ← simple actor
  attributes
  name, client number
  skills
  provide additional information
end class

class Intermediary ← simple actor
  attributes
  name
  skills
  send insurance application
  send mutation
  send damage claim
  provide additional information
end class
class Application ← simple item
   attributes
   personal data client
   type of insurance and conditions
end class

class Mutation ← simple item
   attributes
   client number
   policy number
   information on change
end class

class Damage claim ← simple item
   attributes
   client number
   policy number
   claim
end class

Other external entities are the technical inspectors and insurance experts who can visit a building or a location for which an application or damage claim is received. Their findings are summarized in an expert report which is then used by the fire insurance department to decide whether or not the application or claim is accepted. The inspectors who assess new applications are employed by the firm itself, whereas experts who check damage claims are independent. For our analysis, however, we will use the term expert for both.

class Expert ← simple actor
   attributes
   name
   skills
   devise expert report
end class

The department itself is subdivided into three groups. Two groups process applications and mutations for fire insurance policies. One of these groups handles private accounts, whereas the second deals with corporate accounts. The third group handles damage claims that pertain to existing fire insurance policies.

Each group consists of a head and a number of subordinates. Within the first two groups these subordinates are referred to as 'acceptors'. In the third group they are referred to as 'claim attendants'. The head of each group distributes the incoming mail amongst the subordinates, and manages and supervises their work. The head of the corporate accounts also inspects the mail when it is delivered to the department to see whether important applications or damage claims have arrived. A special task for the head of the claims group is to check and sign all the outgoing mail. Without his signature, a payment can not be performed by the accounting department.
The class definitions for the three group heads are given below:

```plaintext
class Head of corporate accounts ← simple actor
    attributes
        name
        skills
            scan through new applications
            sort and distribute mail to acceptors
end class

class Head of private accounts ← simple actor
    attributes
        name
        skills
            sort and distribute mail to acceptors
end class

class Head of claims group ← simple actor
    attributes
        name
        skills
            sort and distribute mail to claim attendants
            sign outgoing mail
end class
```

The acceptors process new insurance applications as well as mutations for existing insurance policies. They also answer clients’ telephone calls. The work that is assigned to them depends on their level of expertise. Once an application or mutation is assigned to a specific acceptor, this is processed entirely by that person. If an expert assessment is required, this is initiated by the acceptor. Also, the acceptor collects all the additional information he or she requires.

```plaintext
class Acceptor ← simple actor
    attributes
        name
        level of expertise
        skills
            process application
            process mutation
            answer telephone call
end class
```
Claim attendants process damage claims and answer telephone calls which pertain to these claims. Similar to their colleague acceptors, the work which is assigned to them depends on their expertise. A damage claim is processed entirely by the person to whom it is assigned, with one small exception. If an expert assessment is required, the expert report is not processed by the claim attendant himself, but rather by a special expert attendant. The class definitions for both are given below:

```plaintext
class Claim attendant ← simple actor
  attributes
  name
  level of expertise
  skills
  process damage claim
  answer telephone call
end class

class Expert attendant ← simple actor
  attributes
  name
  skills
  process expert's report
end class
```

Apart from the acceptors and claim attendants, the department has a special administrative assistant who prepares the work for others and supports them during the process. The tasks for the administrative assistant include sorting the mail, retrieving printed letters from the printing room, and finding and storing client files. There is only one administrative assistant for the three groups. The class definition is the following:

```plaintext
class Administrative assistant ← simple actor
  attributes
  name
  skills
  sort mail
  find and store files
  retrieve letters from printing room
  sort and prepare computer output
end class
```

Apart from the department’s own personnel, other departments are also involved in the processes of handling applications, mutations and claims. Although these departments also perform work which is not related to the processes of the fire insurance department, we are only interested in those tasks which are related to those processes.

Two departments which are involved are the Mail room and the Central Relations Administration (C.R.A.). The first is responsible for distributing the mail
to the departments and processing the outgoing mail. The second is a central data entry department which registers all new clients and policies. The C.R.A. is also responsible for filming documentation on a client’s micro-film which is used within the different departments. Their definitions are the following:

```plaintext
class Mail room ← simple actor
attributes
  skills
  open and sort the mail
  process outgoing mail
end class

class C.R.A. ← simple actor
attributes
  skills
  register new applications
  register mutation
  film documentation
end class
```

Two other departments involved are Operations and the Premium department. The first takes care of the computer run each night. New insurance policies are printed during this run, and sent to the insurance departments the next morning. On some occasions, the computer is down at night, and the computer output is delayed an extra day. The Premium department checks all the invoices which are printed during the computer run. Definitions for both departments are given below:

```plaintext
class Operations ← simple actor
attributes
  down ratio
skills
  run night batch and print output
end class
```

```plaintext
class Premium dept. ← simple actor
attributes
  skills
  check invoices
end class
```

Finally, there is a special department named Scanner which integrates the correspondence for corporate clients in order to compensate for the product oriented departmental structure. If a corporate client applies for different insurance policies at the same time, the policies and other correspondence are collected from the different insurance departments and sent at once to the client. For each client, a special code is issued by the department and all the outgoing mail is ordered according to these codes. The department is defined as follows:

```plaintext
class Scanner ← simple actor
attributes
  skills
  assign special code
  check registered applications
  collect and combine correspondence
end class
```
With these definitions the entity model for the fire insurance department and the relevant parts of its environment is complete. In figure 6.1 the entity classes are summarized graphically, and some information flows are shown. Similar to the previous chapter, the information flows are clarified in further detail in the task structures representing the main business processes in which the department is involved.

**Figure 6.1** Overview of defined entity classes and information flows
Figure 6.2 The task 'Process incoming applications and mutations'

The distinction which is made above between insurance applications, mutations on existing policies and damage claims is one which is also apparent in the
department's business processes. We start with the processes for new applications and mutations which are processed by the group for private accounts and the group for corporate accounts. Although the expertise required for corporate accounts is quite different from that which is required for private accounts, the processes for both are almost identical. The only difference is the fact that the Scanner department is involved in the process for corporate applications, whereas this is not the case for the private ones. In figure 6.2 the way in which insurance applications and mutations on existing insurance policies are dealt with after they are received by the mail room is displayed.

For the private accounts the process is depicted on the left hand side while the process for corporate accounts is depicted on the right hand side. In both cases, the incoming mail is sent to the fire insurance department where it is inspected by the head of the corporate accounts. Next, the administrative assistant sorts the mail and distributes it to the different groups. Applications are selected and sent to C.R.A. where the new client is registered. For private accounts, some applications are sent directly to the C.R.A. This is the case if a single client applies for different types of insurance at the same time. For corporate accounts, a similar exception is made when a corporate client applies for more than one type of insurance at the same time. These combined applications are sent directly to the Scanner department.

After the applications are registered by the C.R.A. they are returned to the department. Once they have been checked, either by the administrative assistant or by the Scanner department, the applications are assigned to one of the acceptors. For both groups some time can pass before applications are actually assigned due to the work in process inventory which is maintained by each group's head.

Mutations do not have to be registered by C.R.A.. Hence, after the administrative assistant has sorted out the mail the mutations can be assigned by the group's head to the acceptors. Again, some time can pass before mutations are actually assigned to an acceptor due to the fact that an inventory of work in process is also maintained for mutations by each group's head.

Figure 6.2 points out the similarities between the processes for private and corporate accounts. This similarity also holds for the task structures for 'Process insurance application' and 'Process mutation on existing insurance policy'. Therefore we will confine ourselves to discussing the task structures for the private accounts only. We refer to Van Eijden (1991) for an account on similar task structures devised for the corporate accounts group.

We first discuss the process for new insurance applications, followed by that for mutations on existing policies and for damage claims.

1. New insurance applications

The process for handling new insurance applications is depicted in figure 6.3. The process starts with an assessment of the application by the acceptor. After this
Figure 6.3 The task 'Process insurance application', actor: Acceptor
Case 2: An insurance firm

assessment, the acceptor can either be ready to take a decision, or he can require additional information. This can range from simple information which is acquired through the mail to specific information which can only be obtained by letting an expert visit the building for which an insurance policy is requested.

When the acceptor is ready to take a decision, the application can either be accepted or it can be refused. If it is accepted, the acceptor enters the appropriate data for the new insurance policy into the computer system. At night, the Operations department processes and prints these new policies. The next morning, the computer output is sent to the Premium department where the invoices for the new policies are checked. After this, the computer output is returned to the fire insurance department where the administrative assistant sorts it for each group.

Each acceptor assembles the correspondence for the policies they have processed. Hence, the computer output for a client is combined with the general information on the policy and policy conditions and the total package is put into an envelope which is ready to be sent away. After this, the client’s file and the outgoing mail need to be dispatched.

If an application is refused, the acceptor again enters the appropriate data into the computer system. This time, however, no output is generated by the Operations department. Instead, the letter explaining the reason(s) why the application is refused is made up and printed on a local printer in a nearby printing room. Four times a day these letters are retrieved from the printing room by the administrative assistant. Again the acceptor assembles the correspondence for each refused application. The final tasks in the process are again to dispatch the client’s file and the outgoing mail.

In figure 6.3 three tasks are double rimmed, implying that these are specified at a lower level of detail. The first of these is the task ‘Acquire additional information’.

![Diagram](image)

**Figure 6.4** The task 'Acquire additional information', actor: Accepter

132
When additional information is required, the acceptor issues a request for this information. Next, the administrative assistant stores the client’s information in a temporary file, where it awaits the additional information. As soon as the required information is received, the acceptor examines it and assesses the application once again. This process is depicted in figure 6.4.

The next doubled rimmed task in figure 6.3 is the task 'Dispatch client’s file'. For this task the acceptor first collects all the appropriate information in the client’s file. The file is then passed on to the administrative assistant who attaches the correct film jacket. This film jacket contains all the micro film elements which have been collected for that client up till now. The file with the film jacket is then sent to the C.R.A. department where the new information is filmed and added to the film jacket. Finally, the file is returned to the department and is stored by the administrative assistant in the departments own client files. The process is depicted in figure 6.5.

![Diagram](image)

**Figure 6.5** The task 'Dispatch client's file', actor: *Acceptor*

The final compound task is the one for the mail room. Its task structure is depicted in figure 6.6. First, the mail room collects all the outgoing mail from the different departments. This mail is then checked thoroughly in the mail room to see whether a correct address is written on the envelope. Finally, the mail is dispatched to the Dutch mail service each afternoon.
2. Mutations on existing insurance policies

Mutations on existing insurance policies can range from a simple change in the client's name, to a change in the policy conditions or the address, which requires a new assessment of the insurance risk and premium. The process for handling such mutations is depicted in figure 6.7.

First the acceptor inspects the mutation to see if it concerns a change in name or address or a change in the policy conditions. Only if a change in the name or address is reported, the C.R.A. has to be notified. Before this is done, however, the acceptor checks the central administration to see whether the change has already been reported by one of the other departments. This is possible for instance, if the same client also has different insurance policies at Goudse insurances. If the central administration is not yet changed, a mutation form is filled out and sent to C.R.A. where the change is made.

After the C.R.A. has been dealt with, the mutation is assessed. This can be very easy when the change reported does not effect the current insurance policy nor its conditions. Sometimes, however, a mutation can be just as complex as a new insurance application. This can for instance be the case if the client has moved to a different building. Hence, similar to the process for applications, the acceptor may require additional information (see figure 6.4 for the task structure). After the acceptor has received all the information he requires, a decision is taken. If the current policy remains unchanged, the information is entered into the computer system and processed. If the conditions do change, however, the acceptor informs the client or the intermediary and asks if the changes in the policy conditions are
acceptable. If the client accepts the changes the change is made in the computer system and processed. If the client refuses the changes the policy is cancelled. For both situations, a separate task structure is devised, representing the way in which the process continues.

![Diagram](image_url)

**Figure 6.7** The task 'Process mutation on existing insurance policy', actor: Acceptor

If the policy conditions remain unchanged, or if the changes are accepted by the client, a new policy is printed by the Operations department. The computer output is checked by the Premium department after which it is sent back. The administrative assistant checks the output and distributes it to the acceptors. Similar to the process for applications, the correspondence is assembled for the client. Finally, the client’s file is stored and the correspondence is sent out by the mail room. This process is depicted in figure 6.8.
If the client refuses possible changes to the policy conditions, the insurance policy is cancelled. In this case, a letter is written explaining the decision to the client. This letter is printed on a local printer. The administrative assistant retrieves
the letter from the printing room and forwards it to the acceptor. The correspondence is assembled and the client’s file and outgoing mail are dispatched. This process is depicted in figure 6.9.

3. Damage claims

Damage claims are processed by the claims group. The first steps in the process are similar to those for insurance applications and for mutations as the claims come in with the same pile of mail in which the others are received. Again, the head of the corporate accounts inspects the mail first, and the administrative assistant sorts the mail for each group. The damage claims are sent to the claims group, where the group’s head takes care of assigning the claims to the different claim attendants. Again, a delay can occur in this assignment of claims due to the fact that an inventory of work in process is maintained by the group’s head.

In addition to the damage claims which are received by mail, a significant portion of the claims is received by telephone. During the telephone calls, the specifics about the damage claim are written down and afterwards the claim is treated similar to its mailed in counterparts. These first tasks of the process are depicted in figure 6.10.

![Diagram](image)

Figure 6.10 The task 'Process incoming damage claims'
When a claim attendant starts to process a new damage claim, he first determines whether an expert should be sent to the building in question to check the damage. If such an expert assessment is required, a request for an expert report is issued and the client is informed. Further processing is delayed until the expert report is received back.

Figure 6.11 The task 'Process damage claim', actor: Claim attendant
If no expert report is required the attendant can assess the damage claim immediately. During this assessment, the need for further information may arise and the attendant will try to obtain this information. This is performed in a similar fashion as was the case for the acceptors (see figure 6.4). Also, it is possible that the attendant wants to have an expert assessment, although this was initially not required. When all the information is available, the attendant can take a decision.

If an expert report is received, the expert attendant examines it. Based on his findings the need for more information may evolve, or the attendant may be ready to take a decision. The file is handed back to the attendant who was initially working on the specific claim. This attendant first pays the expert. After this payment is confirmed by the Operations department the next day, the attendant determines the next action. Either the expert’s payment is wrong, implying that it needs to be repeated, or the final decision can be made.

If the claim is accepted, and the claim in question is large, a receipt is sent to the client which needs to be signed and sent back by the client before the actual payment can be performed.

The task structure for the damage claim handling process is depicted in figure 6.11.

Figure 6.12 The task ‘Refuse damage claim’, actor: Claim attendant
For the damage claims which are refused, the attendant makes up a letter informing the client about this decision. The administrative assistant retrieves the letter from the local printing room. Correct letters are given to the group's head who checks and signs it. Finally the letter is dispatched by the mail room. This process is depicted in figure 6.12.

For damage claims which are accepted, a payment to the client is issued. The next day, after the Operations department has processed this pay assignment, the group's head checks it and signs it for his consent. If errors were made the attendant has to issue a new payment which can again be checked the next day. Payments which are correct are sent to the accounting department which takes care of the actual fund transfers. Finally, the correspondence, informing the client about the decision, is processed by the mail room. The task structure for this process is depicted in figure 6.13.

Figure 6.13 The task 'Pay damage claim', actor: Claim attendant

This last task structure concludes the static model of the fire insurance department. Based on the process descriptions provided above, a number of conclusions can already be drawn about the current working procedures at the department:

1. All three groups within the department maintain inventories for work in process. New insurance applications, mutations and damage claims are first stored for some days before their processing begins. In most cases, more than one of these inventories is maintained, one for the items that have a high priority and one for items with a low priority. The delays that are introduced this way usually range from one to eight working days.
At the claims group separate inventories are maintained for claims which are difficult or large, and those which are trivial or small. The last group is stacked in one location. These simple claims are only processed by the claim attendants when these have some time left over during the day. Hence, although the processing of these claims is very easy and can be performed quite fast, a significant delay is introduced by using these priorities.

2. All new applications have to be registered by C.R.A.. Single fire insurance applications are, however, not delivered to C.R.A. directly, but are first sent to the fire insurance department. For corporate accounts new applications are also first sent to the Scanner department. Due to this indirect routing these applications can unnecessarily be delayed for up to two days.

3. When additional information is required, or when an expert assessment has to be made, a significant delay occurs. Although this can not be eliminated entirely, the lack of a recall mechanism may cause these delays to be far more than strictly necessary. As an indication: a delay of up to two months is not uncommon.

4. The batch oriented computer system which is used at the department causes a number of inefficiencies in the business processes. New or changed policies are not available until the next day. Hence, acceptors and claim attendants have to postpone their work for a day before it can be settled. If errors are made these have to be corrected which takes another day.

A second problem is the fact that the computer output is not sorted adequately. Hence, acceptors and claim attendants spend much time to find all the output which they require to assemble the correspondence for their clients.

5. The payment for an accepted damage claim takes a long time to be completed. Two main causes can be identified for this: (1) the expert is always payed before the client, and (2) the accounting department takes more than 10 days to approve and issue the final payment.

6. By introducing a special expert attendant in the claims group, the progress for each claim is dependent on a single person. If this person is ill or busy, expert reports are set aside for some time. If each claim attendant would process their own expert reports, this problem would not exist.

Apart from these findings, the interviews with the department’s personnel pointed out that delays are sometimes also caused by the priorities that are assigned to the different tasks within the processes. To illustrate this point a number of examples are given:
Case 2: An insurance firm

- New applications are only assigned to acceptors each morning. Applications which are received from the C.R.A. department around 11 a.m. are hence set aside until the next day.

- Although only half of the computer output has to be checked by the Premium department, all the output is delayed until the Premium department is finished. This is at the beginning of the afternoon. As a consequence of this delay, most of the correspondence is assembled and dispatched too late for the mail room to send it out the same day.

- At 3:30 p.m. the mail room collects the last batch of outgoing mail from the different departments. In most cases this mail can, however, not be completed by the mail room the same afternoon. It would be more effective if the last batch of mail was collected early enough for the mail room to finish it the same day.

To determine the importance of each of these findings, and to increase our understanding of the business processes, a dynamic model system of the department is devised.

The quantities which are required for the model system are determined partly by using expert assessments from the department’s employees and partly by using available data sets.

First the number of applications, mutations and damage claims that is received by the department is determined. As far as applications and mutations are concerned, the amount remains fairly stable throughout the year. Damage claims, however, have a peak at the beginning of the year when winter and spring storms often cause damage to buildings (see figure 6.14). We choose to exclude this seasonal fluctuation from our model system due to the fact that our simulation time will be far less than an entire year.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>applications for private accounts</td>
<td>15 - 25</td>
</tr>
<tr>
<td>mutations for private accounts</td>
<td>65 - 135</td>
</tr>
<tr>
<td>applications for corporate accounts</td>
<td>6 - 12</td>
</tr>
<tr>
<td>mutations for corporate accounts</td>
<td>40 - 60</td>
</tr>
<tr>
<td>damage claims</td>
<td>25 - 60</td>
</tr>
</tbody>
</table>

Table 6.1 Amount of incoming items at the department each day
The amount of items received by the department each day is listed in table 6.1. The amount of mutations is much greater than that of new applications. Furthermore the number of applications and mutations for private accounts exceed those for the corporate accounts. For the claims group, a distinction between private and corporate clients is not made.

In table 6.2 a number of percentages is listed for events within the process of handling new insurance applications. As is clear, most of the applications are accepted. For the corporate accounts, an experts advice is required for almost half of the applications.

<table>
<thead>
<tr>
<th></th>
<th>Private</th>
<th>Corporate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional information required</td>
<td>8 %</td>
<td>45 %</td>
</tr>
<tr>
<td>Application accepted</td>
<td>97 %</td>
<td>95 %</td>
</tr>
<tr>
<td>Application refused</td>
<td>3 %</td>
<td>5 %</td>
</tr>
<tr>
<td>Error in letter or computer output</td>
<td>2 %</td>
<td>2 %</td>
</tr>
</tbody>
</table>

Table 6.2 Probabilities for the insurance application handling process
Case 2: An insurance firm

For mutations, similar events can be distinguished. Two additional events pertain to the first tasks within the process where it is decided whether the C.R.A. should be informed about the mutation, and whether this is already done by another department. The probabilities are listed in table 6.3.

<table>
<thead>
<tr>
<th>Event</th>
<th>Private</th>
<th>Corporate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registration by C.R.A. required</td>
<td>20 %</td>
<td>20 %</td>
</tr>
<tr>
<td>Mutation already registered</td>
<td>50 %</td>
<td>50 %</td>
</tr>
<tr>
<td>Additional information required</td>
<td>12 %</td>
<td>16 %</td>
</tr>
<tr>
<td>Mutation accepted</td>
<td>99 %</td>
<td>98 %</td>
</tr>
<tr>
<td>Insurance policy cancelled</td>
<td>1 %</td>
<td>2 %</td>
</tr>
<tr>
<td>Error in letter or computer output</td>
<td>2 %</td>
<td>2 %</td>
</tr>
</tbody>
</table>

Table 6.3 Probabilities for the mutation handling process

Finally, the probabilities for events with respect to damage claims are listed in table 6.4. This time, a distinction is made between the need for an expert assessment, and the need for other additional information. The reason for this distinction is the fact that a low priority is assigned to those claims which do not need an expert assessment. These claims are set aside and have to wait until one of the attendants has some time left over to process them.

<table>
<thead>
<tr>
<th>Event</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert assessment required</td>
<td>25 %</td>
</tr>
<tr>
<td>Additional information required</td>
<td>30 %</td>
</tr>
<tr>
<td>Damage claim accepted</td>
<td>88 %</td>
</tr>
<tr>
<td>Damage claim refused</td>
<td>12 %</td>
</tr>
<tr>
<td>Error in letter or computer output</td>
<td>6 %</td>
</tr>
</tbody>
</table>

Table 6.4 Probabilities for the damage claim handling process

During the process of handling applications, mutations and damage claims, a number of delays can be identified. Within the private and corporate account groups new applications are discerned to be more important than mutations. Hence,
Section 6.2

Mutations are first stored in an inventory for work in process. After some days, these mutations are processed. Applications on the other hand are processed as soon as they arrive, or at least as soon as possible. At the private accounts group, new applications are assigned to the acceptors immediately. At the corporate accounts group these high priority items are stored in a so-called 'urgent file'. The items in this file are treated before the items in the regular inventory.

In the claims group the damage claims for which an expert assessment is required are treated with higher priority. The easier and small damage claims are only processed when claim attendants have time left over. Hence, for these items an inventory is maintained. The delay for each of the inventories is listed in table 6.5.

<table>
<thead>
<tr>
<th>Inventory Type</th>
<th>Delay (Working Days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Urgency file for corporate applications</td>
<td>1 - 5</td>
</tr>
<tr>
<td>Inventory of mutations for private accounts</td>
<td>1 - 8</td>
</tr>
<tr>
<td>Inventory of mutations for corporate accounts</td>
<td>3 - 20</td>
</tr>
<tr>
<td>Inventory for damage claims</td>
<td>8 - 24</td>
</tr>
</tbody>
</table>

**Table 6.5** Delays in inventories for work in process

Apart from these delays, a significant amount of time can pass when additional information or an expert assessment is required. In figure 6.15 an indication is given of the time it takes before the information or the expert report is received back. The

![Figure 6.15 Delay for acquiring an expert report](image-url)
data in figure 6.15 is gathered at the claims group, which has to use objective external experts. At the private and corporate account groups, inspection of a building is performed by the company’s own inspectors. The delay for these groups is smaller: between two and 10 weeks, where 80% of the reports is received within six weeks.

Using the data discussed above and additional data gathered during the different interviews, the delays and priorities for each task were specified. Also, the delay and probability of each transition was incorporated into the model. By applying the agenda mechanism to this completed model, a dynamic model system was obtained, as was discussed in the third chapter. For an account of the entire model system we refer to Van Eijden (1991) and Allemekinders (1992).

Using an animated version of the model system, a verification is performed with the department’s personnel. First by following a single item through the process, the delays, priorities and probabilities of the different events were checked. Next, the normal work load was generated by the model system and each person was able to check whether the daily work load and sequence of performance of tasks in the model system was in accordance with their actual working habits.

With the verification of the static model and dynamic model system, the specification is completed. Next the validation of the model system is performed to determine whether the model system’s output is correct. Finally the descriptive model system discussed in this section is used to determine potential improvements in the department’s business processes.

6.3 Validation

Similar to our first case study, a structural validation is performed using an animated simulation of the model system. For this purpose, treatments were devised for a series of experiments. In these experiments, the model system’s behavior was inspected while a number of changes had been made to it. In one of the experiments, some of the actors were deleted from the model system. It was determined whether the processes in which these actors are involved were blocked like one would expect. Also, the model system’s behavior was inspected for a larger and smaller work load.

For the replicative validation, a set of sample data is collected at the department. This sample data is compared with the model system’s output in order to determine whether this output is correct. First, however, a treatment for the replicative validation is devised similar to the structural validation discussed above.
Section 6.3

Validation

To devise a treatment for the replicative validation, we begin with the model system's start-up time. In figure 6.16 some traced variables are projected during simulation time.

![Graphs showing simulation time in days for private and corporate accounts groups.](image)

(a) private accounts group  
(b) corporate accounts group

<table>
<thead>
<tr>
<th>Group</th>
<th>Traced Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>level of occupation acceptors, life span complex applications, inventory (x 0.2)</td>
</tr>
<tr>
<td>(b)</td>
<td>life span complex applications, life span complex mutations, inventory (x 0.2)</td>
</tr>
<tr>
<td>(c)</td>
<td>life span small claims, life span complex claims, inventory (x 0.2)</td>
</tr>
</tbody>
</table>

(c) claims group

**Figure 6.16** Projection of traced variables to determine start-up time

In the model system's output a distinction is made between applications and mutations which require additional information, and those which do not. The first groups are referred to as 'complex applications' and 'complex mutations'. The last groups are referred to as 'smooth applications' and 'smooth mutations' respectively. This distinction is made due to the fact that the difference in time spans for smooth and complex items is extremely large. An average of both would not be very useful.
to determine the effect of changes made to the business processes during the solution finding phase. For damage claims the same distinction is made, where the terms 'complex claim' and 'small claim' are used for the two distinct types.

From figure 6.16 the start-up time can be determined. For the private accounts group, a start up time of 11 weeks (77 days) is sufficient. For the corporate accounts group, the start up time needs to be twice this amount, i.e. 22 weeks (154 days). Finally, the claims group requires the longest start-up time of 26 weeks (182 days). Due to the fact that the different groups operate in a very independent fashion, we chose to use three distinct model systems for the three groups. For the experiments with each of these model systems, separate treatments can hence be devised.

Next, the run length needs to be determined. Actually, a single working day can be considered as the longest cycle in the model system. The start-up time makes sure that the work load for each actor is sufficient and corresponds to the actual situation. This would imply that the run length could be as short as three days. However, the number of items that would finish during such a small simulation run is too low to provide meaningful statistics. For example, during a simulation run of three days, an average of 6 applications for corporate accounts and 20 damage claims would be completed. Numbers like these would not yield useful mean and variance values for the time span of items. This is even more so due to the distinction that is made between complex and simple items, implying that the small number of observations is also subdivided into these categories. Hence, a run length of several weeks, rather than days, is selected for the model systems in order to ensure useful output.

As output of the model system, the time spans for the different items is used. In addition to this, data is also collected for the level of occupation for the acceptors in both the private and corporate accounts group. These data can later be used to assess the changes in their work load when alternative business processes are evaluated.

Using the same procedure which is used in the previous case study (see pages 110-111), the required number of replications is determined. Due to the long start-up times each replication will take quite some time to be performed, implying that the number of replications should be minimized. For this reason we choose to use less restrictive values for $\alpha$ and $\delta$. This time we use a significance level $\alpha = 0.10$ and we calculate the width of the 90% confidence interval as $\frac{2}{N_0}$ of the difference between the maximum and minimum values found in the $N_0$ replications. Finally, we select $N_0 = 5$.

The results of the five replications for the model system describing the private accounts group are listed in table 6.6. The results point out that 10 replications will suffice for this model system.
Section 6.3 Validation

<table>
<thead>
<tr>
<th></th>
<th>obs</th>
<th>min</th>
<th>max</th>
<th>δ</th>
<th>stand. dev.</th>
<th>replications required</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean life span smooth applications</td>
<td>5</td>
<td>5.37</td>
<td>6.09</td>
<td>0.48</td>
<td>0.350</td>
<td>10</td>
</tr>
<tr>
<td>mean life span complex applications</td>
<td>5</td>
<td>30.2</td>
<td>40.2</td>
<td>6.7</td>
<td>4.14</td>
<td>7</td>
</tr>
<tr>
<td>mean life span smooth mutations</td>
<td>5</td>
<td>9.63</td>
<td>10.1</td>
<td>0.29</td>
<td>0.190</td>
<td>8</td>
</tr>
<tr>
<td>mean life span complex mutations</td>
<td>5</td>
<td>35.7</td>
<td>39.4</td>
<td>2.4</td>
<td>1.70</td>
<td>9</td>
</tr>
<tr>
<td>level of occupation acceptors</td>
<td>5</td>
<td>84.5</td>
<td>92.1</td>
<td>5.1</td>
<td>2.77</td>
<td>6</td>
</tr>
</tbody>
</table>

*α* = 0.10  \( t_{0.05,4} = 2.132 \)  
Max: 10

Table 6.6 Calculation of the required replications for the private accounts dept.

The results of the five initial replications for the corporate accounts group are listed in table 6.7. This time, 8 replications are sufficient to obtain model system output which meets our demands concerning significance level and accuracy.

<table>
<thead>
<tr>
<th></th>
<th>obs</th>
<th>min</th>
<th>max</th>
<th>δ</th>
<th>stand. dev.</th>
<th>replications required</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean life span smooth applications</td>
<td>5</td>
<td>8.30</td>
<td>8.84</td>
<td>0.36</td>
<td>0.238</td>
<td>8</td>
</tr>
<tr>
<td>mean life span complex applications</td>
<td>5</td>
<td>49.6</td>
<td>55.2</td>
<td>3.7</td>
<td>2.48</td>
<td>8</td>
</tr>
<tr>
<td>mean life span smooth mutations</td>
<td>5</td>
<td>14.2</td>
<td>15.2</td>
<td>0.62</td>
<td>0.349</td>
<td>6</td>
</tr>
<tr>
<td>mean life span complex mutations</td>
<td>5</td>
<td>54.5</td>
<td>59.9</td>
<td>3.6</td>
<td>2.20</td>
<td>7</td>
</tr>
<tr>
<td>level of occupation acceptors</td>
<td>5</td>
<td>76.7</td>
<td>80.0</td>
<td>2.2</td>
<td>1.26</td>
<td>6</td>
</tr>
</tbody>
</table>

*α* = 0.10  \( t_{0.05,4} = 2.132 \)  
Max: 8

Table 6.7 Calculation of the required replications for the corporate accounts dept.

Finally, the results for the damage claim group’s model system are listed in table 6.8. For complex claims, a distinction is made between claims which are received by mail, and claims which are received by telephone. The reason for this distinction is the fact that different priorities are assigned to both. Claims which are received by telephone are processed faster than those which are received by mail.
Table 6.8 Calculation of the required replications for the claims department

Based on the results of the first five replications, the required total number of replications for the claims group is determined to be 7.

Following the initial 5 replications for each of the three model systems, extra replications were performed to reach the required number. For each of the output variables the sample mean, standard deviation and 90% confidence level are listed in table 6.9.

Table 6.9 Results derived from the descriptive model systems
These results clearly show that for some of the items that are processed by the department, fairly long time spans exist. This is demonstrated in further detail by the frequency distributions depicted in figure 6.17. Lengthy time spans are specifically apparent for complex items which take up to four months before a final decision is taken and put into effect. Smooth items, however, also take quite some time before they are finished. This is certainly true when taking into account that the actual work, which is the assessment task, takes only 5 to 15 minutes for these items.

![Histograms showing time spans for smooth and complex mutations.](image)

**Figure 6.17** Time spans for mutations at the private accounts group

Before conclusions can be drawn from the derived output, however, the replicative validation still has to be performed. As stated earlier, sample data is collected for this purpose. This sample data which is collected at the department has to be compared with the model system's output. Due to time constraints, sample data could only be gathered at the private accounts group for a six week period. Hence, for this group the validation can only be performed for the smooth items due to the fact that the complex items take much longer than six weeks to be completed. For the complex items the model system's output is discussed with the department's employees to establish face validity. For the claims group, a rather complete set of data was gathered facilitating the validation of both smooth and complex claims. Unfortunately, no sample data is available from the corporate account's group.

For the validation of the model system's output, the mean and standard deviation of the sample data is compared graphically with the results of the five initial replications performed with the model systems. In figure 6.18 the results are shown for the private account's group. The dot represents the mean life span. The lines above and under the dot represent the values for the mean plus or minus the standard deviation.
Figure 6.18 Validation of model systems output for the private accounts group

Apart from the fact that the deviation in the sample data for applications is slightly larger than the deviation in the model system's output, the output matches the sample data quite accurately. This is also true for the output derived from the model system of the claims group, as illustrated in figure 6.19. This time, the deviation in the model system's output for complex claims is larger than the deviation in the sample data. Also, one of the replications offers time spans for smooth claims which are slightly lower than the sample data shows. Apart from this, however, the similarities are quite clear.

Figure 6.19 Validation of model systems output for the claims group
Now that the model system’s output is shown to correspond with reality, the model system can be used to diagnose causes for the problems within the department and to devise and evaluate changes which can resolve these problems.

6.4 Diagnosis, solution finding and evaluation

The validated descriptive model systems of the three groups within the department are used to evaluate the effect of changes made to the working procedures used within these groups. The effort should result in a set of change alternatives which can be used to achieve the objective of an improved service level.

To devise alternative changes, the model systems were presented to a group of participants in the organization. These participants included those who are involved daily in performing the business processes analyzed, as well as the department’s management. Following the presentation, the participants discussed different options which they expect to be effective. At first, the options that were proposed concerned slight changes to the current working procedures and task priorities. As the management is interested in more drastic changes which may be feasible in the future, however, these changes were also covered during the discussion sessions.

The dynamic model systems were used during the sessions to convey an image of the proposed situations using a graphical animation. Afterwards, the proposed changes were each described in a separate prescriptive model system and an experiment was performed to obtain numerical output from these model systems. During a subsequent session, these results of the change alternatives were presented to the participants. By discussing these results, the participants were able to increase their understanding of effective courses of action. Triggered by these new insights, new alternatives were proposed, where the results of these changes were discussed again in a following session.

During the two to three sessions which were held for each of the groups, a number of change alternatives was proposed. Below, each of these alternatives is discussed, and the results are presented. Due to the similarities in the current situation at the private and corporate accounts group, the same alternatives are proposed for both. We will first discuss both of these groups. Next, the claims group is discussed separately.

6.4.1 The private and corporate accounts groups

During the three sessions that were held for the private and corporate accounts groups, a number of change alternatives was proposed by the participants. Below, each of these alternatives is discussed separately.
1. Reduction of work in process inventories

The first change that was proposed by the participants is a reduction of the inventories for work in process. New applications should be processed immediately when they arrive. Also, acceptors should start the processing of mutations more quickly than is currently the case. To achieve this the department will have to work overtime to reduce the current inventory. From that time on, however, the capacity is sufficient to sustain the lower level of inventory.

The participants proposed to eliminate the so-called 'urgent file' at the corporate accounts group which is currently used for applications, and to reduce the delays for mutations to a maximum of 5 working days. The objective of this measure is to reduce the time span for smooth applications and mutations. In table 6.10 the results of the experiment with the model system in which these changes are described, are listed.

<table>
<thead>
<tr>
<th>Private accounts group</th>
<th>current situation (n = 10)</th>
<th>less inventory (n = 10)</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
</tr>
<tr>
<td>life span smooth appl.</td>
<td>5.79</td>
<td>0.248</td>
<td>5.9</td>
</tr>
<tr>
<td>life span complex appl.</td>
<td>35.1</td>
<td>3.03</td>
<td>35.8</td>
</tr>
<tr>
<td>life span smooth mut.</td>
<td>10.0</td>
<td>0.131</td>
<td>6.7</td>
</tr>
<tr>
<td>life span complex mut.</td>
<td>37.6</td>
<td>1.23</td>
<td>34.6</td>
</tr>
<tr>
<td>occupation acceptants</td>
<td>88.3</td>
<td>1.63</td>
<td>83.1</td>
</tr>
</tbody>
</table>

\[t_{0.025, 18} = 2.101 \quad t_{0.05, 18} = 1.734\]

n.s.: not significant

<table>
<thead>
<tr>
<th>Corporate accounts group</th>
<th>current situation (n = 8)</th>
<th>less inventory (n = 8)</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
</tr>
<tr>
<td>life span smooth appl.</td>
<td>8.61</td>
<td>0.168</td>
<td>5.2</td>
</tr>
<tr>
<td>life span complex appl.</td>
<td>52.5</td>
<td>1.86</td>
<td>48.6</td>
</tr>
<tr>
<td>life span smooth mut.</td>
<td>14.6</td>
<td>0.236</td>
<td>7.7</td>
</tr>
<tr>
<td>life span complex mut.</td>
<td>57.1</td>
<td>1.58</td>
<td>51.0</td>
</tr>
<tr>
<td>occupation acceptants</td>
<td>78.3</td>
<td>0.914</td>
<td>77.2</td>
</tr>
</tbody>
</table>

\[t_{0.025, 14} = 2.120 \quad t_{0.05, 14} = 1.746\]

n.s.: not significant

Table 6.10 Results of a reduction of inventories

154
For the applications at the corporate accounts group, an improvement of more than 3 days is established. At the private accounts group no improvement is achieved for applications due to the fact that this group already processes applications as soon as they arrive. The time span for mutations at the private and corporate accounts group is reduced by 3 and 7 days respectively.

It is interesting to see that besides the improvements in time spans, a reduction of the occupation level of the acceptors is established. Hence, although it will take an extra effort to reduce the inventories, the acceptors will eventually be less busy while processing the same amount of items.

In addition to the reduction of inventories at both groups, a separate experiment is performed where the inventories are eliminated completely. Applications as well as mutations are processed immediately when they arrive.

<table>
<thead>
<tr>
<th>Private accounts group</th>
<th>current situation (n = 10)</th>
<th>no inventory (n = 10)</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
</tr>
<tr>
<td>life span smooth applc.</td>
<td>5.79</td>
<td>0.248</td>
<td>5.8</td>
</tr>
<tr>
<td>life span complex app.</td>
<td>35.1</td>
<td>3.03</td>
<td>36.7</td>
</tr>
<tr>
<td>life span smooth mut.</td>
<td>10.0</td>
<td>0.131</td>
<td>4.6</td>
</tr>
<tr>
<td>life span complex mut.</td>
<td>37.6</td>
<td>1.23</td>
<td>33.0</td>
</tr>
<tr>
<td>occupation acceptants</td>
<td>88.3</td>
<td>1.63</td>
<td>84.2</td>
</tr>
</tbody>
</table>

\[ t_{0.025,18} = 2.101 \quad t_{0.05,18} = 1.734 \quad \text{n.s.: not significant} \]

<table>
<thead>
<tr>
<th>Corporate accounts group</th>
<th>current situation (n = 8)</th>
<th>no inventory (n = 8)</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
</tr>
<tr>
<td>life span smooth applc.</td>
<td>8.61</td>
<td>0.168</td>
<td>5.0</td>
</tr>
<tr>
<td>life span complex app.</td>
<td>52.5</td>
<td>1.86</td>
<td>48.0</td>
</tr>
<tr>
<td>life span smooth mut.</td>
<td>14.6</td>
<td>0.236</td>
<td>4.2</td>
</tr>
<tr>
<td>life span complex mut.</td>
<td>57.1</td>
<td>1.58</td>
<td>47.6</td>
</tr>
<tr>
<td>occupation acceptants</td>
<td>78.3</td>
<td>0.914</td>
<td>77.6</td>
</tr>
</tbody>
</table>

\[ t_{0.025,14} = 2.120 \quad t_{0.05,14} = 1.746 \quad \text{n.s.: not significant} \]

**Table 6.11** Results of eliminating inventories
As is clear from table 6.11 which lists the results of the second experiment, an additional improvement of 2 to 4 days is achieved in the life span of mutations by this change.

2. Reduction of the delay for expert’s advice or additional information

By reducing the different inventories for work in process, a significant reduction of the time spans for smooth items is established. Especially for smooth mutations at the corporate accounts group, where a reduction from 15 to 4 days is established, the improvements look quite promising. However, the improvements for complex items are less obvious. The long time spans that are required for these complex items stem from the long delays that are associated with obtaining an advice from an inspector who visits the building which needs to be insured, or the delays associated with obtaining additional information. Therefore, the second change proposed by the participants is to reduce this delay.

In figure 6.20 the delay for obtaining additional information, either from the client or from an inspector, is given for the current situation, as well as the improved situation. The proposed reduction from an average of 4.2 weeks to 3.6 weeks, as is depicted in this figure, can be established by sending reminders more often to those who need to provide the desired information.

![Figure 6.20 Delay for obtaining additional information](image)

Apart from a reduction of the delay itself, the participants also predict that the number of insurance applications and mutations requiring additional information can be reduced when the Assurance Data Net is used more often. To test the effect of such a reduction the participants proposed to reduce this number by 50%.
Section 6.4 Diagnosis, solution finding and evaluation

To determine the effect of the proposed change, an experiment with the prescriptive model system describing this change is performed. The results of this experiment are listed in table 6.12.

<table>
<thead>
<tr>
<th>Private accounts group</th>
<th>current situation (n = 10)</th>
<th></th>
<th>faster advice (n = 10)</th>
<th></th>
<th>significance level</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
<td>s.d.</td>
<td>t t1</td>
<td>0.05</td>
<td>0.10</td>
</tr>
<tr>
<td>life span smooth applic</td>
<td>5.79</td>
<td>0.248</td>
<td>5.9</td>
<td>0.312</td>
<td>0.476</td>
<td>n.s.</td>
<td>n.s.</td>
</tr>
<tr>
<td>life span complex app.</td>
<td>35.1</td>
<td>3.03</td>
<td>31.9</td>
<td>5.83</td>
<td>1.54</td>
<td>n.s.</td>
<td>n.s.</td>
</tr>
<tr>
<td>life span smooth mut.</td>
<td>10.0</td>
<td>0.131</td>
<td>9.9</td>
<td>0.248</td>
<td>1.69</td>
<td>n.s.</td>
<td>n.s.</td>
</tr>
<tr>
<td>life span complex mut.</td>
<td>37.6</td>
<td>1.23</td>
<td>32.1</td>
<td>2.25</td>
<td>6.73</td>
<td></td>
<td></td>
</tr>
<tr>
<td>occupation acceptants</td>
<td>88.3</td>
<td>1.63</td>
<td>77.3</td>
<td>1.51</td>
<td>15.7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ t_{0.025, 18} = 2.101 \quad t_{0.05, 18} = 1.734 \quad \text{n.s.: not significant} \]

<table>
<thead>
<tr>
<th>Corporate accounts group</th>
<th>current situation (n = 8)</th>
<th></th>
<th>faster advice (n = 8)</th>
<th></th>
<th>significance level</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
<td>s.d.</td>
<td>t t1</td>
<td>0.05</td>
<td>0.10</td>
</tr>
<tr>
<td>life span smooth applic</td>
<td>8.61</td>
<td>0.168</td>
<td>8.8</td>
<td>0.403</td>
<td>0.907</td>
<td>n.s.</td>
<td>n.s.</td>
</tr>
<tr>
<td>life span complex app.</td>
<td>52.5</td>
<td>1.86</td>
<td>49.3</td>
<td>1.83</td>
<td>3.47</td>
<td></td>
<td></td>
</tr>
<tr>
<td>life span smooth mut.</td>
<td>14.6</td>
<td>0.236</td>
<td>14.6</td>
<td>0.253</td>
<td>0</td>
<td>n.s.</td>
<td>n.s.</td>
</tr>
<tr>
<td>life span complex mut.</td>
<td>57.1</td>
<td>1.58</td>
<td>55.6</td>
<td>2.09</td>
<td>1.62</td>
<td>n.s.</td>
<td>n.s.</td>
</tr>
<tr>
<td>occupation acceptants</td>
<td>78.3</td>
<td>0.914</td>
<td>73.4</td>
<td>0.921</td>
<td>10.7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ t_{0.025, 14} = 2.120 \quad t_{0.05, 14} = 1.746 \quad \text{n.s.: not significant} \]

Table 6.12 Results of faster expert's advice and additional information

As far as the time spans for complex items are concerned, significant improvements are only established for complex mutations at the private accounts group and complex applications at the corporate accounts group. The average time span for complex items is reduced by 2 to 5 days. In addition to this, the number of complex items is also reduced, as is illustrated for the private accounts group in figure 6.21.
Figure 6.21 Comparison of time spans for complex mutations for private accounts

It is interesting to see that the occupation level for acceptors is reduced significantly. Evidently, items for which additional information or an expert’s advice is required take much of the acceptors’ time and an effort to reduce the number of these complex items, for instance by more direct (electronic) communication with intermediaries, can be quite worthwhile.

3. Different routing and coordination of the departments involved

During the specification of the descriptive model a number of inconveniently chosen priorities and routings were identified. The third alternative proposed by the participants changes priorities and routings in order to improve the coordination of the different departments involved. The following changes were proposed:

1. All applications, combined and single, are directly sent to the C.R.A. department instead of the indirect routing used for single applications at the moment.

2. When C.R.A. delivers the new applications at the fire insurance department at the end of the morning, these applications are immediately processed instead of setting them aside until the next morning.

3. The computer output which does not have to go by the Premium department is sent directly to the fire insurance department. Here, the correspondence is assembled for the clients with high priority in order to send it out as soon as possible.
4. An extra mail collection is introduced in order to provide more time for the mail room to process the outgoing mail. This way, the chance of getting the correspondence out the same afternoon is increased.

All of these measures can be implemented by making relatively simple changes to the working procedures at the departments involved. To determine the effect of these changes, an experiment is performed which yields the results as listed in Table 6.13.

<table>
<thead>
<tr>
<th>Private accounts group</th>
<th>current situation (n = 10)</th>
<th>better coordin. (n = 10)</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
</tr>
<tr>
<td>life span smooth applic</td>
<td>5.79</td>
<td>0.248</td>
<td>2.6</td>
</tr>
<tr>
<td>life span complex app.</td>
<td>35.1</td>
<td>3.03</td>
<td>31.4</td>
</tr>
<tr>
<td>life span smooth mut.</td>
<td>10.0</td>
<td>0.131</td>
<td>9.4</td>
</tr>
<tr>
<td>life span complex mut.</td>
<td>37.6</td>
<td>1.23</td>
<td>36.7</td>
</tr>
<tr>
<td>occupation acceptants</td>
<td>88.3</td>
<td>1.63</td>
<td>85.2</td>
</tr>
</tbody>
</table>

\[ t_{0.025,18} = 2.101 \quad t_{0.05,18} = 1.734 \quad \text{n.s.: not significant} \]

<table>
<thead>
<tr>
<th>Corporate accounts group</th>
<th>current situation (n = 8)</th>
<th>better coordin. (n = 8)</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
</tr>
<tr>
<td>life span smooth applic</td>
<td>8.61</td>
<td>0.168</td>
<td>6.7</td>
</tr>
<tr>
<td>life span complex app.</td>
<td>52.5</td>
<td>1.86</td>
<td>49.6</td>
</tr>
<tr>
<td>life span smooth mut.</td>
<td>14.6</td>
<td>0.236</td>
<td>14.8</td>
</tr>
<tr>
<td>life span complex mut.</td>
<td>57.1</td>
<td>1.58</td>
<td>57.2</td>
</tr>
<tr>
<td>occupation acceptants</td>
<td>78.3</td>
<td>0.914</td>
<td>75.6</td>
</tr>
</tbody>
</table>

\[ t_{0.025,14} = 2.120 \quad t_{0.05,14} = 1.746 \quad \text{n.s.: not significant} \]

Table 6.13 Results of alternative routing and priorities
The most prominent effect is the reduction of the time spans for smooth applications. The alternative routing, combined with the other measures, facilitates a reduction of 2 to 3 days on the average. This reduction is also achieved for complex applications, although the improvement is relatively small in comparison with the total time span for these items.

During the discussion of the results for this alternative, the value of the animated simulation was demonstrated. One of the acceptors noticed that the change affected the working habits for acceptors. Normally, they are used to processing applications in the morning and mutations in the afternoon. With the animated simulation it became clear that with the new routing and priorities in place, these working habits are reversed, where applications are mainly processed in the afternoon and mutations are processed in the morning.

4. A completely new way of working facilitated by new computer support

For the long range, the department's management proposed to analyze which improvements could be achieved when constraints regarding the current computer support would be neglected. For this analysis it is presumed that an on-line computer system is used instead of the current batch processing system.

With an on-line computer system the smooth applications and mutations can be finished completely within the same day. The output is printed at a local printing device and the correspondence is assembled immediately afterwards. For an optimal use of this possibility, the work in process inventories are eliminated similar to the second change alternative. Also, the changes in routings and priorities which were proposed as the third alternative are adopted here.

In the ideal situation, the new on-line system could be used in conjunction with the ADN network discussed earlier. If this were the case, additional information could be acquired more quickly and redundant data entry would be eliminated.

A prescriptive model system is created in which the new way of working is described. The results of the experiment performed with this model system are listed in table 6.14.

As is made clear by the results, significant improvements are possible when an on-line computer system would be made available. Smooth applications take only one day on the average to be processed while mutations take only slightly longer due to the lower priority which is assigned to them. Complex items still take an average of 4 weeks for private accounts and six weeks for corporate accounts, although this is still an improvement of 1 to 2 weeks. Also, the number of complex items is reduced due to the fact that more information can immediately be obtained using the ADN network for communication.
Table 6.14 Results of a new way of working with new computer support

Triggered by the results of this experiment, a final change alternative is analyzed. This time, the current computer support is maintained but a way of working is implemented which matches the ideal situation as closely as possible. Due to the fact that the batch system is maintained, the correspondence can not be assembled the same day, but needs to be postponed until the next morning. Apart from this however, the new way of working can be implemented without any excessive changes. The only change that is required for the computer system is a change in the order in which the output is printed, eliminating the need to sort it out at the department. The results of the experiment are listed in table 6.15.
### Case 2: An insurance firm

#### Table 6.15 Results of a new way of working with current computer support

As is clear from the results, the improvements are still substantial and only differ slightly from those obtained in the ideal situation. This is also made clear by the frequency distributions of the time spans for the different items at the corporate accounts group as depicted in figure 6.22.
Figure 6.22 Comparison of time spans for corporate accounts

The findings discussed above can be used to determine in which way the department can improve its service levels for clients and intermediaries. Some of the changes can be implemented with little or no effort required. Other alternatives can be used to direct thinking about the department for the longer term. For each of the alternatives, the effects are made clear, not only for the timeliness of the services provided by the department, but also for the working environments of the people involved.
6.4.2 The claims group

The model system of the claims group was used during five subsequent sessions where the group's staff and its management discussed change alternatives. The alternatives that came about during these sessions are discussed below.

1. Excluding the special expert attendant

One of the conclusions that was drawn during the description phase (see point 6 on page 141) was the fact that the use of a special expert attendant whom is responsible for interpreting the expert reports creates a rather vulnerable situation for the claims group. If the expert attendant is absent or busy, the expert reports are set aside for some time, implying that the processing of these damage claims is needless delayed.

The first proposition brought forward by the department's staff was to exclude this special expert attendant from the model system. Instead, each claim attendant would be responsible for the expert reports which relate to the claims they are working on. The expert attendant is then replaced by a general claim attendant who can perform all tasks of the process rather than just review the expert reports.

With this proposed change, a reduction of the time spans for complex items is expected. In addition to this, the fact that an extra person is available for the other tasks can affect the time spans for the simple claims as well. The results of the experiment performed to evaluate this proposal are listed in table 6.16.

<table>
<thead>
<tr>
<th></th>
<th>current situation (n = 7)</th>
<th>no expert attend. (n = 7)</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
</tr>
<tr>
<td>life span small claims</td>
<td>16.2</td>
<td>2.07</td>
<td>12.2</td>
</tr>
<tr>
<td>complex claims mail</td>
<td>42.8</td>
<td>1.20</td>
<td>38.4</td>
</tr>
<tr>
<td>complex claims phone</td>
<td>38.3</td>
<td>0.513</td>
<td>36.5</td>
</tr>
</tbody>
</table>

\[ t_{0.025,12} = 2.179 \quad t_{0.05,12} = 1.782 \quad \text{n.s.: not significant} \]

Table 6.16 Results of excluding the expert attendant

The results point out that a significant improvement in the time spans of all items is obtained. The mean time spans for all items is reduced by about 4 days. Given the fact that realization of the proposed change is quite easy, there was a broad consensus amongst the participants regarding its implementation.
2. Process small claims more often

The most striking conclusion at the end of the description phase was the fact that, although small item take a very small amount of time to be processed, they were treated with a very low priority. When the new claims arrive at the department, the group's head selects the complex claims and distributes these amongst the claim attendants. The remaining claims, which are all small, are placed on a stack in a filing-cabinet. The claim attendants only process these small claims if they have some time left during the day.

As a consequence of the low priority for small claims, the time span for these claims is rather long. On the average, small claims take 16 days to be completed. In a worst case, such a claim can take up to 25 days to be completed. The second change proposed by the participants was to let each of the five claim attendants process small claims during one afternoon each week. Hence, each afternoon one person is available to process these claims. Table 6.17 shows the results of the experiment which is performed to evaluate this proposal.

<table>
<thead>
<tr>
<th>current situation (n = 7)</th>
<th>more small claims (n = 7)</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
</tr>
<tr>
<td>life span small claims</td>
<td>16.2</td>
<td>2.07</td>
</tr>
<tr>
<td>complex claims mail</td>
<td>42.8</td>
<td>1.20</td>
</tr>
<tr>
<td>complex claims phone</td>
<td>38.3</td>
<td>0.513</td>
</tr>
</tbody>
</table>

\[ t_{0.025, 12} = 2.179 \quad t_{0.05, 12} = 1.782 \quad \text{n.s.: not significant} \]

Table 6.17 Results of letting attendants process small claims one afternoon

With this second proposed change, the time span of small claims is reduced by more than 50%. It is interesting to see that the improvement which is obtained by this change does not result in longer time spans for the complex items. Hence, the long time spans for complex items are not caused by a lack of capacity of the claim attendants, but rather by the long delays that occur during the process. In a subsequent experiment, each claim attendant processes small claims during two afternoons each week. In table 6.18 the results of this experiment are shown.
Case 2: An insurance firm

<table>
<thead>
<tr>
<th></th>
<th>current situation (n = 7)</th>
<th>extra small claims (n = 7)</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
</tr>
<tr>
<td>life span small claims</td>
<td>16.2</td>
<td>2.07</td>
<td>3.29</td>
</tr>
<tr>
<td>complex claims mail</td>
<td>42.8</td>
<td>1.20</td>
<td>38.7</td>
</tr>
<tr>
<td>complex claims phone</td>
<td>38.3</td>
<td>0.513</td>
<td>36.3</td>
</tr>
</tbody>
</table>

\[ t_{0.025, 12} = 2.179 \quad t_{0.05, 12} = 1.782 \quad \text{n.s.: not significant} \]

**Table 6.18** Results of letting attendants process small claims two afternoons

This time the delays are eliminated almost completely. On the average, a small claim is finished in less than 4 days. To make clear exactly what the impact of the proposed changes on the time span for small claims is, the frequency distribution of

![Bar chart showing time spans for small claims](chart.png)

**Figure 6.23** Comparison of time spans for small claims
this time span is shown for the current situation as well as for the two proposed changes in figure 6.23. The improvements are striking, especially for the second change option where each of the claim attendants processes small claims during two afternoons each week.

3. Differentiate between laborious and effortless claims

Another proposal which aims for improvements in the time span of small claims is to differentiate between laborious and effortless claims. When the group’s head sorts out the mail, the small claims for which almost no effort is required can be selected along with the complex claims, and distributed. The remaining small claims are placed on the stack similar to the way in which they are currently treated.

The idea behind this alternative is that the selected effortless claims can be processed very easily in between other tasks. These claims will then be finished faster, without disturbing the working habits currently used.

During one week the group’s head has counted the small claims which are considered laborious, and those which are effortless. As it turned out, some 75% of all the small claims require a very small effort to be dealt with. Only 25% of the claims were considered to be laborious. To stay on the safe side, it was presumed that 50% of the claims was effortless. The results of the experiment are listed in table 6.19.

<table>
<thead>
<tr>
<th>current situation</th>
<th>effortless claims</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n = 7) mean</td>
<td>s.d.</td>
<td>mean</td>
</tr>
<tr>
<td>life span small claims</td>
<td>16.2</td>
<td>2.07</td>
</tr>
<tr>
<td>complex claims mail</td>
<td>42.8</td>
<td>1.20</td>
</tr>
<tr>
<td>complex claims phone</td>
<td>38.3</td>
<td>0.513</td>
</tr>
</tbody>
</table>

| t_{0.025, 12} = 2.179 | t_{0.05, 12} = 1.782 | n.s.: not significant |

Table 6.19 Results of differentiating between laborious and effortless claims

Again, significant improvements are obtained for all items. On the average, the time span for small claims is reduced by almost 50%. Half of the claims is even finished within 5 days, as is made clear in figure 6.24.
Case 2: An insurance firm

Figure 6.24 Reduction of the time span by selecting effortless claims

4. Simulation of a storm

In addition to the experiments discussed above, a number of experiments is performed to determine the effect of changes which are not related to the business processes themselves. The first of these experiments regards the effect of a significant increase of the work load on the number of claims kept in the inventory. Such a raise in work load can occur if a winter or spring storm hits the country, damaging a great number of houses and other buildings.

In the experiment we want to determine how well the department can cope with such an increase in claims. The storm is simulated in the current situation, as well as for the situation where small claims receive more attention, or when the effortless claims are selected first. The storm is simulated by doubling the number of incoming claims during a period of two weeks starting on the 210th day in simulation time. In figure 6.25 the results for the three situations are shown.

In the current situation, the level of inventory is quite instable, and it takes a long time before the department is recovered from an increase in the work load. In both change options, the recovery is faster. In the situation where claim attendants process small claims during one afternoon each week the inventory does not rise very much when the increase in work load occurs.
The best option turns out to be the situation where laborious and effortless claims are distinguished. In this situation the inventory not only recovers faster, but also stabilizes at a much lower level.

5. A new way of working with improved computer support

Similar to our last experiment at the private and corporate account groups, the last experiment explores the potential improvements that would be possible if the constraints imposed by the current computer system would not exist. For the claims group these constraints are very limited. Actually, the only problem is the fact that letters are not printed at the department itself, but rather by the Operations department during the system run at night. In the final experiment the participants presumed that local printing devices would be available. All the computer output could then be collected immediately and the claim could be dealt with completely without delays. In addition to this change it is presumed that an expert's advice can be obtained more quickly due to the use of a reminding system and more frequent use of the ADN network. The results of the experiment are listed in table 6.20.
Case 2: An insurance firm

<table>
<thead>
<tr>
<th></th>
<th>current situation (n = 7)</th>
<th>improved support (n = 7)</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
</tr>
<tr>
<td>life span small claims</td>
<td>16.2</td>
<td>2.07</td>
<td>3.76</td>
</tr>
<tr>
<td>complex claims mail</td>
<td>42.8</td>
<td>1.20</td>
<td>25.8</td>
</tr>
<tr>
<td>complex claims phone</td>
<td>38.3</td>
<td>0.513</td>
<td>23.6</td>
</tr>
</tbody>
</table>

\[ t_{0.025,12} = 2.179 \quad t_{0.05,12} = 1.782 \] n.s.: not significant

**Table 6.20** Results of improved computer support

It is clear from these results that the improved computer support facilitates significantly lower time spans for all items. The average time span of small items is reduced to less than a quarter of its current value, while the time span of complex items decreases with 40%.

The investments which are required to implement the proposed change are minimal. Recently a new internal fibreglass communication network has been installed at Goudse insurances. This network can be used to redirect computer output to the insurance departments implying that the only costs are those for the local printing devices.

### 6.5 Summary and conclusions

In this chapter, an application of the dynamic modelling approach in the private sector has been discussed. The analysis concerned a department for fire insurance within a Dutch insurance firm. Departing from a general question concerning the way in which the service level for clients and intermediaries can be improved, a detailed description of the current organization was made. The way in which different people and department cooperate within the main business processes of the department was emphasized in the analysis.

Using the descriptive models devised for the three groups which are distinguished within the department, discussion sessions were organized for both the department's personnel as well as its management. During these sessions, change alternatives were proposed by the participants. Next, the model was used to determine the effect of the proposed changes, where the numerical model output was used to determine the effect on the timeliness of the services, and the animation of the simulation was used to demonstrate which changes may occur for the working habits of the department's personnel.

170
The conclusions for this chapter are, like was the case in the previous chapter, divided into those which pertain to our dynamic modelling approach, and those which pertain to the organization we analyzed.

1. The dynamic modelling approach

The analysis described in this chapter demonstrates the applicability of the dynamic modelling approach in the private sector. The main difference with the analysis presented in the previous chapter, is the fact that opportunities for improvements rather than problems were emphasized. Despite this difference, however, the approach of describing the current organization in detail before possible changes are considered proved to be effective just the same. Again, the descriptive model provided the essential yard stick which facilitates a basis for evaluating the effects of change.

The use of a dynamic model system in making clear what a new situation will look like has proven to be very effective. Although the numerical model system output pertaining to the timeliness of the services is emphasized in the previous section, the model systems also facilitated a projection of what the new work place for all those involved will be like. The fact that the impact of the proposed changes on everyone’s working environment is made clear beforehand precludes a sceptic attitude which might otherwise have obstructed the change process.

2. The fire insurance department

With the changes occurring in the Dutch and European market place, Goudse insurances is looking for ways to increase the quality and timeliness of the services they provide to their clients and intermediaries.

In this chapter, the analysis of one insurance department has yielded a number of alternative changes which can be implemented to realize the desired improvements. A close look at the current working procedures, making clear where and when different departments need to cooperate, has indicated a number of inconveniently chosen routings, time schedules and priorities. These insights can be used to obtain significant improvements with relative ease. Here, the model system describing the changed situation not only makes clear what the effects are for the services, but also makes clear to all those involved what changes may occur in their work environment and working habits.

Apart from the relatively small changes, extrapolated from the status quo, the knowledge about the department’s business processes is also used to determine the effects of more extensive changes such as a switch from batch to on-line processing. The fact that the effects of such a measure are known beforehand, makes it easier to plan ahead for the organization on the longer term. Investments required for the more drastic changes can be compared with their potential effects and a well reasoned decision can be made. For the fire insurance department it has become
clear, however, that many of the improvements that can be obtained by employing a new computer system are also possible with the current computer support. It is our conviction that changes in the working procedures and improved coordination between departments provide far more potential for improving the service levels than improved technical support does. Also, if new technologies are to be employed, changes in the business processes are certainly required to make the investments worthwhile.
7

Case 3: A municipal department

7.1 Introduction

Our third case study addresses a municipal department which is responsible for the construction and buildings administration within the municipality of Rheden. Hence, the case study again concerns the public sector, although this time the analysis was prompted by a desire to improve the level of service provided to citizens rather than problems associated with the department's internal productivity, as was the situation in our first case study. As such, this third case study is complementary to the first two and establishes a more complete coverage of our domain of interest.

Rheden is a medium-sized municipality located in a rural, wooded area in the east of the Netherlands. In 1985 the organization structure of the municipality was reorganized in order to meet new demands imposed on them by the Dutch central government. The main consideration which prompted this reorganization was the desire to improve the quality of services provided to citizens by the municipal departments. Changes were also needed because of the budget cuts throughout the 1980s.

As the city council aims at a quality improvement of the services provided to citizens, it participates in a national project on civic service centers. Such service centers will provide the same services that are offered by municipalities at the moment, but in addition will provide those services which are currently organized at the provincial or national level. The objective is to establish a single desk where people can come to get answers to any question pertaining to public facilities in the Netherlands.

The first initiative within the municipality is to integrate service desks which are distributed amongst the different departments. In a later stage, the services provided at the provincial or national level can be incorporated in these service desks.
In the new, simplified organization structure, four main divisions are distinguished:

1. **Managing division**: This division encompasses the departments which are involved with finance, personnel management and information facilities.

2. **Town development**: This division is concerned with town and country planning, building regulations and the maintenance of municipal land.

3. **Public constructions and facilities**: This division takes care of public buildings, the local road network and municipal cleansing.

4. **Social services**: This last division administers a variety of subsidies and unemployment benefits provided by the Dutch government.

In addition to these four divisions the municipal organization encompasses a local police force and fire brigade.

Our case study focuses on the second division, *Town development*. Specifically, the *Department of Building Regulation* which resides within this division is our subject of analysis. This department's main responsibility is to ensure that all construction that takes place on municipal land proceeds conform to building regulations. Control is effectuated mainly through building permits and inspection of both actual and prospective construction sites.

Within this department a specific problem has arisen as a result of a change in national legislation. The particular problem has to do with building permits which are issued by the department. Up till now, a period of two months was available for the department to process an application and for the city council to make a final decision about the permit. If this period is not sufficient, the decision about the building permit can be postponed. In practice, such postponements are used regularly, implying that a substantial number of applications takes longer than two months to be processed. The main reason for these postponements is the fact that the department does not reject applications which do not meet the standards that apply to building permit applications. Instead, the department helps its clients to resolve possible problems in the application. This effort often takes longer than the two months time available.

With a new housing act, the Dutch government is going to impose a new standard time table for treating building permit applications. The main objective is to eliminate differences which currently exist between different municipalities. According to the new legislation a decision about a building permit has to be taken within three months. If a decision is not taken within this period of time, the building permit is considered to be granted. Hence, if the department exceeds the time limits imposed, the consequences are quite severe.
To determine the main causes for the excessive processing time for building permit applications, a model of the department and its current procedures is made. In subsequent sections, the specification and validation of this model is discussed, followed by the diagnosis of and solution finding for the problems perceived.

### 7.2 Specification

With the preliminary problem statement provided in the introduction, the emphasis of the analysis effort is to create a descriptive model which can be used to determine the main reasons why building permit applications take a long time to be processed. Hence, a demarcation of the modelling effort is that only the processing of applications is taken into consideration. Based on the findings in our previous case studies, the analysis will focus on the coordination and cooperation that is required between the different departments and offices involved. The exact content of each task, and the expertise required for it, is of less interest to us.

Similar to the previous case studies, we start with the specification of entity classes which are used to describe the department and its environment. Following these definitions, the task structures for processing building permit applications are given.

We start with the clients who apply for a building permit. To do so, a building plan and the appropriate technical information is sent to the department for building regulation. In some cases the applicant first sends a sketch of the building he plans to make. The use of such a sketch plan minimizes the costs and effort made by the client when it turns out that the proposed building is in conflict with the town development plans.

During the judgement process, objections against the building plans may arise based on technical grounds or town development plans. It is also possible that the proposed building does not meet the property qualifications established within the municipality. A district architect determines whether or not a building meets these qualification and if it does not, the applicant can either withdraw his application or revise the building plan. Based on the above the following two entity classes can be specified:

```plaintext
class Applicant ← simple actor
  attributes
  personal data
  skills
  send a building sketch
  apply for a building permit
  revise a building plan
  withdraw an application
end class

class Building permit ← simple item
  attributes
  application
  location
  building plan
  technical information
end class
```
Next, we specify the entity classes for the department’s own personnel. The eleven people that work at the department can be subdivided in the following way: one head of department, three construction engineers, three building supervisors, two administrative assistants and two legal assistants. Processing building permit applications takes virtually all of the engineers’ time, and it accounts for some 15 to 20% of the working time for the others.

Each of the functionaries distinguished within the department is responsible for specific tasks within the process:

- The head of department manages the applications handling process. To maintain a general view of the applications currently processed by the department, the head of department inspects each application that is received. Furthermore, the head of department is responsible for discussing the building plans with the district’s architect. Finally, he checks each final advice which is given to the city council.

- The construction engineers are mainly responsible for processing the applications. They check the building plans against the town development plans and building regulations. During the judgement process, the construction engineer also collects all the advice that is required from other departments. Finally, the construction engineers are responsible for maintaining contact with clients.

- The building supervisors visit existing buildings and prospective construction sites to collect information that is required to make an adequate judgement. During such visits, all the questions raised by the construction engineer are answered. Also, the people who live near the building location can be visited in order to assess any possible objections they may have against the building plans.

- The administrative assistants take care of the registration of applications, and perform the paperwork during the process. Specifically, the preparation of an advice to the city council and the dispatching of granted permits is performed by these assistants.

- Finally, the legal assistants are mainly occupied with the people who appeal against the city council’s decision. As this appeal procedure does not have to be finished within the three month period set by the government, we will not elaborate this process in the model.
The entity class definitions for the different functionaries are specified below:

```plaintext
class Head of department ← simple actor
    attributes
        name
    skills
        inspect application
        discuss building plan with architect
        check final advice
end class

class Construction engineer ← simple actor
    attributes
        name
        district responsibility
    skills
        treat building permit application
end class

class Building supervisor ← simple actor
    attributes
        name
    skills
        visit building location
end class

class Administrative ass. ← simple actor
    attributes
        name
    skills
        register application form
        prepare advice for city council
        dispatch building permit
end class

class Legal assistant ← simple actor
    attributes
        name
    skills
        process appeal
end class
```
Besides the department’s own personnel, third parties are also involved in the application handling process. To start with, a district architect reviews all the building plans for new buildings and for plans which alter the external appearance of an existing building. This architect evaluates these building plans to determine if they are conforable to existing property qualifications. If the architect disapproves the plans, he discusses this with a building committee. This committee can either agree with the architect, or overrule his decision and approve the building plans.

class District architect ← simple actor
attributes
name
skills
assessment building plan
end class

class Building comm. ← simple actor
attributes
skills
judge building plan
end class

During the judgement process, a construction engineer can collect advice from other municipal departments. Departments which are often addressed for this purpose are the department for town and country planning, the environmental department and the fire brigade. The definition for these departments is kept very simple as we are only interested in their contribution to the building permit application handling process.

class Other department ← simple actor
attributes
name
response time
skills
offer advice
end class

Finally, the head of division can be asked to inspect the building plans if some of the rules need to be bent in order to accept the application. Furthermore, the city council has to make the final decision on granting a building permit. Again, very simple definitions are used:

class Head of division ← simple actor
attributes
name
skills
inspect building plans
end class

class City council ← simple actor
attributes
skills
make final decision
end class
The entity classes defined above, and the information flows that can be identified between these classes, are summarized graphically in figure 7.1. The defined entity classes provide a demarcation of the scope of the analysis. In addition to the different people active within the department, a number of external advisors and other involved parties are identified and their relationship with the department of building regulation is made clear.

Figure 7.1 Overview of defined entity classes and information flows

To enhance our understanding of the information flows, specifically between the different people active within the department, a task structure for processing building permit applications is devised.
There are two ways by which an applicant of a building permit can contact the department, as depicted in figure 7.2. The first is to send in the appropriate forms and drawings by mail. These will be received in the mail room and forwarded to the department of building regulation. Alternatively, a client can come to the department’s service desk and hand over the application personally. In this case, one of the construction engineers will scan through the application quickly and the client can ask questions about the procedure. Also, if information is missing this can be resolved immediately or the application can be returned to the client who can supplement it.

![Diagram of the task 'Receive and process building permit applications']

**Figure 7.2** The task 'Receive and process building permit applications'

Applications which are received by the department are first inspected by the head of department who scans through the application. By doing so, the head of department maintains a general overview of the kind of applications that are currently in consideration. The head of department assigns a date of arrival to the application. This date is used to determine the total amount of time it took to process the application. Hence, in the new situation, the three months time period will start on that specific date. We note that this date is only assigned if a complete application is available. In the case of a sketch plan such a date is not assigned until all the appropriate forms are made available.

Next the application is registered by an administrative assistant. Some specific details, such as the specific town development plan that applies to the application, can only be registered after the construction engineer has reviewed the application. Within the municipality, some 135 of these development plans exist, each pertaining to a specific area of the municipal land. After a construction engineer has determined the appropriate development plan, the registration is supplemented, and the application is returned to the construction engineer who can then prepare a decision.

The construction engineer investigates the application in order to make a decision on granting the building permit. During this investigation different departments can be addressed for advice, which will be explained further on. As a result of the construction engineer’s investigation, four outcomes are possible:
1. The applicant will withdraw his application. This will occur when serious objections are made during the investigation and the client does not want to revise his building plans.

2. The application is refused. Again objections are found against the building plans but this time the client does not want to withdraw his application. In some cases the client will appeal against the decision in which case the application is transferred to one of the legal assistants.

Figure 7.3 The task 'Process building permit application', actor: Construction eng.
3. The decision about an application is postponed. In this case, the building plans are compatible with the current town development plans for the building location, but conflict with new development plans for the specific area. The decision is postponed until the new development plan is in force at which time the application can be refused. Obviously, the client is informed about this procedure and is enabled to revise the building plan in order to meet the requirements imposed by the new development plans.

4. Finally, the application can be accepted implying that the building permit will be granted. In this case an administrative assistant prepares the paperwork and the city council confirms or overrules the decision during its meeting.

The task structure for processing applications is given in figure 7.3. It is clear that the actual work is performed during the task 'Make a decision about the building permit'. Hence, we zoom in on this task to describe it in more detail in figure 7.4.

The construction engineer starts to check the building plan against the building regulations. The technical information and drawings are checked to determine whether all building regulations are met by the design.

Next, the construction engineer verifies the building plan with the town development plans for the building location. In addition to the plan currently in force, possible new plans are also checked because the building plan should obviously be compatible with such new development plans. If the building plan is in conflict with the development plans, it is determined whether possible exemptions from the rules can be made for the client.

If problems occur, either with the development plans or building regulations, these problems can be discussed with the applicant. After this, the construction engineer can either be ready to make a decision, or can require additional information or advice.

If advice is required, the construction engineer will collect this. The outcome can again be discussed with the applicant. With the acquired advice the construction engineer will reconsider the building plan and proceed similar to the first time. Several iterations of this verify-advice-verify cycle are possible if the outcome of one advice implies that further information is needed.

At some point in time the construction engineer will be able to make a decision. The application can either be refused or accepted right away, or it can be discussed with the head of division. The head of division is only consulted for specific cases, where the application can only be accepted by 'bending the rules'.

If the application is refused, this is discussed with the applicant. The client can either withdraw the application or revise his building plan. If the second option is chosen the revised building plan goes through the same verification procedure that its predecessor has already followed.
Figure 7.4 The task 'Make decision about building permit', actor: Construction eng.
If the application is accepted, a publication of the building plan may be required. Such a publication enables other citizens to appeal against the granted permit. If no appeal is made, the building permit can be prepared and the construction engineer is finished with his work.

If an appeal is made against the decision to grant a permit, one of the legal assistants proceeds with the application. As stated earlier, this appeal procedure will not be discussed in further detail.

We zoom in on the task 'Obtain advice from other departments'. As depicted in figure 7.5 three types of advice are distinguished.

![Diagram](image)

**Figure 7.5** The task 'Obtain advice from other departments', actor: Construction eng

First, a visit to the building location may be required. In this case one of the building supervisors is asked to do so, and to fill out a form with questions devised by the construction engineer.

The second type of advice concerns the property qualifications which are established by the district architect. This architect determines whether the building
which is proposed fits into the landscape at the building location. If the architect wants to refuse the building permit, a so-called building committee investigates the building plan. This committee can either agree with the architect and refuse the plan, or it can overrule the architect’s decision. The architect is only available once every two weeks which is represented by the JoinAndSplit symbol in figure 7.5.

Finally other departments can be addressed to obtain further information. Some examples are the department for town and country planning, the environmental department, the department of traffic and the fire brigade.

When an advice is required for the first time, the building plans are always first sent to the building supervisor. When the building plans have been revised and an advice is required for the second time, the supervisor is no longer required and the architect is addressed directly.

During a visit to the building location, the building supervisor has to answer all the questions raised by the construction engineer. Therefore, the building plans are first examined closely. After the answers have been determined, the building supervisor checks to see whether or not exemptions are required to be able to accept the application. If such exemptions are required, the people in neighbouring buildings have to give their consent. Hence, the supervisor visits these people if exemptions are required. Finally, the appropriate form is filled out and returned to the construction engineer. This procedure is represented in figure 7.6.

**Figure 7.6** The task 'Visit building location', actor: Building supervisor
The only task that has not yet been discussed concerns the preparation of a permit decision for the city council which is a subtask identified in figure 7.3. A task structure for this task is given in figure 7.7.

![Task structure diagram](image)

**Figure 7.7** The task 'Prepare decision for city council', actor: Administrative ass.

After the construction engineer has decided to accept the application, and hence grant the building permit, the city council has to confirm this decision, or overrule it. To prepare for the city council meeting, the administrative assistant has to perform some paperwork.

First, it is determined whether the council meeting will take place within the time limits of two months after the application was received. If this is not the case, a postponement has to be arranged by the construction engineer.

Next, the final advice is prepared for the city council. The head of department checks this final advice and if mistakes are found, returns it to the administrative assistant.

Finally, the advice is dispatched to the city council which takes the final decision after which the building permit can be issued.
With this last task structure the static model of the department of building regulation is complete. Similar to our previous case studies, these static descriptions are reviewed by the different departments involved in the application handling process.

Based on the static description, some preliminary causes for a long time span of the process can be identified:

1. The registration of the application and the determination of the appropriate town development plans is performed with a low priority. Hence, from the start of the process, excessive delays can be observed if other building plans occupy the construction engineers.

2. If discrepancies are found between town development plans and a client’s building plan, extensive efforts are made to determine how a permit can still be granted. Such an investigation is very time consuming as it involves cooperation with the town development planning department.

3. A great amount of time is lost if advice is required from other departments or from the district architect. A period of three weeks is not unusual for such advice. Also, once an advice is obtained, the construction engineer can take quite a while before he resumes his investigation, again due to the priorities assigned to the different types of work.

4. Finally, if communication is required with the client, or if the client wants to revise his building plan, much time is usually lost.

To increase our understanding of these and possible other causes for the excessive time span of the process a dynamic representation of the department is devised.

To determine the input data required for our model system a sample of building permit decisions is taken from the department’s records. This sample is used to determine the frequency of arrival for applications, and the probabilities of the different alternative paths in the task structures.

For the sample from the historical files, one week is chosen at random from each month in 1990, and all the building permits that were issued during those weeks are selected. The sample contains a total of 148 applications. The collected sample does not contain all the required information due to the fact that the department does not maintain files from applications which are not granted. Hence, additional information is collected from interviews with the department’s personnel.

First, the arrival of applications is considered. In figure 7.8 and figure 7.9 the distribution of applications over the days of the week, and over the year is projected.

187
Figure 7.8 Distribution of applications through the week

Although the applications are evidently not evenly spread across the year, we used the average of 8.3% per month in our model system as it is our intention to analyze causes for delays using an average work load.

Figure 7.9 Distribution of applications through the year

On the average, the number of applications per year is about 620 which amounts to about 12 each week. In the model system, the distribution of applications through the week is modelled in accordance with figure 7.8.
At the department, a distinction is made between so-called A-applications and B-applications. The first concern building permits which can only be granted if exemptions are made to the rules concerning building regulations or town development plans. For these applications an advice from other departments is more often required. As a consequence, the time span for these A-applications is significantly larger than that of the B-applications.

In figure 7.10 the percentage of A and B applications is depicted. This figure also specifies the percentage of forms received through the mail and at the department’s service desk.

![Diagram](image)

**Figure 7.10** Additional information on applications received by the department

Next, the probability of the different outcomes of the process is determined. First, the probabilities of the different outcomes for the construction engineers is determined. The possible outcomes of the decision can either be to accept the application or to refuse it. However, the client can also withdraw his application or revise his building plan. In figure 7.11 the probabilities for these outcomes is presented.

![Diagram](image)

**Figure 7.11** Probabilities of decision outcomes for construction engineers
If the client decides to revise his building plan, the new plan is evaluated and a decision is taken again. Hence, eventually the 26% of the applications for which a revised plan is provided will be subdivided amongst the other three categories. With the assumption that the probabilities of accepting, refusing or withdrawing a revised building plan are equal to those of a building plan which is evaluated for the first time the probabilities for the different outcomes is determined and presented in figure 7.12.

![Pie chart showing probabilities](image)

**Figure 7.12** Probabilities of eventual decision outcomes

The next category of information pertains to advices which are required from other departments. In the historical files the different advices which were required for each application are written down. In table 7.1 the probability for such advices from the different departments is specified.

<table>
<thead>
<tr>
<th>Department</th>
<th>Observations</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fire brigade</td>
<td>2</td>
<td>1.3 %</td>
</tr>
<tr>
<td>Maintenance of municipal land</td>
<td>4</td>
<td>2.6 %</td>
</tr>
<tr>
<td>Public constructions and facilities</td>
<td>5</td>
<td>3.2 %</td>
</tr>
<tr>
<td>Environmental department</td>
<td>8</td>
<td>5.1 %</td>
</tr>
<tr>
<td>Town and country development</td>
<td>15</td>
<td>9.6 %</td>
</tr>
<tr>
<td>Building supervisor</td>
<td>128</td>
<td>82 %</td>
</tr>
<tr>
<td>Building committee</td>
<td>147</td>
<td>94 %</td>
</tr>
</tbody>
</table>

**Table 7.1** Probabilities for requiring advice
From table 7.1 it can be derived that advice is required from other departments, not including the building committee, with a probability of 21.8%. This does not imply, however, that 21.8% of the applications requires an advice because for some applications more than one advice is required. From the sample data it is determined that 18% of the applications require one or more advices from other departments.

As is indicated in table 7.1 the building committee is asked for advice for almost all of the building plans. Only if the application concerns a change which is inside an existing building, and hence does not affect its external appearance, can an advice from the building committee be omitted.

As depicted in figure 7.5 the head of department first discusses a building plan with the district architect. If this architect approves the plan, it does not have to be judged by the entire committee. Hence, the committee is only consulted if the architect refuses the plan. In figure 7.13 this is illustrated. Also, the probabilities for acceptance and rejection for both the architect and the committee are depicted in this figure.

![Pie charts showing probabilities of advice acceptance and refusal.]

**Figure 7.13** Judgement by the district architect and the building committee

The time it takes before an advice is received back is quite long. In the sample data the date of issuing and receiving the advice was recorded. Based on these dates it is determined that 70% of the advices is returned within four weeks time. The remaining advices can even take up to 16 weeks to be received. In figure 7.14 the time span for advices which were found in the sample data are depicted.
Case 3: A municipal department

Figure 7.14 Time span of advice from other departments

The advice from the building committee is *not* incorporated in figure 7.14. The time span for these advices is determined in the following way:

- The district architect visits the department on every other tuesday. Hence, if advice is required, an application can be delayed between 1 and 13 days.

- If the architect approves the building plan, the application is delayed another 6 days due to the fact that these plans are not received back until the next monday.

- If the architect disapproved the building plan, the entire building committee has to judge the plan which takes between $2\frac{1}{2}$ and 4 weeks.

The final delay associated with requested advice is the delay for a visit to the building location by one of the supervisors. From the sample data it is determined that the supervisor takes an average of 7 days to visit the building location and return the question form to the construction engineer.

Finally, the probabilities for some other events in the model system are determined. These are used to ascertain that the correct amount of applications is passed along the different alternative paths through the task structure. The values determined for this purpose are summarized in table 7.2.
Section 7.2 Specification

| Sketch plan received before actual application | 27 | 17%  
| Consult head of division | 13 | 8%  
| Appeal against decision of the city council | 1 | 0.6%  
| Publication for objections from other citizens | 14 | 9%  

Table 7.2 Probabilities of events

If the head of division is consulted, a delay of 1 to 6 calendar days can occur due to the fact that this is only performed once a week. If a publication of a building plan is required, a delay of 28 to 32 days occurs caused by the paper work involved and the fact that the information bulletin is only published once a month.

The data that is collected from the sample of historical building permit applications and from interviews with the department’s personnel, are used to construct the model system of the department (for the entire model system, see Trommel (1991)). With this model system several experiments are performed for verification purposes. An animated simulation showing the different actors with their current task is used to ascertain that the sequence of tasks and the delays and probabilities are specified correctly.

Basically two types of experiments are used. The first generates a single application and the sequence of tasks for this application is followed on the animated display. In the second type of experiment the correct amount of applications is generated and the animated display is used to check whether the work load and the priorities for each actor are realistic.

Having specified and verified both the static model and the dynamic model system of the department of building regulation, the next step is to validate the model system and use it to analyze the current situation and to generate and evaluate solutions for the problems perceived.

7.3 Validation

For the structural validation a number of experiments is performed to check whether the model system’s behavior under certain conditions is as one would expect it to be. For this purpose a treatment is devised and a series of experiments is performed with the model system. Specifically, various values for the number of phone calls
and visits of clients at the service desk have been used to determine the effect on the work load of the construction engineers and the time span of the process as whole. The work load that was predicted by the model system in these experiments were as we had expected them to be. Experiments were not performed for the work load of the other people and departments involved in the process due to the fact that these people and departments are only partly occupied with tasks related to the building permit application handling process.

Following the structural validation, a treatment for the replicative validation is devised. Similar to the previous studies, the start-up time is determined first, followed by the run length, the model system's output and the number of replications.

![Graph showing the mean time spans for A and B applications](image)

**Figure 7.15** Projection of mean time spans to determine start-up time

The start-up time for the model system is determined by projecting the average time span for A and B-applications against the simulation time. As is shown in figure 7.15 the start-up for the model system is 250 days. This is quite long, but it is explained by the great amount of time that passes when advice is to be obtained.
To calculate the run length, the longest cycle in the model has to be determined. For this purpose, the time span of asking advice from the building committee is used due to the fact that this is the most common and one of the most time consuming advices contained in the model. If the district architect approves the building plan, the advice only takes between 7 and 14 days. However, if the plan has to be judged by the entire committee, this can take up to 28 days extra. In total, a time span of about 35 days is required for the building committee’s advice. For the run length we chose to use 100 days (± 3×35 days).

The most important model system’s output is the total time span of processing A and B-applications. Due to the fact that B-applications take less time to be processed, the output of data for A and B-application is separated. Due to the fact that rejections of building plans are rarely discerned, we are mainly interested in the time span of A and B-applications which are accepted. In addition to these outputs, the level of occupation for construction engineers is an output of the model system.

Using the same procedure as applied in the previous case studies, the number of replications can be determined using the values for start-up time, run length and model output specified above. To determine the number of replications, an initial experiment is performed of \( N_0 = 12 \) replications. We select a significance level \( \alpha = 0.05 \). For the width of the \((1-\alpha)\) confidence interval of each output variable we choose to use \( \frac{1}{3} \) of the difference between the minimum and maximum value found in the \( N_0 \) replications. The results are listed in table 7.3.

<table>
<thead>
<tr>
<th></th>
<th>obs</th>
<th>min</th>
<th>max</th>
<th>( \delta )</th>
<th>stand. dev.</th>
<th>replications required</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean life span A-applications</td>
<td>12</td>
<td>73.2</td>
<td>97.3</td>
<td>8.0</td>
<td>6.63</td>
<td>14</td>
</tr>
<tr>
<td>mean life span B-applications</td>
<td>12</td>
<td>67.2</td>
<td>74.7</td>
<td>2.5</td>
<td>2.72</td>
<td>23</td>
</tr>
<tr>
<td>number of A-appl. accepted</td>
<td>12</td>
<td>47</td>
<td>79</td>
<td>10.7</td>
<td>8.11</td>
<td>12</td>
</tr>
<tr>
<td>number of B-appl. accepted</td>
<td>12</td>
<td>72</td>
<td>105</td>
<td>11</td>
<td>9.05</td>
<td>14</td>
</tr>
<tr>
<td>level of occupation constr. eng.</td>
<td>12</td>
<td>86.5</td>
<td>96.7</td>
<td>3.4</td>
<td>3.52</td>
<td>21</td>
</tr>
</tbody>
</table>

\( \alpha = 0.05 \quad t_{0.025,11} = 2.201 \)  \quad \text{Max: 23}

**Table 7.3** Calculation of the number of replications required

An additional experiment with 23-12=11 replications is performed with the model system. For each of the output variables, the sample mean, standard deviation and 95% confidence interval are listed in table 7.4.
Table 7.4 Results derived from the descriptive model system

The results clearly illustrate the long time spans for the applications handling process. On the average the difference between the time spans for A and B-applications is more than two weeks. However, even the so-called easy B-applications take an average of ten weeks to finish.

Figure 7.16 Graphical comparison of sample data and model system output
Before any conclusions are drawn from the model system's output, its correctness has to be ascertained (Bots, Dur and Trommel 1992). Hence, the replicative validation has to be performed. To get an idea of how well the model output fits the sample data the frequency distribution of both is projected in a histogram in figure 7.16.

In figure 7.16 the distinction between A and B-applications is not made. Hence, the frequency distributions pertain to both types. From the figure we can see that the match between the sample data and model system's output seems to be quite close. The most apparent differences are the fact that the model system generates less applications that finish during the first 20 days, and more applications that finish after 200 days. Apart from these differences the frequency distributions are almost identical.

As a more formal comparison between the model system’s output and the sample data, a Pierson's test or $\chi^2$-test is used (Bhattacharyya and Johnson 1977; Ringuest 1987).

For this test a number of categories is defined for the time spans that are discerned. The chosen categories for A and B-applications are listed in table 7.5. Different categories for A and B-applications have been chosen due to the fact that more B-applications are finished in a short time.

<table>
<thead>
<tr>
<th>A-applications</th>
<th>B-applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>class 1</td>
<td>0 - 40 days</td>
</tr>
<tr>
<td>class 2</td>
<td>41 - 70 days</td>
</tr>
<tr>
<td>class 3</td>
<td>71 - 110 days</td>
</tr>
<tr>
<td>class 4</td>
<td>111 - 150 days</td>
</tr>
<tr>
<td>class 5</td>
<td>151 - 240 days</td>
</tr>
</tbody>
</table>

**Table 7.5** Chosen categories for the $\chi^2$-test

For the $\chi^2$-test the number of applications in each category for the sample data and for the model system's output is compared. For this purpose the number of observations for each category in the sample data is adjusted to the number of observations derived from the model system.
Next the measure of fit is determined in the following way:

- Let \( s_i \) be the number of observations in category \( i \) for the sample data (\( i = 1..5 \))
- Let \( m_i \) be the number of observations in category \( i \) for the model output
- Let \( N_s \) be the total number of observations for the sample data
- Let \( N_m \) be the total number of observations for the model output

The adjusted frequencies for the sample data are determined as: 
\[
a_i = \frac{s_i \cdot N_m}{N_s}
\]

Now the difference between the two data sets is determined as: 
\[
\chi^2 = \sum_{i=1}^{5} \frac{(m_i - a_i)^2}{a_i}
\]

The hypothesis that the two sets of data are identical is accepted if the difference is less than or equal to the appropriate \( \chi^2 \) value. We use a confidence level of \( \alpha=0.05 \) and find that our critical value is: \( \chi^2_{0.05,4} = 9.49 \).

In table 7.6 the results are listed for A-applications. The difference found is 5.08 implying that our model system is valid for A-applications.

<table>
<thead>
<tr>
<th>class ( s_i ) ( (N_s = 64) )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_i )</td>
<td>11</td>
<td>19</td>
<td>15</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>( a_i )</td>
<td>22.3</td>
<td>38.6</td>
<td>30.5</td>
<td>22.3</td>
<td>16.3</td>
</tr>
<tr>
<td>( m_i ) ( (N_m = 130) )</td>
<td>19</td>
<td>36</td>
<td>39</td>
<td>25</td>
<td>11</td>
</tr>
</tbody>
</table>

*measure of difference: 5.08*

**Table 7.6** Results of the \( \chi^2 \)-test for A-applications

In table 7.7 the results are listed for B-applications. Again, the difference of 7.59 is less than the critical value of 9.49 implying that the model is valid for B-applications also.
Table 7.7 Results of the $\chi^2$-test for B-applications

Now that the model system's output is shown to correspond with reality, the model system can be used to analyze perceived problems and to find and evaluate alternative solutions for these problems.

7.4 Diagnosis, solution finding and evaluation

From the results of the descriptive model system a clear indication of the problems is obtained. The time it takes to process A-applications is nearly three months and even the B-applications which are considered easy take more on the average than the two months that is actually available. When, in the near future, the new housing act is in force, the department will not be able to finish all the applications within the three months (90 days) time limit. For A-application, 37 % of the decisions is taken after 90 days. For B-applications 25 % of the decisions is taken late. With the new housing act in force, the building permits for these applications would have to be granted without being 'tested'.

In the previous section, a number of causes for excessive time spans is mentioned. To determine the validity of these causes, experiments are performed with the model system. Below, each distinguished cause is discussed separately.

Cause 1: Slow start

The first cause mentioned is the fact that some days can pass before the processing of a new application starts. The reason for this is the fact that other tasks are assigned a higher priority. To determine the effect of this cause the priority of new applications is raised. Now, a construction engineer will first deal with these new applications and initiate the required advices, before his other tasks are performed.
Case 3: A municipal department

Chapter 7

With the change in priorities, an improvement in the time span for both A and B-applications is expected. The measure should not affect the number of applications which are accepted, nor should it effect the level of occupation for the construction engineers as the same work is still performed, be it in a different sequence. In table 7.8 the results of the experiment are listed.

<table>
<thead>
<tr>
<th></th>
<th>current situation (n = 23)</th>
<th>faster start (n = 23)</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
</tr>
<tr>
<td>life span A-appl.</td>
<td>87.8</td>
<td>5.76</td>
<td>83.9</td>
</tr>
<tr>
<td>life span B-appl.</td>
<td>70.1</td>
<td>3.44</td>
<td>67.4</td>
</tr>
<tr>
<td>number of A accepted</td>
<td>64.5</td>
<td>7.98</td>
<td>63.1</td>
</tr>
<tr>
<td>number of B accepted</td>
<td>88.4</td>
<td>8.92</td>
<td>84.9</td>
</tr>
<tr>
<td>occupation constr. eng.</td>
<td>91.6</td>
<td>3.45</td>
<td>91.7</td>
</tr>
</tbody>
</table>

\[ t_{0.025, 44} = 2.016 \quad t_{0.05, 44} = 1.680 \quad \text{n.s.: not significant} \]

**Table 7.8** Results of a faster start of the process

At the significance level \( \alpha = 0.05 \), only a significant decrease in the time span for A-applications is established. The reduction of 4 days, however, is hardly enough to solve the department’s problem.

**Cause 2: Extensive efforts to accept plans**

The second cause for the long time spans mentioned in the previous section is the fact that the department makes a great effort to accept building plans even when these contradict with town development plans. To do so, however, many other departments have to be consulted and, hence, construction engineers need to spend a great amount of time on these applications.

To determine the effect of this cause, a new policy is implemented in the model system. Now, if a building plan conflicts with the town development plans currently in force, or with those which will become in force in the near future, the building plan is refused. In the case of such a rejection the client is no longer able to revise the building plans. If the client wishes to change the plans a new application has to be filled out.

As a result of the new policy, an improvement in the time span for A-applications is expected. Due to the fact that these A-applications are refused more quickly, construction engineers will have more time available for the easier applications. Hence, the time span for B-applications can decrease also. The number
of A-applications that are accepted, however, will decrease. This is not true for the B-applications because these, by definition, meet the town development plans. Finally, the level of occupation can decrease due to the fact that less effort is required from the construction engineers. In table 7.9 the results of the experiment are listed.

<table>
<thead>
<tr>
<th></th>
<th>current situation (n = 23)</th>
<th>new policy (n = 23)</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
</tr>
<tr>
<td>life span A-applic.</td>
<td>87.8</td>
<td>5.76</td>
<td>76.8</td>
</tr>
<tr>
<td>life span B-applic.</td>
<td>70.1</td>
<td>3.44</td>
<td>66.2</td>
</tr>
<tr>
<td>number of A accepted</td>
<td>64.5</td>
<td>7.98</td>
<td>37.9</td>
</tr>
<tr>
<td>number of B accepted</td>
<td>88.4</td>
<td>8.92</td>
<td>88.4</td>
</tr>
<tr>
<td>occupation constr. eng.</td>
<td>91.6</td>
<td>3.45</td>
<td>86.2</td>
</tr>
</tbody>
</table>

\[ t_{0.025, 44} = 2.016 \quad t_{0.05, 44} = 1.680 \]

**n.s.: not significant**

**Table 7.9 Results of the new policy**

The effects that were predicted do indeed occur. This time, the time span for A-applications is improved by 11 days, which is quite significant. The reverse side of the medal, however, is the fact that a significant increase in the number of rejected applications is caused by the new policy. The high level of service which was provided by the department, is now replaced by a policy of rather blunt rejections. One of the side effects of such a measure is an increase in the number of appeals made by clients, which will affect the work load of the legal assistants.

**Cause 3: Long delays for obtaining advice**

The third cause for the department’s problems is the long delays associated with obtaining advice from other departments. Not only the time it takes before the advice is received is considered important, but also the delay at the department between receiving an advice, and proceeding with the application’s processing. Again, due to the priorities set by the construction engineers, some days can pass before a construction engineer resumes his work on an application.

In the model system, a number of changes is made to evaluate the effect of this cause. First, a higher priority is assigned to applications which are waiting for an advice. Hence, the work on such an application is resumed more quickly. Also the delays of advice from other departments is reduced from an average of more than
30 days, to an average of 6 days. Finally, the delay for obtaining an advice from the building committee is reduced. The district architect will visit the department once every week instead of once every two weeks. If a building plan is rejected by the architect, the building committee will provide its advice within an average of 11 days, instead of the current 23 day average.

With these measures a decrease in the time span of both A and B-applications should be provided. It should not affect the number of applications that are accepted. Also, the effect on the work load of construction engineers will be small. The results of this experiment are listed in table 7.10.

<table>
<thead>
<tr>
<th></th>
<th>current situation (n = 23)</th>
<th>faster advice (n = 23)</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
</tr>
<tr>
<td>life span A-applic.</td>
<td>87.8</td>
<td>5.76</td>
<td>72.5</td>
</tr>
<tr>
<td>life span B-applic.</td>
<td>70.1</td>
<td>3.44</td>
<td>55.4</td>
</tr>
<tr>
<td>number of A accepted</td>
<td>64.5</td>
<td>7.98</td>
<td>67.0</td>
</tr>
<tr>
<td>number of B accepted</td>
<td>88.4</td>
<td>8.92</td>
<td>86.3</td>
</tr>
<tr>
<td>occupation constr. eng.</td>
<td>91.6</td>
<td>3.45</td>
<td>92.8</td>
</tr>
</tbody>
</table>

\[ t_{0.025, 44} = 2.016 \quad t_{0.05, 44} = 1.680 \quad \text{n.s.: not significant} \]

Table 7.10 Results of a faster advice

The measure certainly establishes a significant reduction of time spans. For both A and B-applications these time spans improve with more than two weeks. The improvements attained by this measure are illustrated in figure 7.17 where the frequency distribution for time spans is projected for both the current situation, as well as the situation where a fast advice is given.

Clearly, the histogram for the situation where an advice is given more quickly is shifted to the left. Also, the variance is reduced by decreasing the maximum time span from over 200 days to 150 days. Despite these improvements, however, 20% of the applications is still finished after 90 days, and will hence be too late when the new housing act is in force.
Figure 7.17  Time span for A-applications in current situation and with faster advice

Cause 4: Delays caused by the client
The fourth and final cause mentioned in the previous section is the delays which are caused by clients during the process. Specifically, much time is lost when a client wants to revise his building plans.

In our fourth experiment the possibility of revising building plans is removed from the model. Hence, applications will be rejected immediately if serious objections are made. The difference with the changed policy discussed under cause 2 is the fact that the construction engineers will still make a great effort to try to accept the application. If this is impossible, however, the application is rejected instead of offering the possibility to revise the plan.

The expected effect of the change is a decrease in the time span for A-applications. The number of accepted A-applications will, however, decrease due to this measure. The effect for B-applications will be minimal due to the fact that these less often give rise to objections. The results of the experiment are listed in table 7.11.
Case 3: A municipal department

<table>
<thead>
<tr>
<th>current situation (n = 23)</th>
<th>exclude revisions (n = 23)</th>
<th>( t )</th>
<th>0.05</th>
<th>0.10</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
<td>s.d.</td>
<td></td>
</tr>
<tr>
<td>life span A-applic.</td>
<td>87.8</td>
<td>5.76</td>
<td>83.9</td>
<td>5.72</td>
</tr>
<tr>
<td>life span B-applic.</td>
<td>70.1</td>
<td>3.44</td>
<td>68.4</td>
<td>5.12</td>
</tr>
<tr>
<td>number of A accepted</td>
<td>64.5</td>
<td>7.98</td>
<td>49.7</td>
<td>8.04</td>
</tr>
<tr>
<td>number of B accepted</td>
<td>88.4</td>
<td>8.92</td>
<td>85.9</td>
<td>8.99</td>
</tr>
<tr>
<td>occupation constr. eng.</td>
<td>91.6</td>
<td>3.45</td>
<td>87.9</td>
<td>2.36</td>
</tr>
</tbody>
</table>

\( t_{0.025, 44} = 2.016 \) \( t_{0.05, 44} = 1.680 \) n.s.: not significant

Table 7.11 Results of excluding plan revisions

As could be expected, the results are less promising than those of using the new policy (table 7.9). Some improvement is made, although the decrease in time span of four days does not really solve any problems. As was the case with the change in policy, the increase in the number of rejected applications will have some ill effects on the work load of the legal assistants.

In addition to evaluating the four causes identified earlier, a fifth experiment is performed to determine the effect of reducing the number of phone calls and the number of clients who need to be served at the department’s service desk. With the intention of the city council to create a single service desk for all the municipal departments, this reduction can be achieved. As the phone calls and desk services interrupt the construction engineers throughout the day, an improvement can be expected from such a reduction. The results of the experiment are listed in table 7.12.

By reducing the number of service tasks for construction engineers, their occupation level decreases. However, the time spans for both A and B-applications does not improve significantly. These results improve the confidence in our belief that the long time spans in the process are mainly related to the cooperation and coordination of the many different departments involved in the process.
Section 7.4 Diagnosis, solution finding and evaluation

<table>
<thead>
<tr>
<th></th>
<th>current situation (n = 23)</th>
<th>less service tasks (n = 23)</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
</tr>
<tr>
<td>life span A-applic.</td>
<td>87.8</td>
<td>5.76</td>
<td>85.3</td>
</tr>
<tr>
<td>life span B-applic.</td>
<td>70.1</td>
<td>3.44</td>
<td>68.3</td>
</tr>
<tr>
<td>number of A accepted</td>
<td>64.5</td>
<td>7.98</td>
<td>64.3</td>
</tr>
<tr>
<td>number of B accepted</td>
<td>88.4</td>
<td>8.92</td>
<td>86.9</td>
</tr>
<tr>
<td>occupation constr. eng.</td>
<td>91.6</td>
<td>3.45</td>
<td>85.6</td>
</tr>
</tbody>
</table>

\[ t_{0.025, 44} = 2.016 \quad t_{0.05, 44} = 1.680 \]

n.s.: not significant

Table 7.12 Results of reducing the number of service tasks

Up till now, the best improvement is obtained by reducing the time span for advices from other departments and from the building committee. Before we turn to an evaluation of the different proposals discussed above, however, two combinations of change alternatives are evaluated.

First, the reduction of the time span for advices is combined with excluding plan revisions. Also, in light of the service desks under development, the number of service tasks for construction engineers is reduced. The results of this experiment are listed in table 7.13.

<table>
<thead>
<tr>
<th></th>
<th>current situation (n = 23)</th>
<th>combination 1 (n = 23)</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
</tr>
<tr>
<td>life span A-applic.</td>
<td>87.8</td>
<td>5.76</td>
<td>70.1</td>
</tr>
<tr>
<td>life span B-applic.</td>
<td>70.1</td>
<td>3.44</td>
<td>56.5</td>
</tr>
<tr>
<td>number of A accepted</td>
<td>64.5</td>
<td>7.98</td>
<td>49.9</td>
</tr>
<tr>
<td>number of B accepted</td>
<td>88.4</td>
<td>8.92</td>
<td>86.1</td>
</tr>
<tr>
<td>occupation constr. eng.</td>
<td>91.6</td>
<td>3.45</td>
<td>87.6</td>
</tr>
</tbody>
</table>

\[ t_{0.025, 44} = 2.016 \quad t_{0.05, 44} = 1.680 \]

n.s.: not significant

Table 7.13 Results of excluding plan revisions and fast advice
The time spans for both A and B-applications are reduced significantly, although the extra improvements in comparison with the experiment with fast advice are minimal. The only difference is the fact that this time the occupation level for construction engineers is also reduced.

The second combination joins the new policy with a faster advice from other departments. Again, the reduction of service tasks is included in the combination. The results are listed in Table 7.14.

<table>
<thead>
<tr>
<th></th>
<th>current situation (n = 23)</th>
<th>combination 2 (n = 23)</th>
<th>significance level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>s.d.</td>
<td>mean</td>
</tr>
<tr>
<td>life span A-applic.</td>
<td>87.8</td>
<td>5.76</td>
<td>66.2</td>
</tr>
<tr>
<td>life span B-applic.</td>
<td>70.1</td>
<td>3.44</td>
<td>56.3</td>
</tr>
<tr>
<td>number of A accepted</td>
<td>64.5</td>
<td>7.98</td>
<td>39.2</td>
</tr>
<tr>
<td>number of B accepted</td>
<td>88.4</td>
<td>8.92</td>
<td>87.9</td>
</tr>
<tr>
<td>occupation constr. eng.</td>
<td>91.6</td>
<td>3.45</td>
<td>86.1</td>
</tr>
</tbody>
</table>

\[ t_{0.025, 44} = 2.016 \quad t_{0.05, 44} = 1.680 \]

Table 7.14 Results of new policy and fast advice

Again, a significant improvement is obtained for the time spans of A and B-applications. This time, the average time span for A-applications even improves with an additional 6 days in comparison with the situation where only a faster advice is given. The increase in the number of rejections, however, may be disadvantageous for the legal assistants.

Now that the improvements which can be obtained with the changes proposed above have been established, an evaluation of the alternatives is called for. First, the difficulties of implementing each change are assessed. Next, the extent to which each alternative resolves the department’s problem is determined.

For the first proposal, where the construction engineers start sooner with new applications, minor changes need to be made to the way in which construction engineers perform their work. The reward for such a simple change is a reduction of 3 to 4 days on the average time span for both A and B-applications.
The second change, where a different policy is introduced, is also easy to implement. The construction engineers only have to change their procedures for A-applications which contradict town development plans. These applications are refused. An advantage of this simplified procedure is that uncertainty is reduced for the department’s personnel, as well as for its clients. Apart from reducing uncertainty, the advantage of the new policy is a significant reduction of the time spans for A-applications with an average of 11 days. A disadvantage can be that more applicants will appeal against a refusal when the number of refused applications increases.

As a third experiment, it is proposed that the time span for advice from other departments should be reduced. In addition to this, the construction engineers should continue their work on an application immediately after an advice is obtained. To implement these measures an agreement should be made with the other departments about a maximum delay. For the building committee, special measures should be taken to increase the frequency of the visits of the district architect. Also, an advice from the building committee should be obtained more quickly when the architect disapproves the plans. Finally, the construction engineers will have to review their priorities to make sure that applications will not be further delayed after the advice has been received.

For the fourth proposal, where a revision of building plans is excluded from the process, a minor change needs to be made. The only difference is the fact that if serious objections are made against a building plan, the construction engineer will reject it immediately, whereas at the moment the client is offered the possibility to revise his plans. By implementing this minor change, however, only a slight improvement is made for the time span of A-applications.

The fifth proposal is to reduce the number of service tasks for construction engineers. The idea behind this proposal is that when a single service desk is made for the different departments, this service desk will answer the phone calls and help the clients with their questions, whereas the construction engineers need to perform these tasks at the moment. Hence, to obtain the proposed reduction of service tasks, the integrated service desk needs to be created. The effect of this measure is that the occupation level for construction engineers decreases. The improvements for the time span for A and B-applications is, however, not significant.

Finally, the required changes for implementing the two combined proposals discussed above is simply the combination of the measures that need to be taken for each individual proposal.

To determine the extent to which each of the alternatives proposed can resolve the department’s problem, the percentage of A and B-applications which take more than 90 days to be finished is determined.
In addition to this, three definitions of the time span are introduced:

1. The first definition is what is used at the moment: The time span is the time which passes between receiving an application and taking a decision. Obviously, if a client withdraws the application before a decision is taken, the time span will end at that time.

2. In the second definition of time span, the process ends when a client wants to revise the building plan. Hence, when a revised plan is made available, the time span starts at zero again.

3. The third definition of time span measures the same amount of time which is used in the first definition, but subtracts the amount of time that was used by the client to make revisions. When a client wants to revise the building plan, the time span stops, and its measurement is resumed as soon as the revised plans are made available.

With the second definition, the high degree of service provided at the moment is replaced by a policy of rather blunt rejections. If objections are found, the application is refused immediately. Revised building plans are considered to be entirely new plans for which a time span of three months is again available.

In table 7.15 the percentage of applications which takes more than 90 days is listed for the current situation, as well as for each of the alternatives proposed above.

<table>
<thead>
<tr>
<th></th>
<th>A-applications</th>
<th>B-applications</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Def. 1</td>
<td>Def. 2</td>
</tr>
<tr>
<td>Current Situation</td>
<td>37 %</td>
<td>5 %</td>
</tr>
<tr>
<td>Faster start</td>
<td>36 %</td>
<td>5 %</td>
</tr>
<tr>
<td>New policy</td>
<td>28 %</td>
<td>4 %</td>
</tr>
<tr>
<td>Faster advice</td>
<td>20 %</td>
<td>3 %</td>
</tr>
<tr>
<td>Exclude revisions</td>
<td>33 %</td>
<td>9 %</td>
</tr>
<tr>
<td>Decrease service tasks</td>
<td>31 %</td>
<td>4 %</td>
</tr>
<tr>
<td>Combination 1</td>
<td>23 %</td>
<td>7 %</td>
</tr>
<tr>
<td>Combination 2</td>
<td>20 %</td>
<td>5 %</td>
</tr>
</tbody>
</table>

Table 7.15 Percentage of applications finished late for each proposed change
Section 7.4  Diagnosis, solution finding and evaluation

From these results it is clear that a fast advice provides the best results for A-applications, while combination 2 is best for B-applications. The table also demonstrates that if the current definition is used, there is still a significant number of applications which take longer than 90 days to be finished. A change in definition can resolve the problem, although this may have some undesirable side effects. With the second definition, the number of refused applications is raised, which may cause an increase in the number of appeals made against the department’s decisions. The third definition requires a good registration.

Based on the results presented so far, the department’s staff proposed to evaluate one final alternative to try to resolve the problem without changing the definition of the time span. In this final proposal, the number of sketch plans is increased. A sketch plan provides the possibility that the first checks are performed while the actual application is not yet received. The most important delay that is eliminated by using such sketch plans is the delay for the building committee. As this committee is primarily interested in the external appearance of a building, the technical details do not have to be available for them to give an advice. Hence, the building committee can give their advice based on the sketch plan while the official application is not yet received.

In table 7.16 the percentage of applications finished after 90 days is listed for the different alternatives with a varying percentage of sketch plans.

<table>
<thead>
<tr>
<th>Percentage sketch plans</th>
<th>A-applications</th>
<th>B-applications</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10 % 30 % 50 % 70 % 90 %</td>
<td>10 % 30 % 50 % 70 % 90 %</td>
</tr>
<tr>
<td>Current Situation</td>
<td>34 % 26 % 20 % 14 % 9 %</td>
<td>21 % 18 % 13 % 9 % 3 %</td>
</tr>
<tr>
<td>Faster start</td>
<td>33 % 25 % 20 % 14 % 9 %</td>
<td>21 % 17 % 13 % 9 % 3 %</td>
</tr>
<tr>
<td>New policy</td>
<td>30 % 21 % 18 % 11 % 9 %</td>
<td>21 % 17 % 12 % 8 % 3 %</td>
</tr>
<tr>
<td>Faster advice</td>
<td>22 % 16 % 10 % 10 % 3 %</td>
<td>9 % 6 % 5 % 3 % 2 %</td>
</tr>
<tr>
<td>Exclude revisions</td>
<td>35 % 25 % 22 % 12 % 9 %</td>
<td>21 % 17 % 12 % 10 % 3 %</td>
</tr>
<tr>
<td>Decrease service tasks</td>
<td>30 % 24 % 20 % 13 % 11 %</td>
<td>24 % 18 % 12 % 9 % 3 %</td>
</tr>
<tr>
<td>Combination 1</td>
<td>20 % 14 % 15 % 8 % 6 %</td>
<td>11 % 10 % 6 % 4 % 1 %</td>
</tr>
<tr>
<td>Combination 2</td>
<td>20 % 12 % 13 % 8 % 7 %</td>
<td>11 % 9 % 5 % 5 % 2 %</td>
</tr>
</tbody>
</table>

Table 7.16 Percentage of applications finished late while using sketch plans

Evidently, by increasing the percentage of sketch plans, and by implementing the alternative of fast advice or one of the combinations, the department’s problem can nearly be resolved without changing the time span definition. For the few
applications that would still pass the time limits of 90 days, the decision should be postponed. These postponements may only be used occasionally according to the new housing act implying that the percentage should be kept as low as possible. If the current definition of time span is upheld, half of the A-applications should be preceded by a sketch plan. For B-applications 30% sketch plans will suffice.

7.5 Summary and conclusions

In this chapter we reported on a business reengineering process within a department of a Dutch municipality, which was aimed at resolving a problem that has arisen due to legislative changes proposed by the Dutch government. Similar to our previous two case studies a detailed description is made of the current situation within the department. Based on this description an initial set of causes for the perceived problem of long time spans for building permit decisions is identified. By using a dynamic model system of the department the exact contribution of each of these causes is determined. As it turned out, only two of the causes were actually accountable for the long time spans: (1) long delays for advices from other departments, and (2) extensive efforts to help clients to get their building plans accepted.

As a result of the problem diagnosis, different solution strategies emerged. The effect of these alternative solutions was evaluated. As it turned out, the time span of 90 days which was aimed for was still too short for a significant portion of the applications. In order to completely resolve the department’s problem either a change in the definition of the time span is required, or a number of applications should be preceded by so-called sketch plans which enable the department to perform part of the process before the actual application is received.

Similar to our previous two chapters, the conclusions are divided into two categories. The first conclusions relate to our dynamic modelling approach and its use within the business reengineering process. The second type of conclusions relate to the specific organization which is analyzed in this chapter.

1. The dynamic modelling approach

With respect to the dynamic modelling approach we can conclude that the use of a dynamic model system helps to set a yardstick for problems in the existing situation, and to objectively evaluate the effectiveness of alternative courses of action. Although a set of causes for the problem was already derived from the static entity and task descriptions, the exact contribution of these causes to the problem could not yet be assessed. With the results of the experiments performed with different model systems, the effect of alternative courses of action could be determined and a choice could be made.
A second conclusion is that during the evaluation, more considerations need to be taken into account than just the model system output. The difficulties of implementing each alternative solution need to be determined, and the side effects on parts of the organization which were not contained within the modelled scope should be made clear. Based on these different considerations, and supported by the knowledge about what the organization will look like after each solution is implemented, the department’s management and personnel will be able to make a well reasoned choice.

2. The department of building regulation

The expertise of the department’s personnel made it possible to locate the major causes for the department’s problem in an early stage. This expertise can, however, be deceptive if it cannot be objectively tested in one way or another. For the department of building regulation it is shown that a clear description of the current organization, and the analysis of the behavioral aspects thereof, provides such an objective testing base. The problem diagnosis performed with the descriptive model then leads to a set of solution strategies for which the pros and the cons can be determined.
8

Epilogue

8.1 Introduction

In this thesis we have addressed the problem of supporting people who have to contrive, design, evaluate, plan and implement changes in an organization in order to cope with changes in the organization's environment. We observed that today's post-industrial society is not stable, and is characterized by increasing complexity, hostility and turbulence. This unstable environment is more taxing on an organization's adaptability, i.e. its capacity for coping with change, than a stable one.

Our research has focused on those organizations which provide services that have a high information content. The intangible characteristics of these services cause the knowledge about the business processes to be dispersed and partial for every organization member. The tacit knowledge that resides in the heads and hands of the organization's staff, the many vantage points from which the organization can be viewed, and the complexity of the business processes make it hard for even the most capable manager to fully understand the current situation and the impact of organizational change.

To improve the support offered to those who need to decide about appropriate changes for an organization, the use of a dynamic modelling approach is proposed in this thesis. This approach, as well as the environment which is developed to support it, are applied in three organizations. In this concluding chapter the research findings collected in these three organizations are reviewed, and the consequences of these findings are discussed.
8.2 Research findings

As stated in chapter 1, the objective of our research is to come up with a way to effectively support the analysis of business processes in information intensive organizations and to support decision making on rearrangements in these business processes. Based on a review of literature concerning organization design and problem solving the requirements for such support have been made clear. These requirements, combined with an assessment of current methods, are used to deduce two major guidelines for improved support: (1) a comprehensible representation of the current organization should be made available, and (2) the effects of organizational change should be made clear in advance to its implementation. At the closing of chapter 2 a protective belt consisting of four hypotheses is presented which covers these guidelines. The first hypothesis captures the objective of our dynamic modelling approach. The remaining three hypotheses discuss the characteristics of our dynamic modelling approach and the environment supporting it. These characteristics are discussed first:

Hypothesis 2

*Comprehensible models of information intensive organizations can be devised by using three concepts which are easy to use for non-modelling experts: information worker, information processing task and information item.*

In chapter 3 these modelling concepts are captured in specific modelling techniques. The concepts *information worker* and *information item* are part of the entity model, where they are represented by two distinguished classes: *Actor* and *Item* respectively. For *information processing task* a representation is introduced which makes explicit the subtasks of which an information processing task is comprised, and the sequence in which these are performed.

In all three case studies the modelling concepts, along with the graphical representations that were used, facilitated models which could easily be understood by the participants, despite the fact that these were in most cases not specifically trained in modelling techniques. The fact that the concepts used are not abstract artifacts but relate directly to real world phenomenon which can be observed in the organization enhanced the comprehensibility of the eventual models. Also, the set of three modelling concepts turned out to be sufficient for the domain we selected, i.e. information intensive organizations. Hence, our findings in the three case studies discussed support this hypothesis.
Hypothesis 3

The effect of changes in information intensive organizations can be made clear from the micro, meso and macro perspective if the behavior of the business processes can be exposed, i.e. if these business processes can be simulated over time.

In chapter 3 a mechanism is introduced which enables the analysis of dynamic model behavior. Without changing the conceptual framework, a dynamic model system can be generated based on an entity and functional model.

In the three case studies we performed, the use of simulation has proven to be very fertile. First, the use of a graphical animation of the descriptive model system is used in all three situations to enhance the participant's understanding of the current situation within their organization. Next, simulation is used to diagnose perceived problems and to evaluate effects of organizational change.

With the mechanism we introduced, consequences of organizational changes are made clear at all three perspectives: At the macro level, the contribution of an organizational change for the service level provided to the organization's clients is made clear. At the meso level, it is made clear which coordination is required between the different people and departments involved, and how this coordination is to be obtained. Finally, at the micro level each individual is able to see in advance what the consequences of an organizational change are for his or her specific working environment and working habits.

In each case study the consequences of organizational changes are considered simultaneously from all three perspectives. Hence, our findings uphold this hypotheses.

Hypothesis 4

The modelling approach envisioned by the preceding statements can be adequately supported by an environment comprised of support tools which are constructed using information technology available today.

Preliminary evidence for this hypothesis was already given in chapter 4, where an analysis of the functional requirements for a dynamic modelling support environment yielded no elements which could not be realized using information technology which is available today. This is further demonstrated by our prototypical implementation of this support environment.

Our implementation of the support environment is applied in a practical setting during the three case studies. The experiences of using this environment are very promising. The most prominent aspect is the fact that changes to a model can be made using graphical modelling tools, where a dynamic model system is generated by the environment without requiring special coding efforts. This was specifically useful during experimentation with alternative model systems. Proposed changes
were incorporated in the model and the effect was determined by examining the resulting animated simulation. A drawback of the prototype environment was the relatively low simulation speed. Some of the experiments discussed in the previous chapters took more than 10 hours to perform. Although this requires some patience from the analyst, it did not have a significant effect on the change process itself. As it turned out, the participant’s agendas imposed more constraints for arranging a next session than the lead time for the experiments did.

In short we can say that our hypothesis is upheld by the results of applying the prototype environment in a practical setting.

The first hypothesis summarizes the objective of our research:

Hypothesis 1

*By providing comprehensible models of an organization’s business processes and by facilitating a mechanism to evaluate the impact of changes to the business processes from the micro, meso and macro perspective, adequate support for the business reengineering process in information intensive organizations is obtained.*

Without exception the case studies discussed in chapters 5 to 7 provide evidence for this statement. The point of departure in all three cases was one where a problem needed to be solved or opportunities for improvements were sought for. Existing methods and techniques, however, had not provided effective courses of action which could be implemented.

Our proposed dynamic modelling approach has been applied in three organizations. First, a detailed model of the current situation was constructed, verified and validated. The graphical representations that were used to present the model to the participants in the organization proved to be clear even for those who were not specifically trained in modelling techniques. This, in conjunction with our use of graphical animated simulation techniques, facilitated a clear understanding of the current organization for all those involved. It also evoked the participants to contribute actively to the solution finding endeavour. To accumulate the evidence for this hypothesis the findings of each of our case studies are summarized:

1. **The government department (chapter 5)**

   The problem that prompted our first case study was the fact that many transaction forms were finished late, implying that the department’s personnel had to perform complex tasks manually while these can normally be performed by the department’s computer system.

   By using a detailed description of the current situation it was made clear that the solution which was proposed by the project team would not be able to resolve the perceived problem. Due to the fact that the participants had obtained
a clear understanding of their organization, however, they were able to come up with an easy to implement and effective solution strategy — a solution which did not require towering investments.

2. The insurance firm (chapter 6)

The second case study was not initiated from a problem statement, but rather from a belief of the firm’s management that improvements could be made to the level of service provided to its clients.

Based on a detailed model of one of the firm’s departments a variety of improvements were proposed by the department’s personnel and its management. The improvements included small changes which were extrapolated from the status quo, as well as more extensive changes which were used to determine the great potential of different arrangements of business processes when supplemented by the use of modern technologies. Again, the fact that the current situation was understood clearly by all participants resulted in a set of effective change strategies.

3. The municipal department (chapter 7)

Our last case study was triggered by a change in legislation which affected a municipal department. To meet the requirements posed by the new legislation, changes had to be made to the working procedures used at the department.

Using a description of the current situation, a wide range of alternative changes could be analyzed. Although a number of causes for the perceived problem were already identified initially, the contribution of each of these causes could be assessed accurately using the model constructed. As a result of the analysis, some of the causes initially identified turned out to have no effect. Hence, the modelling effort has established a clear understanding of which courses of action can effectively be pursued.

In all case studies, use of the dynamic modelling approach yielded change alternatives which are determined to be effective in resolving the problems perceived. We conclude that our protective belt can not be falsified by our case study findings:

The dynamic modelling approach which emphasizes the use of a detailed descriptive model and of simulation techniques to determine the effect of organizational change, adequately supports the process of reengineering information intensive organizations.
In our research, the feasibility and effectiveness of dynamic modelling as an approach to support organizational change has been made clear. Based on our findings some directions for future research can be indicated.

First, the desired level of detail at which a description of the current organization should be made, needs to be determined. An important consideration in this respect is the effect of detailed descriptions of an existing situation on the imagination of the participants when they are designing changes. When the existing situation is described in too much detail, people may be inclined to come up with only minor changes of the status quo. When, on the other hand, a recognizable representation of the current situation is lacking, participants will have difficulty finding effective propositions for change. In addition to these considerations, the economic factor is also important. A detailed description requires a significant amount of time and energy, the costs of which need to be balanced against potential benefits of the modelling effort.

A second direction for future research is the use of animated simulation. Which aspects of an organization need to be represented in an animation in order to obtain recognizable models? Especially the use of more than one view of the organization has not been considered in our research. Our experiences do point out, however, that too much detail in an animation may distract the participant’s attention from matters that are really important, implying that animation should be used cautiously.

### 8.3 Consequences for information systems development

The organizations we have focused on in our research have business processes which primarily concern information processing. As such, our domain of interest coincides partly with those organizations in which information systems are developed and maintained on a regular basis.

Without exception our findings point out that significant improvements can be obtained in information intensive organizations if changes are made to the way in which the people and departments operate. By reengineering the main business processes an organization can reduce operational costs, and at the same time improve the level of services provided to its clients. In the practice of information systems development, however, changes to the tasks and coordination of information workers in organizations are often neglected. Instead, as Thiadens (1992) points out, information technology is used to directly automate the existing situation.

The need for more rigorous changes in organizations in order to meet new service level demands and to raise productivity in the service sector has recently been pointed out quite often in literature (Huber 1990; Clemons and Weber 1991; Drucker 1991a; Keen 1991; Meier and Sprague 1991; Michels and Welsh 1991; Sol 1992). The potential of such a reengineering approach has also been indicated (Hammer 1990; Roach 1991). As a result, business reengineering approaches are
currently being advocated by almost every management consultancy firm, although these approaches still tend to lack a sound basis of methods and techniques. In this thesis, we not only pointed out the need and the potential of reengineering business processes, but also indicated a modelling approach which facilitates the actual implementation of business reengineering in organizations.

Based on our research findings, we propagate a change in information systems development practice. It is our belief that an assessment of the potential of organizational changes should be made, prior to taking a decision regarding the use of new information technologies. Such an assessment can be used to determine the need for information technology, and if such a need is made clear, it can be used to determined whether this technology should be used to facilitate organizational changes, or just to automate the current situation. Such an assessment does not have to involve a full-fledged analysis similar to the ones presented in this thesis. A quick inventory of those operations which are crucial to a firm’s success, supplemented with some type of quick dynamic modelling can prevent a firm from making unnecessary investments while alternative measures are available (Dur and Van Meel 1991).

For the information systems development practice we therefore conclude:

In the practice of information systems development, more attention should be paid to potential improvements which can be obtained by changing business processes in organizations. By introducing new technologies without changing the business processes the capitalization of investments can be far from optimal.

With the great challenge of productivity and earning capacity increases sought for in the service industry this change in attitude is, to our opinion, a prerequisite for success.
References

References


References


References


References


228
References

References


References


References


References

Author index

A
Aburdene, P.  3
Ackoff, R.L.  3, 12
Adams, D.A.  20, 21
Agre, P.G.  12
Allemekinders, P.  146
Anzenhofer, K.  69

B
Balbo, G.  45
Banks, J.  80
Bell, P.C.  20
Benbasat, I.  9
Berger, P.  1
Bergeron, F.  20
Betz, F.  17
Bhattacharyya, G.K.  197
Birtwistle, G.M.  46
Bolckow, R.  81
Borland 81
Bostrom, R.P.  19
Bots, P.W.G.  11, 17, 22, 31, 33, 81, 197
Bowen, W.  5
Bracchi, G.  28
Brancheau, J.C.  2
Bravoco, R.R.  45
Brightman, H.J.  14
Brodie, M.L.  28
Brussaard, B.K.  121
Bubenko, J.A., Jr.  19
Burns, J.R.  80

C
Cameron, K.  3
Checkland, P.  16
Chen, M.  2, 20
Child, I.L.  18
Child, J.  22
Churchman, C.W.  17
Clemons, E.K.  1, 218
Conrath, D.W.  44
Cox, J.  15

D
Dachouve, N.  20, 70
Daft, R.L.  17
Dahl, O.J.  31, 46
De Leeuw, A.C.J.  67
De Sitter, U.  67
DeMarco, T.  19
Dennis, J.B.  45
Derrick, E.J.  46
Di Felice, P.  44
Drucker, P.F.  2, 3, 5, 218
Duncan, R.B.  3
Dur, R.C.J.  31, 81, 85, 87, 89, 107, 197, 219

E
El Sawy, O.A.  2
Emery, F.E.  3

F
Fähnrich, K.P.  81
Featheringham, F.R.  11
Fleischer, M.  5
### Author Index

<table>
<thead>
<tr>
<th>Author</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Galbraith, J.R.</td>
<td>3, 17, 22</td>
</tr>
<tr>
<td>Genrich, H.J.</td>
<td>45</td>
</tr>
<tr>
<td>George, C.S., Jr.</td>
<td>6</td>
</tr>
<tr>
<td>Goldratt, E.M.</td>
<td>15</td>
</tr>
<tr>
<td>Gordon, G.</td>
<td>80</td>
</tr>
<tr>
<td>Groenenboom, G.J.</td>
<td>67</td>
</tr>
<tr>
<td>Guinen, P.J.</td>
<td>19</td>
</tr>
<tr>
<td>Hammer, M.</td>
<td>v, 2, 218</td>
</tr>
<tr>
<td>Hedberg, B.L.T.</td>
<td>4</td>
</tr>
<tr>
<td>Heinen, J.S.</td>
<td>19</td>
</tr>
<tr>
<td>Higgins, L.F.</td>
<td>31</td>
</tr>
<tr>
<td>Hills, P.R.</td>
<td>46</td>
</tr>
<tr>
<td>Hirschheim, R.A.</td>
<td>1, 19</td>
</tr>
<tr>
<td>Hoare, C.A.R.</td>
<td>27</td>
</tr>
<tr>
<td>Hoover, S.V.</td>
<td>62, 80</td>
</tr>
<tr>
<td>Huber, G.P.</td>
<td>1-4, 67, 68, 218</td>
</tr>
</tbody>
</table>

### I

<table>
<thead>
<tr>
<th>Author</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ives, B.</td>
<td>20</td>
</tr>
</tbody>
</table>

### J

<table>
<thead>
<tr>
<th>Author</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jackson, M.</td>
<td>19</td>
</tr>
<tr>
<td>Jensen, K.</td>
<td>45</td>
</tr>
<tr>
<td>Johnson, R.A.</td>
<td>197</td>
</tr>
<tr>
<td>Joshi, K.</td>
<td>68</td>
</tr>
</tbody>
</table>

### K

<table>
<thead>
<tr>
<th>Author</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kamper, S.</td>
<td>45</td>
</tr>
<tr>
<td>Kappel, G.</td>
<td>45</td>
</tr>
<tr>
<td>Karam, G.M.</td>
<td>28</td>
</tr>
<tr>
<td>Kaye, A.R.</td>
<td>28</td>
</tr>
<tr>
<td>Keen, P.G.W.</td>
<td>1-3, 12, 20, 23, 68, 218</td>
</tr>
<tr>
<td>Kieser, A.</td>
<td>21, 22</td>
</tr>
<tr>
<td>Klein, H.K.</td>
<td>19</td>
</tr>
<tr>
<td>Kleijn, J.P.C.</td>
<td>67</td>
</tr>
<tr>
<td>Kramer, J.</td>
<td>70</td>
</tr>
<tr>
<td>Kriz, J.</td>
<td>81</td>
</tr>
<tr>
<td>Kubicek, H.</td>
<td>21, 22</td>
</tr>
<tr>
<td>Kung, C.H.</td>
<td>19</td>
</tr>
</tbody>
</table>

### L

<table>
<thead>
<tr>
<th>Author</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lakatos, I.</td>
<td>9, 24</td>
</tr>
<tr>
<td>Land, F.</td>
<td>19</td>
</tr>
<tr>
<td>Lang, J.R.</td>
<td>14</td>
</tr>
<tr>
<td>Langefors, B.</td>
<td>19</td>
</tr>
<tr>
<td>Lawrence, P.R.</td>
<td>22</td>
</tr>
<tr>
<td>Lazarević, B.</td>
<td>44</td>
</tr>
<tr>
<td>Learmonth, G.P.</td>
<td>20</td>
</tr>
<tr>
<td>Lederer, A.L.</td>
<td>20</td>
</tr>
<tr>
<td>Lee, M.C.S.</td>
<td>20, 21</td>
</tr>
<tr>
<td>Lee, R.M.</td>
<td>45</td>
</tr>
<tr>
<td>Lengel, R.H.</td>
<td>17</td>
</tr>
<tr>
<td>Lesuisse, R.</td>
<td>20, 70</td>
</tr>
<tr>
<td>Lewin, K.</td>
<td>12</td>
</tr>
<tr>
<td>Lincoln, T.</td>
<td>1, 21</td>
</tr>
<tr>
<td>Lochovsky, F.H.</td>
<td>31, 44</td>
</tr>
<tr>
<td>Lockemann, P.C.</td>
<td>31</td>
</tr>
<tr>
<td>Lorsch, J.W.</td>
<td>22</td>
</tr>
<tr>
<td>Loucopoloulos, P.</td>
<td>45</td>
</tr>
<tr>
<td>Lundeborg, M.</td>
<td>19</td>
</tr>
</tbody>
</table>

### M

<table>
<thead>
<tr>
<th>Author</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macdonald, I.G.</td>
<td>19</td>
</tr>
<tr>
<td>Macintosh, N.B.</td>
<td>17</td>
</tr>
<tr>
<td>MacMillan, I.C.</td>
<td>20</td>
</tr>
<tr>
<td>Mansell, G.</td>
<td>9</td>
</tr>
<tr>
<td>Marsden, J.R.</td>
<td>20, 21</td>
</tr>
<tr>
<td>Martin, J.</td>
<td>3, 20</td>
</tr>
<tr>
<td>Mason, R.O.</td>
<td>17</td>
</tr>
<tr>
<td>Massapati, G.H.</td>
<td>45</td>
</tr>
<tr>
<td>Masuda, Y.</td>
<td>3</td>
</tr>
<tr>
<td>McDaniel, R.R., Jr.</td>
<td>1, 2, 4</td>
</tr>
<tr>
<td>McFarlan, F.W.</td>
<td>20</td>
</tr>
<tr>
<td>McIntyre, S.C.</td>
<td>31</td>
</tr>
<tr>
<td>Meier, J.</td>
<td>15, 218</td>
</tr>
<tr>
<td>Michels, A.J.</td>
<td>5, 218</td>
</tr>
<tr>
<td>Miles, R.E.</td>
<td>3</td>
</tr>
<tr>
<td>Millar, V.E.</td>
<td>20</td>
</tr>
<tr>
<td>Mintzberg, H.</td>
<td>4, 14, 16, 18, 22, 67</td>
</tr>
<tr>
<td>Mišić, V.</td>
<td>44</td>
</tr>
<tr>
<td>Mitroff, I.I.</td>
<td>11, 13, 17</td>
</tr>
<tr>
<td>Moon, D.A.</td>
<td>31</td>
</tr>
<tr>
<td>Morell, J.A.</td>
<td>5</td>
</tr>
<tr>
<td>Morgeson, J.D.</td>
<td>80</td>
</tr>
<tr>
<td>Mosser, T.</td>
<td>20, 44</td>
</tr>
<tr>
<td>Author</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>Naisbitt, J.</td>
<td>3</td>
</tr>
<tr>
<td>Ng, K.</td>
<td>70</td>
</tr>
<tr>
<td>Nierstrasz, O.M.</td>
<td>31</td>
</tr>
<tr>
<td>Nolan, R.L.</td>
<td>20</td>
</tr>
<tr>
<td>Norton, D.P.</td>
<td>20</td>
</tr>
<tr>
<td>Novels, M.D.</td>
<td>70</td>
</tr>
<tr>
<td>Nunamaker, J.F., Jr.</td>
<td>2</td>
</tr>
<tr>
<td>Nygaard, K.</td>
<td>31, 46</td>
</tr>
<tr>
<td>Nystrom, P.C.</td>
<td>4</td>
</tr>
<tr>
<td>O</td>
<td></td>
</tr>
<tr>
<td>Olle, T.W.</td>
<td>19</td>
</tr>
<tr>
<td>Ören, T.I.</td>
<td>62</td>
</tr>
<tr>
<td>P</td>
<td></td>
</tr>
<tr>
<td>Parker, R.C.</td>
<td>21</td>
</tr>
<tr>
<td>Parsons, G.L.</td>
<td>20</td>
</tr>
<tr>
<td>Pauchant, T.C.</td>
<td>2</td>
</tr>
<tr>
<td>Pegden, C.D.</td>
<td>60, 80</td>
</tr>
<tr>
<td>Pegden, P.</td>
<td>80</td>
</tr>
<tr>
<td>Pernici, B.</td>
<td>28</td>
</tr>
<tr>
<td>Perry, R.F.</td>
<td>62, 80</td>
</tr>
<tr>
<td>Petri, C.A.</td>
<td>45</td>
</tr>
<tr>
<td>Pingry, D.E.</td>
<td>20, 21</td>
</tr>
<tr>
<td>Pooley, R.</td>
<td>44</td>
</tr>
<tr>
<td>Porter, M.E.</td>
<td>20, 121</td>
</tr>
<tr>
<td>Pritsker, A.A.B.</td>
<td>62, 80</td>
</tr>
<tr>
<td>Putnam, A.G.</td>
<td>20</td>
</tr>
<tr>
<td>Q</td>
<td></td>
</tr>
<tr>
<td>Quillian, R.</td>
<td>31</td>
</tr>
<tr>
<td>Quinn, J.B.</td>
<td>15</td>
</tr>
<tr>
<td>Quinn, R.E.</td>
<td>3</td>
</tr>
<tr>
<td>R</td>
<td></td>
</tr>
<tr>
<td>Raczynski, S.</td>
<td>81</td>
</tr>
<tr>
<td>Ramakrishna, H.V.</td>
<td>14</td>
</tr>
<tr>
<td>Raphael, B.</td>
<td>31</td>
</tr>
<tr>
<td>Reitman, W.R.</td>
<td>16</td>
</tr>
<tr>
<td>Ringuest, J.L.</td>
<td>197</td>
</tr>
<tr>
<td>Roach, S.S.</td>
<td>5, 218</td>
</tr>
<tr>
<td>Robey, D.</td>
<td>2</td>
</tr>
<tr>
<td>Rockart, J.F.</td>
<td>20</td>
</tr>
<tr>
<td>Ross, D.T.</td>
<td>19</td>
</tr>
<tr>
<td>Russel, E.C.</td>
<td>80</td>
</tr>
<tr>
<td>S</td>
<td></td>
</tr>
<tr>
<td>Sandmayr, H.</td>
<td>81</td>
</tr>
<tr>
<td>Sargent, R.G.</td>
<td>45</td>
</tr>
<tr>
<td>Scanlan, D.A.</td>
<td>44</td>
</tr>
<tr>
<td>Schein, E.H.</td>
<td>12</td>
</tr>
<tr>
<td>Schiel, U.</td>
<td>44</td>
</tr>
<tr>
<td>Schrefl, M.</td>
<td>45</td>
</tr>
<tr>
<td>Schriber, T.J.</td>
<td>80</td>
</tr>
<tr>
<td>Schwenk, C.R.</td>
<td>121</td>
</tr>
<tr>
<td>Sciore, E.</td>
<td>27</td>
</tr>
<tr>
<td>Shannon, R.E.</td>
<td>61</td>
</tr>
<tr>
<td>Sheiderman, B.</td>
<td>81</td>
</tr>
<tr>
<td>Silva, E.</td>
<td>28</td>
</tr>
<tr>
<td>Simon, H.A.</td>
<td>16, 18</td>
</tr>
<tr>
<td>Smith, D.C.P.</td>
<td>27</td>
</tr>
<tr>
<td>Smith, G.F.</td>
<td>12, 14</td>
</tr>
<tr>
<td>Smith, J.M.</td>
<td>27</td>
</tr>
<tr>
<td>Snow, C.C.</td>
<td>3</td>
</tr>
<tr>
<td>Sol, H.G.</td>
<td>1, 3, 11, 12, 16, 19, 22, 24, 31, 56, 62, 63, 67, 80, 89, 218</td>
</tr>
<tr>
<td>Solvberg, A.</td>
<td>16, 19</td>
</tr>
<tr>
<td>Sprague, R.H., Jr.</td>
<td>15, 23, 218</td>
</tr>
<tr>
<td>Stokking, E.J.</td>
<td>81</td>
</tr>
<tr>
<td>Strassmann, P.A.</td>
<td>2, 5, 21, 22</td>
</tr>
<tr>
<td>Straub, D.W.</td>
<td>2</td>
</tr>
<tr>
<td>Streng, R.J.</td>
<td>22</td>
</tr>
<tr>
<td>Susman, G.L.</td>
<td>9</td>
</tr>
<tr>
<td>Szymankiewicz, J.</td>
<td>80</td>
</tr>
<tr>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Tanenbaum, A.S.</td>
<td>75</td>
</tr>
<tr>
<td>Thesen, A.</td>
<td>64, 110</td>
</tr>
<tr>
<td>Thiadens, T.J.G.</td>
<td>218</td>
</tr>
<tr>
<td>Toffler, A.</td>
<td>3</td>
</tr>
<tr>
<td>Travis, L.E.</td>
<td>64, 110</td>
</tr>
<tr>
<td>Trommel, C.</td>
<td>193, 197</td>
</tr>
<tr>
<td>Tsalgatidou, A.</td>
<td>45</td>
</tr>
<tr>
<td>Tsichritzis, D.C.</td>
<td>31</td>
</tr>
<tr>
<td>V</td>
<td></td>
</tr>
<tr>
<td>Van Dort, E.P.W.M.</td>
<td>123</td>
</tr>
<tr>
<td>Van Eijden, I.P.</td>
<td>130, 146</td>
</tr>
<tr>
<td>Van Gundy, A.B.</td>
<td>6</td>
</tr>
<tr>
<td>Van Hee, K.M.</td>
<td>24, 45</td>
</tr>
</tbody>
</table>
Author index

Van Meel, J.W.  219
Van Soest, J.  64, 110
Verkoulen, P.A.C.  45
Verrijn-Stuart, A.A.  69
Versendaal, J.M.  81
Vonk, R.  20

W
Walsham, G.  19
Watson, R.T.  2
Weber, B.W.  1, 218
Weick, K.E.  3, 17
Weigl, K.H.  70
Weinreb, D.  31
Welke, R.J.  11
Welsh, T.  5, 218

Wetherbe, J.C.  2
White, G.M.  45
Whiting, J.W.M.  18
Wierda, F.W.  20, 22, 80
Wilson, J.R.  62
Wise, M.J.  45
Wiseman, C.  20
Woo, C.C.  31

Y
Yadav, S.B.  45
Young, L.F.  16
Yourdon, E.  19, 20

Z
Zeigler, B.P.  62
Zeigler, J.E.  81
Subject index

A
active entity 28
actor behavior 31
actor predicate 39
agenda 47
  formal definition 48
agenda mechanism
  events for 46
  formal definition 49
agenda unit 87
aggregation 29, 31, 63
ancestor predicate 37
animation 70, 75, 87
animation background 76
ANOVA test 78
assurance data net 122
attribute 28
autocorrelation 63

B
behavioral modelling 44
benefit application 97
blue print 15
bottom line 20
bounded rational 18
browsing facilities 73
building block 44
building committee 185
building permit 174
building regulations 176, 182
business case 23
business problem 12, 24
business process
  behavior of ... 24
business reengineering 6, 14
Business System Planning 20
button list 83

C
case study 9
CASE tools 20
Cinema 80
civic service center 173
class predicate 32
coding errors 62
cognitive limitations 18
competitive advantage 2
completeness 20
complex application 147
complex claim 148
complex mutation 147
component 34
compound actor 29
compound item 29
compound task 34
computer based support 68
conceptual model 13, 56
conceptualization 57
concurrency 45
condition 34
  initialization 62
  run control 63
confidence interval 110, 112, 148
confidence level 64, 78, 198
connected predicate 40
Subject index

consistency 20
c constituent 29
c constituent predicate 32
coordination 33
copy context operator 41
core unit 84
correspondence 60, 100
critical success factors 20
current job 48
customer satisfaction 67

D
D.U.B. 89
data manager 77
data set base 72
decision making
  economic theory 18
decomposition 34, 37
default rule-book 48
delay predicate 38
Department for Unemployment
  Benefits 89
descendant predicate 37
descriptive model 14, 55
diagnosis 14, 65
diagram 19
differentiation 21
direct manipulation 81
direct writer 121
disaggregation 31
distribution function 60, 75
division of labour 22
DMSE 54
cross reference table 79
functional requirements of 69
global architecture 73
implementation of 80
memory management 84
printer driver 85
programming language 80
run-time simulation library 86
support tools in 71
user interface 81
document base 72
document processor 78
Dutch Ministry of the Interior 89
dynamic modelling 24
dynamic modelling support
  environment, see DMSE
E
efficiency 2
enabling technology 2
encapsulation 31
encompass predicate 32
end-user computing 2
entity class 27
  acceptor 125
  administrative assistant 126, 177
  applicant 175
  application 124
  batch of forms 94
  benefit application 92
  building committee 178
  building permit application 175
  building supervisor 177
  C.R.A. 127
  city council 178
  claim attendant 126
  client 92, 123
  computing center 94
  construction engineer 177
  correspondence 92
  damage claim 124
  district architect 178
  expert 124
  expert attendant 126
  finance 95
  head of division 178
  head of department 177
  head of claims group 125
  head of corporate accounts 125
  head of private accounts 125
  head of office 93
  income statement 92
  intermediary 123
  investigation department 95
  legal assistant 177
entity class (continued)
  mail room 127
  manual payment 95
  mutation 124
  operations 127
  other department 178
  phone call 92
  premium dept. 127
  registration 92
  regulation clerk 93
  regulation office 93
  scanner 127
  transaction form 94
  verification group 94
  verifier 94
entity model 31
entity model base 71
entity modeller 73
entity state 73
environment 53
equivocality 17
European common market place 5, 121, 171
evaluation 67
event 46
events unit 87
experiment 62, 67, 107, 193
experimental design 67
experimental frame 60
experimentation 65
F
F-test 78
face validity 65, 112
formal definition
  agenda 48
  agenda mechanism 49
  entity model 32
  functional model 36
job 47
  task 38
  task structure 39
forms unit 84
functional model 36
functional model base 71
functional requirements 54, 68
  of the DMSE 69
G
globalization 3
Goudse Insurances 121
GPSS 80
graphic icon 69, 76
graphical user interface 81
H
hard core 24
housing act 174
I
implementation 15, 68
inactive account 95
income statement 92, 99
Information Engineering 19
information intensive organization 4
information item 22, 27
information processing task 27
information systems development
  methods 19
information technology 1
information worker 22, 27
initial component 34
initialization conditions 62
innovation 21
input data 60, 62
instantiation 31
integration 22
interpreter 75
interrupt 48
investment 67
ISAC 19
J
job 47
  active 48
  current 48
  formal definition 47
  interrupt 48
  waiting 48
join operator 42
## Subject index

<table>
<thead>
<tr>
<th>K</th>
<th>knowledge-based system 123</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>learning model 17</td>
</tr>
<tr>
<td></td>
<td>level of aggregation 59, 68</td>
</tr>
<tr>
<td></td>
<td>life-cycle 3, 46</td>
</tr>
<tr>
<td></td>
<td>lists unit 82</td>
</tr>
<tr>
<td></td>
<td>logistics 67</td>
</tr>
<tr>
<td></td>
<td>loosely coupled systems 3</td>
</tr>
<tr>
<td>M</td>
<td>macro perspective 22</td>
</tr>
<tr>
<td></td>
<td>Mann-Whitney test 78</td>
</tr>
<tr>
<td></td>
<td>market focus 5</td>
</tr>
<tr>
<td></td>
<td>mean 78</td>
</tr>
<tr>
<td></td>
<td>memory management 84</td>
</tr>
<tr>
<td></td>
<td>menu 82</td>
</tr>
<tr>
<td></td>
<td>meso perspective 23</td>
</tr>
<tr>
<td></td>
<td>metabusiness 3</td>
</tr>
<tr>
<td></td>
<td>micro perspective 23</td>
</tr>
<tr>
<td></td>
<td>model</td>
</tr>
<tr>
<td></td>
<td>conceptual 13, 56</td>
</tr>
<tr>
<td></td>
<td>descriptive 14, 55</td>
</tr>
<tr>
<td></td>
<td>dynamic 24</td>
</tr>
<tr>
<td></td>
<td>executable 56</td>
</tr>
<tr>
<td></td>
<td>manipulation 20</td>
</tr>
<tr>
<td></td>
<td>prescriptive 14, 66</td>
</tr>
<tr>
<td></td>
<td>scientific 13</td>
</tr>
<tr>
<td></td>
<td>static 23</td>
</tr>
<tr>
<td></td>
<td>structural 56</td>
</tr>
<tr>
<td></td>
<td>model frame 60</td>
</tr>
<tr>
<td></td>
<td>model system 56, 142</td>
</tr>
<tr>
<td></td>
<td>modelling concepts 19</td>
</tr>
<tr>
<td></td>
<td>information item 22</td>
</tr>
<tr>
<td></td>
<td>information processing task 22</td>
</tr>
<tr>
<td></td>
<td>information worker 22</td>
</tr>
<tr>
<td></td>
<td>multiple processor computing 45</td>
</tr>
<tr>
<td>N</td>
<td>naming conventions 69</td>
</tr>
<tr>
<td></td>
<td>non-modelling experts 16</td>
</tr>
<tr>
<td>O</td>
<td>O&amp;I: Directorate of Organization and Information 90, 91</td>
</tr>
<tr>
<td></td>
<td>object oriented modelling 31</td>
</tr>
<tr>
<td></td>
<td>object system 19</td>
</tr>
<tr>
<td></td>
<td>objective rational 18</td>
</tr>
<tr>
<td></td>
<td>opportunity 12</td>
</tr>
<tr>
<td></td>
<td>organization</td>
</tr>
<tr>
<td></td>
<td>definition 21</td>
</tr>
<tr>
<td></td>
<td>environment 3</td>
</tr>
<tr>
<td></td>
<td>information intensive 4</td>
</tr>
<tr>
<td></td>
<td>objectives 20</td>
</tr>
<tr>
<td></td>
<td>strategies 20</td>
</tr>
<tr>
<td></td>
<td>structure 4</td>
</tr>
<tr>
<td></td>
<td>organization design 2, 67</td>
</tr>
<tr>
<td></td>
<td>organizational change</td>
</tr>
<tr>
<td></td>
<td>refreezing stage 12</td>
</tr>
<tr>
<td></td>
<td>unfreezing stage 12</td>
</tr>
<tr>
<td>P</td>
<td>passive entity 28</td>
</tr>
<tr>
<td></td>
<td>Petri net 45</td>
</tr>
<tr>
<td></td>
<td>pick-list 82, 86</td>
</tr>
<tr>
<td></td>
<td>Pierson's test 197</td>
</tr>
<tr>
<td></td>
<td>post-evaluation 15, 68</td>
</tr>
<tr>
<td></td>
<td>post-industrial society 3</td>
</tr>
<tr>
<td></td>
<td>PostScript 85</td>
</tr>
<tr>
<td></td>
<td>predicate</td>
</tr>
<tr>
<td></td>
<td>actor 39</td>
</tr>
<tr>
<td></td>
<td>ancestor 37</td>
</tr>
<tr>
<td></td>
<td>class 32</td>
</tr>
<tr>
<td></td>
<td>connected 40</td>
</tr>
<tr>
<td></td>
<td>constituent 32</td>
</tr>
<tr>
<td></td>
<td>delay 38</td>
</tr>
<tr>
<td></td>
<td>descendant 37</td>
</tr>
<tr>
<td></td>
<td>encompass 32</td>
</tr>
<tr>
<td></td>
<td>structure 36</td>
</tr>
<tr>
<td></td>
<td>subtask 36</td>
</tr>
<tr>
<td></td>
<td>task 39</td>
</tr>
<tr>
<td></td>
<td>trigger 39</td>
</tr>
<tr>
<td></td>
<td>prescriptive model 14, 66</td>
</tr>
<tr>
<td></td>
<td>printer driver 85</td>
</tr>
<tr>
<td></td>
<td>private sector 5, 121</td>
</tr>
<tr>
<td></td>
<td>privatization 90</td>
</tr>
</tbody>
</table>
problem
  diagnosis 65
  ill-structured 16
  statement 57
  structured 16
  versus opportunity 12
problem solving
  general model 13
  human ... 16
procedural rational 18
productivity 5
profit 67
program base 71, 75
programming language
  Modula-2 81
  Pascal 81
  PostScript 85
  SIMULA 46
protective belt 25
prototype 9
public sector 5, 89, 173
Q
quasi-parallel processing 46
R
regulation office 92
replication 62
replicative validation 64, 108, 146, 194
research approach 7
research programme 24
research question 6, 24
response time 68
Rheden 173
rule-book 48, 61
default 48
run 62
run control conditions 63
run length 63, 109, 148, 195
S
SADT 19
sample data 151, 187
satisficing 18
scanner unit 82
scientific model 13
scroll bar 83
sequence control mechanism 46
sequencing 34
service 5
significance level 148
Siman 60, 80
simple actor 28
simple item 28
simple task 34
SIMSCRIPT 80
SIMULA 46
simulation language 80
  DEMOS 46
  GPSS 80
  Siman 60, 80
  SIMSCRIPT 80
  SIMULA 46
  SLAM 80
simulation library 86
simulation program 75
simulation routines 75
simulator 75
sketch plan 175
skill 28
slack resource 3
SLAM 80
small claim 148
smooth application 147
smooth mutation 147
solution 67
solution finding 14, 66
specialization 27, 31
specification 14
  of input data 60
  of model system 60
  of output data 63
  of structural model 58
split operator 42
start-up time 62, 108, 147, 194
statistical analyzer 78
statistical test
\( \chi^2 \)-test 78, 197
ANOVA test 78
F-test 65, 78
Mann-Whitney test 78
Pierson’s test 197
t-test 65, 78, 114
stats unit 87
stoch unit 87
storage unit 84
storm 168
strategic target 20
strategic thrust 20
structural model 56
structural validation 64, 107, 146, 193
structure predicate 36
Structured Analysis 19
substantive rational 18
subtask predicate 36
super class 27
systelological problem 11
system 53

T
t distribution 64
t-test 78, 114
task 27, 33
compound 34
decomposition 37
delay 38
formal definition 38
goal 38
hierarchy 37
input 38
operation 38
output 38
simple 34
uncertainty 22
task modeller 74, 85
task predicate 39
task structure 34
formal definition 39
graphical representation 35
technology push 1
tools 71
top line 20
town development plan 175, 180
transition 34
condition for 34
copy context operator 41
formal definition 43
join operator 42
split operator 42
translator 75
treatment 107
treatment 194
treatment 62, 107, 146, 194
trial-and-error 65, 66
trigger 60
trigger predicate 39
Turing test 65

U
uncertainty 17, 22
unit 81
urgent file 154

V
validation 19
replicative 64, 108, 146, 194
structural 64, 107, 146, 193
variance 78
verification 20, 107, 146, 193
of entity model 59
of functional model 60
of model system 61
of structural model 59
version handling 71, 85
views of entity model 76
visualization 20

W
window 83
windows unit 83
work place 23

Y
yard-stick 65, 113
Samenvatting

Inleiding

Informatietechnologie wordt reeds enkele decennia toegepast in organisaties in zeer uiteenlopende branches. In eerste instantie werd deze technologie ingepast in bestaande processen en in de bestaande besturing van de organisatie. Steeds vaker, echter, wordt informatietechnologie gezien als een technologie die het mogelijk maakt om een organisatie op een andere manier in te richten of te besturen. De nadruk verschuift hierbij van technische vraagstukken naar organisatorische; de vraag hoe de doelstellingen van een organisatie kunnen worden verwezenlijkt staat voorop, waarbij informatietechnologie één van de middelen is die voor de realisatie van die doelstellingen ingezet kan worden.

De keuze om de aandacht te verschuiven van het inpassing van informatietechnologie in de bestaande werkwijze naar het herinrichting van de bedrijfsprocessen zelf is niet geheel een keuze die door vrije wil wordt ingegeven, maar wordt eerder gestimuleerd door veranderingen in de omgeving waarbinnen organisaties uit zowel de publieke als de private sector moeten opereren. Natuurlijke grenzen zoals tijd en afstand zijn in betekenis afgenomen, hetgeen geresulteerd heeft in een mondiale afzetmarkt waar de concurrentie heviger en sneller is dan voorheen. Deze ontwikkeling doet zich onder meer voor binnen de dienstensector en bij het openbaar bestuur, waar het primaire proces van organisaties veelal informatieverwerking betreft. Binnen dergelijke informatie-intensieve organisaties ontstaat de noodzaak om sneller in te spelen op ontwikkelingen in de markt — een behoefte aan adaptiviteit. Hierbij zijn de kwaliteit en tijdigheid van dienstverlening in deze organisaties belangrijke aandachtspunten geworden.

Het inspelen op veranderingen in de omgeving is echter niet eenvoudig. Het effect van veranderingen in de bedrijfsprocessen en gebruikte technologieën op de effectiviteit van de bedrijfsvoering is moeilijk te bepalen. Hierbij worden deze moeilijkheden veelal versterkt door de aanzienlijke omvang en complexiteit van een organisatie.
Samenvatting

De centrale vraagstelling binnen het onderzoek is de vraag hoe diegenen die verantwoordelijk zijn voor het bedenken, ontwerpen, realiseren en evalueren van de juiste veranderingen in informatie-intensieve organisaties adequaat kunnen worden ondersteund.

Om de complexiteit van organisaties op te vangen maakt men gebruik van vereenvoudigde afbeeldingen ofwel modellen. De eerste deelvraag die aan het begin van het onderzoek is geformuleerd is dan ook:

Welke modelleer technieken kunnen, wanneer zij worden gebruikt voor het modelleren van informatie-intensieve organisaties, het inzicht in problemen en kansen van een organisatie vergroten voor diegenen die verantwoordelijk zijn voor het ontwerpen en invoeren van veranderingen?

In aanvulling op de modelleer technieken zelf moeten de gebruikers ervan ook worden ondersteund bij het toepassen van deze technieken om te komen tot effectievere veranderingsvoorstellen. Hierbij kan men denken aan ondersteuning voor het creatieve ontwerpproces, voor het tot stand brengen van een goede communicatie tussen alle betrokkenen en voor het evalueren van de effecten van voorgestelde veranderingen. Met oog op dergelijke ondersteuning is de tweede deelvraag geformuleerd:

Welke faciliteiten moeten onderdeel uitmaken van een omgeving die erop gericht is mensen te ondersteunen bij het uitvoeren van veranderingsprocessen in informatie-intensieve organisaties?

Probleemoplossen met dynamische modellen

Organisatieverandering kan gezien worden als een proces van probleemoplossen. In de bestaande situatie zijn knelpunten aan te wijzen, of er is een idee over hoe een verbetering kan worden gerealiseerd. Kortom, men is ontevreden over het huidige functioneren van de organisatie en wil daar verandering in brengen.

Uitgaande van de bestaande literatuur over probleemoplossen wordt een werkwijze voorgesteld die begint met het beschrijven en valideren van een zogenaamd kenmodel: een model dat de huidige situatie beschrijft. Dit model wordt gebruikt om aan te geven waar zich een probleem bevindt en welke oorzaken men voor de geïdentificeerde problemen kan aangeven.

Door het begrip van de bestaande situatie is het mogelijk om verbeteringsvoorstellen te formuleren. Deze worden beschreven in zogenaamde maakmodellen. Door een evaluatie van de verschillende opties ten opzichte van de bestaande situatie uit te voeren kan men komen tot een keuze, en kan de gekozen verandering worden gerealiseerd en ingevoerd.

248
Uitgaande van deze visie op veranderingsprocessen in organisaties kan van effectieve ondersteunen alleen dan sprake zijn wanneer alle stappen uit het proces van probleempoplossen worden ondersteund. Bestaande methodieken voor (informatie)systeemontwerp bieden technieken en gereedschappen die bepaalde stappen uit dit proces ondersteunen. Op twee gebieden is echter verbetering mogelijk:

1. Doordat de participatie van betrokkenen uit de organisatie cruciaal is voor het welslagen van een veranderingsproces, is het belangrijk dat de modellen die gedurende het veranderingsproces worden gebruikt gemakkelijk toegankelijk zijn — ook voor mensen die hierin geen specifieke training hebben gehad.


Het evalueren van het effect van veranderingen in een organisatie kan hierbij op drie nivo's plaatsvinden: het macro nivo — de relatie van de organisatie met zijn omgeving; het meso nivo — de wijze waarop de interne bedrijfssprocessen zijn georganiseerd; en het micro nivo — de werkplek van een individu (ook wel informatiewerker genoemd).

Om te komen tot een verbeterde ondersteuning ten aanzien van de twee bovengenoemde punten wordt het gebruik van dynamisch modellen voorgesteld. Dergelijke modellen bieden niet alleen inzicht in de statische aspecten van bedrijfssprocessen, maar kunnen deze processen ook nabootsen of simuleren. Een visuele terugkoppeling van dergelijke simulaties wordt hierbij als essentieel onderdeel van de ondersteuning gezien.

Het simuleren van de bedrijfssprocessen van een organisatie in hun huidige vorm en in een voorgestelde toekomstige situatie, biedt de mogelijkheid deze toekomstige situatie te evalueren alvorens deze daadwerkelijk is gerealiseerd. Het inzicht dat langs deze weg wordt verkregen over de verschillende veranderingsalternatieven maakt het mogelijk een goed afgewogen keuze te maken. Doordat de consequenties op het nivo van de bedrijfssprocessen en op het nivo van de werkplek voor alle betrokkenen verduidelijkt worden, vergemakkelijkt het gebruik van dergelijke dynamische modellen tevens de implementatie van het gekozen alternatief.

De modelleerwijze die in dit proefschrift wordt voorgesteld gaat uit van drie modelleerconcepten: een informatiewerker of actor, een informatieobject of item, en een informatieverwerking of taak. Een taak representeert een bepaalde behandeling van een item door een actor, of een samenstelling van meerdere subtaken (eventueel uitgevoerd door verschillende actoren) welke in een bepaalde volgorde op een item
Samenvatting

worden uitgevoerd. Om de volgorde van de verschillende subtaken, ofwel de onderlinge coördinatie van taakuitvoering door de betrokken actoren, tot uitdrukking te brengen, wordt een zogenoemde taakstructuur gebruikt.

In de modelleerwijze is naast de statische afbeelding van bedrijfsprocessen in de vorm van taakstructuren ook rekening gehouden met dynamische aspecten zoals tijdsduur en prioriteit van taken, en de werkwijze van de betrokken actoren. Zodoende biedt de omgeving welke het dynamisch modelleren ondersteunt de mogelijkheid om, uitgaande van een beschreven model, een simulatie van de bedrijfsprocessen uit te voeren — dit zonder een aanvullende vertaalslag te hoeven maken.

Naast de modelleertechnieken zelf wordt ook een beschrijving gegeven van de verschillende activiteiten die onderdeel uitmaken van het proces van organisatieverandering. Deze beschrijving dient als basis voor het bepalen van de functionele specificaties van de gereedschappen die de aanpak ondersteunen. Als onderdeel van het onderzoek is vervolgens een prototype omgeving gebouwd waarin deze ondersteuning gestalte heeft gekregen.

Toetsing van de bruikbaarheid van dynamisch modelleren

Om de bruikbaarheid van de voorgestelde werkwijze en in het bijzonder de dynamische modelleerwijze te toetsen zijn deze toegepast in diverse praktijk situaties. In de toetsing zijn zowel organisaties uit de publieke sector (centraal en decentraal) als uit de private sector betrokken. In elk van de studies wordt getoetst of het gebruik van dynamisch modelleren zoals dat in dit proefschrift is beschreven de betrokkenen in staat stelt om effectieve veranderingsvoorstellen voor hun organisatie te bedenken en te evalueren.

1. Centrale overheid

De eerste casus betreft de Hoofdafdeling Pensioenen en Wachtgelden (P&W) binnen het Ministerie van Binnenlandse Zaken. Deze afdeling is verantwoordelijk voor het uitkeren van wachtgelden en pensioenen aan voormalig overheidspersoneel. Als gevolg van de taakafstoting en inkrimping van de centrale overheid heeft deze afdeling eind jaren tachtig te kampen gehad met een aanzienlijke uitbreiding van het aantal cliënten. In 1988 leidde dit tot een groot project dat erop gericht was een nieuw computersysteem voor de afdeling te ontwikkelen waarmee een productiviteitswinst van 25% behaald moest kunnen worden. Deze verbetering zou voldoende zijn om de problemen binnen de afdeling op te kunnen lossen.

Met de ontwikkelde modelleertechnieken is een model opgesteld van de huidige bedrijfsprocessen binnen P&W. Dit kenmodel is gebruikt om vast te stellen wat precies de knelpunten waren en welke oorzaken hiervoor konden worden aangewezien. Tijdens het modelleren werd veel aandacht besteed aan de communicatie met het personeel van de afdeling opdat ook zij inzicht kregen in de oorzaken die ten grondslag lagen aan hun problemen.
Het kenmodel is in eerste instantie gebruikt om het effect van het voorgestelde computersysteem te evalueren. Zoals uit deze evaluatie bleek kon dit systeem op zich de problemen van de afdeling niet oplossen. De nadruk lag bij het nieuwe systeem op het vervroegen van de binnenkomst van verklaringen inzake inkomsten — een formulier waarvan zo’n 4000 exemplaren per maand moesten worden verwerkt. Zoals uit het kenmodel bleek was echter niet de late beschikbaarheid van deze formulieren maar de arbeidsintensiviteit van het afhandelingsproces de belangrijkste oorzaak voor de geconstateerde problemen. Met dit inzicht werd een alternatieve oplossing voorgesteld waarbij de formulieren die geen verandering toonden ten opzichte van de vorige maand vroegtijdig uit het proces werden gehaald. Met het maakmodel dat deze verandering beschreef is de haalbaarheid en de effectiviteit van deze oplossing aangetoond.

2. Private sector

De tweede casus betreft een afdeling binnen de Goudse Verzekeringen BV. De betreffende afdeling is verantwoordelijk voor het accepteren en muteren van nieuwe aanvragen voor brandverzekeringen en voor het uittrekken van schadeclaims voor deze verzekeringen. Doordat de Goudse Verzekeringen een intermediair maatschappij is, wordt zij geconfronteerd met de wens van assurantie tussenpersonen voor een snelle en flexibele dienstverlening. In dit kader was men geïnteresseerd in de mogelijkheden voor het verbeteren van de dienstverlening uitgaande van de bestaande middelen.

Naar aanleiding van een eerder onderzoek naar de toepasbaarheid van kennisystemen binnen de afdeling Brandverzekeringen was het duidelijk dat niet de ondersteuning van specifieke taken in het proces, maar eerder de coördinatie van de verschillende bij het proces betrokken mensen en afdelingen een bron voor verbetering was. In het geconstrueerde kenmodel van de afdeling is dan ook primair gekeken naar deze coördinatie.

Naar aanleiding van het inzicht dat met het kenmodel verkregen werd in de huidige situatie waren de betrokkenen in staat om een breed scala aan veranderingsoverstallen te genereren. Het effect van de voorgestelde veranderingen kon met behulp van maakmodellen worden bepaald. Zoals uit deze evaluatie bleek konden zeer eenvoudige wijzigingen in het proces leiden tot aanzienlijke verbeteringen in de snelheid van dienstverlening. Naast deze veelal kleine aanpassingen van de huidige situatie zijn ook een aantal ingrijpende alternatieven bekeken. Deze dienen voor het management als oriëntatie op mogelijke verbeteringen op de middellange termijn. Met de verkregen resultaten is men nu in staat om nieuwe ontwikkelingen, zoals het Assurantie Data Net, op hun waarde te schatten.
3. Decentrale overheid

De derde en laatste casus betreft de afdeling Bouw- en Woningtoezicht binnen de gemeente Rheden. De betreffende gemeente is betrokken bij een landelijk project voor de invoering van zogenoemde civic service centers — plaatsen waar burgers terecht kunnen voor alle gemeentdiensten en op termijn wellicht ook voor diensten die op provinciaal of landelijk nivo zijn georganiseerd.

Door Bouw- en Woningtoezicht als voorbeeld te stellen wilde de gemeente nagaan of het mogelijk is de dienstverlening aan burgers te verbeteren. De directe aanleiding voor het project was het nieuwe bouwbesluit dat door de overheid ingevoerd zou gaan worden. Hoewel dit bouwbesluit een ruimere termijn stelt voor het afhandelen van bouwaanvragen, wordt een voorwaarde gesteld dat alle aanvragen die niet binnen de termijn zijn afgehandeld verondersteld worden verleend te zijn. In de huidige situatie zou dit betekenen dat zo’n 40% van de aanvragen zonder getoetst te zijn zullen worden geaccepteerd.

Zoals dit in de voorgaande studies ook is gedaan begon de analyse met een gedetailleerde beschrijving van het huidige afhandelingsproces voor bouwaanvragen. Hierbij lag, mede gelet op de ervaringen bij de tweede casus, de nadruk op de coördinatie van de betrokken mensen en afdelingen binnen dit proces. Bij het ontwerpen en evalueren van veranderingsalternatieven bleek weer dat een aantal kleine wijzigingen in het proces aanzienlijke verbeteringen te zien gaven in de gemiddelde behandelingstermijn. Naast wijzigingen in het proces is ook gekeken naar verschillende definities van behandelingstermijn daar andere definities meer ruimte bieden voor een goede dienstverlening zonder dat de wettelijk vastgestelde restricties in gevaar komen. Het resultaat is dat de afdeling binnen het nieuwe bouwbesluit met behulp van een aantal eenvoudige wijzigingen een kwalitatief hoogstaande dienstverlening kan waarmaken.

Conclusies

Het onderzoek laat zien dat een goede ondersteuning mogelijk is voor mensen die verantwoordelijk zijn voor het bedenken en implementeren van veranderingen in organisaties. Het is mogelijk gebleken begrijpelijke en herkenbare modellen te construeren die de betrokkenen een helder inzicht geven in de problemen binnen een bestaande organisatie, alsmede in de oorzaken die aan deze problemen ten grondslag liggen. Het gebruik van dynamische modellen die niet alleen inzicht geven in de statische aspecten van bedrijfssprocessen maar deze tevens in de tijd na kunnen booten geeft hieraan een grote bijdrage.

De haalbaarheid van dynamisch modelleren is aangetoond door een ondersteuningsomgeving te realiseren waarin deze modelleertechniek wordt ondersteund.
De bruikbaarheid van zowel de aanpak als de hulpmiddelen is getoetst aan de hand van drie praktijksituaties. In alle studies blijkt dat de betrokkenen in staat zijn geweest om effectieve veranderingen te ontwerpen. Doordat het effect van deze veranderingen inzichtelijk werd gemaakt voor zowel de organisatie als geheel als voor elk individu dat in de processen betrokken is, draagt de aanpak ook bij aan een goede en snelle implementatie van deze veranderingen. Op basis van de verkregen resultaten concluderen wij:

De dynamisch modelleeraanpak, die gekenmerkt wordt door een gedetailleerde beschrijving van de huidige situatie en door het gebruik van simulatie om het effect van veranderingen te kunnen analyseren, geeft adequate ondersteuning voor veranderingsprocessen in informatie-intensieve organisaties.

In de onderzochte praktijksituaties blijken (kleine) organisatorische wijzigingen vaak een zeer aanzienlijk verbetering in de prestaties van een organisatie te kunnen bewerkstelligen. Bij het ontwikkelen van geautomatiseerde informatiesystemen gaat men echter veelal uit van de bestaande organisatie, en past men informatietechnologie in in de bestaande bedrijfsprocessen en in de bestaande sturing. De ondervindingen van dit onderzoek pleiten ervoor binnen informatiesysteem-ontwikkeling meer aandacht te besteden aan het herinrichten van bedrijfsprocessen waarbij informatietechnologie kan worden ingezet om dergelijke wijzigingen in organisaties te implementeren:

Binnen de praktijk van het ontwikkelen van informatiesystemen zou men meer aandacht moeten schenken aan mogelijke verbeteringen die verkregen kunnen worden door veranderingen aan te brengen in de bedrijfsprocessen en de sturing hiervan. Als informatietechnologie wordt toegepast in organisaties zonder dergelijke wijzigingen door te voeren kan het rendement van de benodigde investeringen verre van optimaal zijn.