Designing Organizational Coordination
Designing Organizational Coordination

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof.ir. K.F. Wakker,
in het openbaar te verdedigen ten overstaan van een commissie
door het College van Dekanen daartoe aangewezen,
op donderdag 25 april 1996 te 10:30 uur
door

Daniël Thaddeus Theodorus VAN EIJCK

informatica ingenieur

geboren te Arnhem
Dit proefschrift is goedgekeurd door de promotors:

Prof.dr. H.G. Sol

Samenstelling promotiecommissie:

Rector Magnificus, voorzitter
Prof.dr. H.G. Sol, Technische Universiteit Delft, promotor
Prof.dr. J.H.T.H. Andriessen, Technische Universiteit Delft
Prof.dr. J.C. Arnbak, Technische Universiteit Delft
Prof.dr. A. Bosman, Rijksuniversiteit Groningen
Prof.dr.ir. J.L.G. Dietz, Technische Universiteit Delft
Prof.dr. A.K.F. Faludi, Universiteit van Amsterdam
Prof.dr.ir. R.E.C.M. van der Heijden, Technische Universiteit Delft
To Marion
Colophon

Published and distributed by:

D.T.T. van Eijck
Dirklangendwarsstraat 29
2611 HZ Delft
The Netherlands
☎ +31.15.215.8921

Delft University of Technology
School of Systems Engineering, Policy Analysis and Management
P.O. Box 5015
2600 GA Delft
The Netherlands
☎ +31.15.278.2318, Fax: +31.15.278.3429

Cover Photo: © 1996 M.C. Escher / Cordon Art - Baarn - Holland. All rights reserved. Cover design by G.M.A. van der Harst.

CIP-DATA ROYAL LIBRARY, THE HAGUE

Eijck, Daniel T.T. van,
Designing Organizational Coordination
Doctoral Dissertation, Delft University of Technology
With illustrations, index, literature list and summary in Dutch
ISBN 90-5638-007-9 / NUGI 684
Key words: coordination / organizational management / process management.

© Copyright 1996 D.T.T. van Eijck.

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

'With a single stroke of analytic thought he split the whole world into parts
of his own choosing, split the parts and split the fragments of the parts,
finer and finer until he had reduced it to what he wanted it to be.'
- Robert Pirsig

Preface .............................................................................................................. XI

1. Organizational Coordination: A Design Approach
   1.1 Recent organizational trends ...................................................... 1
   1.2 Organizational coordination ..................................................... 4
   1.3 Communication in business processes ................................. 12
   1.4 Research scope and objectives ............................................ 15
   1.5 Research philosophy and research approach .................... 19
   1.6 Research outline ................................................................. 25

2. Organizational Coordination in Practice
   2.1 Choosing a research environment .......................................... 27
   2.2 Case 1: A large shipping company .................................... 28
   2.3 Case 2: A major Dutch bank .............................................. 32
   2.4 Delineation of the problem area ......................................... 36
   2.5 Case 3: Dutch Telecom service engineers ....................... 38
   2.6 Case 4: A large insurance company .................................. 42
   2.7 Extending the conceptual groundwork .............................. 52

3. A Conceptual Framework for Coordination
   3.1 Introduction ................................................................. 55
   3.2 Business processes .......................................................... 55
   3.3 Coordination and coordination problems ......................... 61

vii
Designing Organizational Coordination

3.4 Measuring organizational performance ........................................... 67
3.5 Coordination and technology......................................................... 70
3.6 Towards the development of a design system .................................... 76

4. A METHODOLOGY FOR COORDINATION DESIGN

4.1 Introduction ....................................................................................... 77
4.2 Way of thinking ................................................................................. 78
4.3 Working method ................................................................................ 83
4.4 Modeling approach .......................................................................... 90
4.5 Management approach .................................................................... 111

5. SUPPORT TOOLS FOR COORDINATION DESIGN

5.1 Introduction ....................................................................................... 115
5.2 Conceptualization support ............................................................... 116
5.3 Specification support ....................................................................... 119
5.4 Conclusions ...................................................................................... 138

6. APPLYING THE DESIGN SYSTEM: TWO CASES REVISITED

6.1 Introduction ....................................................................................... 143
6.2 Case 1: Dutch Telecom service engineers (revisited) ......................... 144
6.3 Case 2: Insuraco (revisited) ............................................................. 163

7. APPLYING THE DESIGN SYSTEM: THREE NEW CASES

7.1 Introduction ....................................................................................... 185
7.2 Case 3: Dutch Life insurance acceptance ........................................ 186
7.3 Case 4: Dutch Life ‘policy surrender’ .............................................. 216
7.4 Case 5: Dutch General Mail Logistics ............................................ 237

8. EPILOGUE

8.1 Introduction ....................................................................................... 267
8.2 The research environment ............................................................... 268
8.3 The design system ........................................................................... 275
8.4 Research philosophy and research approach .................................... 282
8.5 Further research ............................................................................... 284

REFERENCES............................................................................................ 287
Contents

AUTHOR INDEX .................................................................................................................. 301

SUBJECT INDEX .................................................................................................................. 305

SAMENVATTING .................................................................................................................. 309

CURRICULUM VITAE .......................................................................................................... 321
Designing Organizational Coordination
Preface

‘What does it matter? Science has achieved wonderful things of course, but I'd far rather be happy than right any day.’
- Douglas Adams

Modern organizations are constantly facing the great challenge of improving the performance of their business. Assisting organizations in this task is an even greater challenge. In this dissertation we have taken up this challenge in a scientific endeavor. We developed a design system that assists organizations in improving the performance of their business processes. A design system in which better coordination stands central. A design system inspired and validated by current organizational practice.

Challenges are hardly ever taken up alone. First of all there is the need of a challenger. I am grateful to Henk Sol for taking up a role that can be best explained by the German term ‘Doktorvater’. I admire his everlasting energy and sharpness, especially in the period of writing this dissertation. I also want to thank Andreas Faludi for stimulating me to write a dissertation in the first place, for our discussions and for proofing the entire manuscript with unmatched enthusiasm and accuracy.

The case studies that contributed so greatly to this research required the cooperation of various people in organizations. Many people in several organizations have contributed to this research, but I would like to thank Anton Bosman, John Heller, Wim den Ouden, Eugène Philips, Roel van Rijnbach and Jan van Roosendaal in particular for their hospitality and cooperation.
Designing Organizational Coordination

An important aspect of finishing a dissertation is a supportive work environment. In the four years I worked at SEPA, the school grew from fifteen people to well over a hundred. Despite this growth, the environment has never lost its spontaneity and atmosphere of constructive criticism. I owe a lot to all my colleagues who made SEPA such a wonderful place to work. In particular, I owe Martijn Babeliowsky, Jeroen van Meel and Thijs Straatman for their valuable ideas and comments, for their companionship at lunch and for their ability to drink beer at conference hotel bars. Special thanks go to Gert-Jan de Vreede for being an unforgettable roommate, for taking his share in Far Away and Beyond Inc., and for being unable to score in our basketball matches.

Some students are capable of improving the research assistant’s quality of life. I owe Michiel Bouwman for investing so much time and energy in the Dutch General cases. I thank him and Gerard Kuijlaars for understanding the challenge and for building the prototype of the DOC/S tool.

Prefaces usually end with gratitude for the ‘home-front’. I was very lucky to have three of them. First of all, I thank my parents and my brother Ramon for their love and constructive support both metaphorically, in terms of my scientific work, and literally in terms of refurbishing my house. I thank the Faludis for their help and support in the past four years and for giving me that ‘home-coming feeling’ at Nieuwe Plantage 3. Finally, I thank Marion for her love and enthusiasm in life and for making this dissertation worth the effort.

Daniël T.T. van Eijck
Delft, March 1996
1. Organizational coordination: a design approach

'Indeed, research is often guided by the desire to improve practice'
- Andreas Faludi

1.1 Recent organizational trends

Over the last ten years, organizations in the service industry have been gaining in importance. The growth of these organizations has led to the establishment of large corporations in, for example, the financial business (banks, insurance companies and investment trusts), the transport business (airlines, shipping lines, road transport organizations and transfer centers), the consulting business (management consultants, legal- and financial advisors) and more recently the media and entertainment business (newspapers, television networks, home use software developers). Common to these industries is that information plays an important role in their core business processes. For their business processes to function properly, they have to rely on information being available at the right place and at the right time.

Organizations tend to merge and cooperate in order to achieve a strategic advantage in a complex and highly competitive market. Currently, as markets become international, this growth transcends national boundaries. As a result of growth, internationalization and other market developments, organizations have to operate in a distributed manner, in order to cope with problems of scale. Many organizations try to divide their processes into business units or divisions. Because information is of such great importance to the business processes, distribution generates the need for new information and communication technologies that support these business processes effec-
Designing Organizational Coordination

tively. On the other hand, tendencies of growth and distribution are partly enabled by information and communication technology like telephony, fax, electronic mail, local and wide area networks, Electronic Data Interchange (EDI), Internet, the World Wide Web and videoconferencing (see e.g. Keen, 1988). It is only now that organizations in the service industry are discovering these technologies.

Sol (1992, p. 9) identifies these developments as a change towards a 'global, networked society' in which 'organizations become like networks of small, independent subsystems, supporting the tasks of staff in their specific working places'. The turbulent, uncertain and highly competitive environment forces organizations to be more effective. The effectiveness of organizations is greatly determined by the way they solve coordination problems (Sol 1982, 1992). This implies the need for support in the development of coordinated business processes. From the above, we derive the following contemporary features of large organizations in the service industry:

- Business processes are information intensive and interdependent.
- For several reasons, business processes show dispersion.
- To be effective, these processes have to be coordinated.
- Information and communication technology may help with this.

We use an example to illustrate the problems that this implies. The example concerns an existing organization of which we have anonymized the name.

An example: Accurate Software Creators Inc.

Accurate Software Creators Incorporated (ASCI) produces and supplies financial and business-specific computer software for small to medium-scale organizations in Europe. The company was founded in the beginning of the eighties and has grown to become a market leader in the Netherlands for several different kinds of software. It has experienced several coordination problems in its distributed and information intensive business processes:
Organizational coordination: a design approach

1. Because of the complexity of its products, ASCI is forced to operate a service organization, preferably close to the client. This leads to the establishment of a central phone help-desk and several regional service offices. Problems that cannot be handled by the central phone staff have to be taken over by employees in the regional offices, who may be able to visit the client.

**Problem:** How to provide all employees (regional as well as central) with the right customer, problem and product information, without loosing time and without letting service drop to an unsatisfactory level.

2. Because of the high wages in Western Europe, ASCI is forced to explore the possibilities of software development in foreign countries like India and Russia. However, testing the software has to take place in the Netherlands, sometimes even with pilot clients. Project management should also be kept in Dutch hands, to keep up the service level.

**Problem:** How can project teams of Russian or Indian programmers be employed and managed from a distance, keeping performance (i.e. test results) at a maximum and traveling costs to a minimum.

3. For its standard financial software, ASCI has a very large installed base in the Netherlands and Belgium. Due to changing regulations and tax laws, some software modules have to be changed annually or even more frequently. Furthermore, Belgium and the Netherlands have different tax systems, and clients have different software modules.

**Problem:** How to automatically provide a large volume of clients with the right software upgrades, without having to invest in a costly, high-speed, high-volume floppy disk duplication system.

Poor design of the business processes and the supporting information technology can be disastrous for the corresponding parts of the organization. Organization designers who have to solve the above problems need techniques to create well coordinated business processes, effectively supported by information and communication technology.
Designing Organizational Coordination

The main goal of this research is, therefore, the development of methodological and technological support for the design of coordinated business processes. We discuss this research objective in more detail in section 1.4. However, to determine our focus, we first conduct an inquiry into the realm of organizational coordination and communication.

1.2 Organizational coordination

In this section we present a short chronological overview of the literature on organizational coordination. Based on this overview, we give a framework for coordination design and a definition of organizational coordination that is central to this book.

1.2.1 Literature on organizational coordination

Organizational coordination is a concept that is widely described in the literature on organization design. As a starting point, we use Luther Gulick's encouragement to '... those who are concerned scientifically with the phenomena of getting things done through co-operative human endeavor ...' (Gulick and Urwick, 1937). The 'Papers on the Science of Administration', edited by the latter authors, are referred to in many publications since, and seem to have had a broad impact on the literature on administration and management.

According to Gulick, 'the theory of organization, ..., has to do with the structure of co-ordination imposed upon the work-division units of an enterprise.' (Gulick, 1936). Urwick (1933) refers to Mooney and Reiley (1931) who state that coordination is equal to organization. The two main principles of organization put forward by Gulick and Urwick (1937) and many others after them are (see also Bosman 1995):

- Work has to be divided as the resources of one man are limited.
- The division of work into tasks gives rise to the adjustment, the coordination of these tasks.
Organizational coordination: a design approach

We regard these as the two axioms of organizational activity.

Simon (1945) speaks of coordination as 'the process of informing each other as to the planned behavior of others.' Furthermore he mentions organizational influence mechanisms that are mainly meant to bring about coordination in the activities of the organization's members. These mechanisms are: the division of work, the establishment of standards, the downward transmission of authority and influence, providing channels of communication and the training and indoctrination of the members of the organization. From this, we extract the following organizational devices that can be used to establish coordination:

- Standards.
- Authority.
- Communication.
- Training.

We will argue later that communication is a key to the coordination of even the most basic organizational activities.

Thompson (1967) ties coordination to interdependence. He distinguishes between pooled interdependence, where part-systems contribute to, and depend on, the whole without the need for mutual interaction, sequential interdependence, where the output of one part system depends on the input of another, and reciprocal interdependence, where part-systems depend on each other's input in order to produce output. Pooled interdependence is coordinated by rules and standards, sequential interdependence is coordinated by plans, and reciprocal interdependence can be coordinated by mutual adjustment. Thompson clearly finds a systems approach to the analysis of organizations useful. We agree and assert that an organization can be seen as a system.

Galbraith (1973, 1977) sees coordination as a main issue in organization design. In fact he sees organization design as the search for coherence between the goals or purposes for which an organization exists, the people that do the work and the patterns of division of labor.
Designing Organizational Coordination

and interunit coordination. Galbraith brings in the concept of organizational purposes and goals, from which we extend our view on organizations as systems that act to attain a certain goal: organizations are purposeful, even purposive systems (see also Ackoff and Emery, 1972).

Kieser and Kubicek (1977, 1983) distinguish between four different ways to achieve coordination: by means of personal intervention, self adjustment, programs (pre-established work processes) and plans. They also identify feedback coordination and future coordination (Vorauskoordination). They define feedback coordination as coordination resulting from errors or disturbances and future coordination as coordination to prevent future errors.

Mintzberg (1979) gives an excellent literature review and concludes that coordination is the main subject in works that discuss problems of organization design. Mintzberg (1983) states that 'every organized human activity gives rise to two fundamental and opposing requirements: the division of labor into various tasks to be performed, and the coordination of these tasks to accomplish the activity', thereby rephrasing Gulick (1933). He asserts that 'the structure of an organization can be defined simply as the sum total of the ways in which its labor is divided into distinct tasks and in which coordination is achieved among these tasks', thereby linking coordination to organizational structure.

Mintzberg describes three mechanisms concerned with coordination, communication and control. These mechanisms are referred to as coordinating mechanisms, and they include mutual adjustment, direct supervision and standardization. He mentions three ways of achieving coordination by means of standardization: standardization of skills, of work processes and of outputs. Mutual adjustment achieves coordination by means of a process of informal communication. Direct supervision achieves coordination by having one person take responsibility for the work of others, issuing instructions to them and monitoring their actions. Standardization achieves coordination by programming or specifying (in advance) the required skills and knowledge, the content or the results (outputs) of work. As regards the relation between com-
Organizational coordination: a design approach

plexity and coordinating mechanisms, Mintzberg argues: ‘As organizational work becomes more complicated, the favored means of coordination seems to shift from mutual adjustment to direct supervision to standardization, preferably of work processes, otherwise of outputs or else of skills, finally reverting back to mutual adjustment.’

Kickert (1979) gives a system-theoretical definition of structure as the combination of decomposition and coordination (i.e. division of labor and coordination). The former is defined as: ‘to form relevant part-systems of the original system’ and the latter as: ‘to form relevant relationships between those part-systems’. However, according to Kickert there is more to coordination than forming relevant relationships and therefore he defines coordination as the control of a system of part-systems.

Holt (1988) also mentions a tight relationship between coordination and organization. He states that coordination is ‘an elusive concept’, and particularly frustrating when it is lacking. According to Holt (1988), coordination is a ‘kind of dynamic glue that binds tasks (that result from the division of labor, -DvE) together into larger, meaningful wholes (business processes, -DvE)’. It serves to establish relationships between tasks and their products. Clearly, Holt (1988) sees coordination as something dynamic and process-oriented.

Veryard (1994) sees information and information processing as very important to coordination in organizations. He mentions information systems as important means to achieve coordination between and within organizations.

Malone and Crowston (1994 p. 90) state that ‘coordination ... can occur in many kinds of systems: human, computational, biological and others’. They define coordination as managing dependencies between activities. They mention actors and (interdependent) activities as key concepts and take a process-oriented approach. We conclude that organizational activities and their performers (actors) are very important concepts for the analysis and design of coordinated business processes. We elaborate on this in section 1.2.3.
Designing Organizational Coordination

In this short review, we observe that coordination is the answer to problems arising from the dispersal of organizational activities over organizational units. However, we did not focus on how this division of activities can be achieved. Organizational units can be grouped according to several criteria. We elaborate on this in the next subsection.

1.2.2 Coordination and dispersed organizational activity

Gulick and Urwick (1937) state four different ways in which components of organizations can be grouped: (1) common purpose or contribution to the larger organization, (2) common processes, (3) a particular clientele, or (4) a particular geographic area. Thompson (1967) argues that the problem with this scheme for organizations lies in the fact that components of complex organizations are not unidirectional. An organization, for example, the components of which are grouped according to geographic location, will have problems to coordinate common processes. Thompson states that organizations will group their components in a way that minimizes coordination costs. The costs of coordination by mutual adjustment are higher than the costs of coordination by plan, which are in turn higher than the costs of coordination by standardization (Thompson, 1967).

Mintzberg's (1983) definition of decentralization is strongly tied to coordination. He uses the concept of decision making power to define decentralization: '...to the extent that the power is dispersed among (many) people, we shall call the structure decentralized.' However, centralization and decentralization are no absolute values but should be treated as the two ends of a continuum whereby centralization '...is the tightest means of coordinating decision making in an organization.' Furthermore, Mintzberg distinguishes between two kinds of decentralization: Vertical decentralization is defined as the dispersal of formal power down the chain of line authority. Horizontal decentralization is the extent to which non-managers control decision processes. We follow Mintzberg (1983) by not using the term decentralization in the physical sense. The physical dispersal of services or parts of an organization is called distribution. A distributed organization is
Organizational coordination: a design approach

therefore specified as an organization of which the parts show physical dispersion.

Intuitively, it seems logical to say that distributed organizations have less possibilities to coordinate interdependent processes than other organizations. For example, the mechanism of direct supervision seems less applicable to geographically dispersed organizations. Mutual adjustment between members of distinct parts of the organization will, for these organizations, involve at least some kind of communication technology. We assert here that distributed organizations are very useful places to examine coordination because, in order to be effective, coordination has to be made more explicit and more formal in these organizations. Further elaboration on this issue is given in section 1.5.

1.2.3 A framework and definition

As a starting point for our research, we now present a framework for the design of organizational coordination. The framework should enable us to achieve organizational performance improvement by analyzing and designing organizational processes and structures. Bots and Sol (1988) proposes three perspectives from which organization design and improving organizational performance can be considered:

- The macro-perspective focuses on cooperation between organizations, by searching for possibilities that improve interorganizational performance.
- The meso-perspective focuses on the coordination of processes in an organizational setting, by searching for possibilities that improve organizational performance.
- The micro-perspective focuses on the primary tasks carried out by individuals within organizations, by searching for possibilities that improve individual performance.

From the literature review, we conclude that organization design traditionally focuses on elements of organizational structure. The establishment and interrelation of organizational units, departments and the people that perform tasks have been more important than the ac-
Designing Organizational Coordination

tual design of tasks and processes themselves. Organization charts, for example, have been far more popular as an instrument of communication, than process descriptions, not only with organization scientists, but also with managers and consultants. Recent literature on the topic of business (re)engineering, shows a shift of focus towards the design of organizational processes, as opposed to organizational structure (Hammer, 1990, Davenport and Short, 1990, Drucker, 1991, Dur, 1992, Hammer and Champy, 1993, Meel, 1994). The design of organizational coordination is (or at least should be) a combination of both structure and process elements (see also Galbraith, 1977). However, in our opinion, business process design should precede the design of the organization's structure. In other words: the organization’s structure should serve the purpose of the business processes which, in turn, should serve the organizational goals. If we combine the perspectives of Sol (1988) with the process and structure issues, we arrive at the following framework for the design of organizational coordination:

<table>
<thead>
<tr>
<th>Coordination design</th>
<th>Process focus</th>
<th>Structure focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macro perspective</td>
<td>Business chains</td>
<td>Organizations</td>
</tr>
<tr>
<td>Meso perspective</td>
<td>Business processes</td>
<td>Departments</td>
</tr>
<tr>
<td>Micro perspective</td>
<td>Tasks</td>
<td>Resources</td>
</tr>
</tbody>
</table>

*Table 1.1: A framework for the design of coordination.*

Although this book deals with all aspects of this framework, we are mainly interested in coordination issues in process design. Our definition of coordination is therefore a process-oriented one.

Kickert (1979) comes close to what we consider the essence of coordination. However, under his definition, the problem remains that of defining relevant relationships.

To prevent a definition that equates coordination with organization (Mooney and Reiley 1931) and in view of the above remarks, we restrict ourselves to a definition that enables designing coordination from the point of view of organizational processes and that combines the following three important concepts:
Organizational coordination: a design approach

- Organizational actors, being humans, groups or machines, capable of carrying out tasks or actions.
- The mutual influencing and interdependence of processes.
- The attaining of objectives that relate to these processes.

**Definition 1.1:** *Coordination is the mutual influencing of organizational processes carried out by two or more actors in order to attain a certain objective.*

We deliberately chose the term influencing instead of adjustment. The term mutual adjustment as used by Mintzberg (1979) denotes only one coordination mechanism, whereas we prefer to focus on all forms of coordination in business processes. The concept of objectives or goals is important because having common objectives is the reason why processes are interdependent. Business processes can influence each other without there being a common objective, resulting in non-coordination. In essence, many coordination *problems* are caused by the pursuit of different goals. Interdependence of processes leading to the need for coordination can take two forms:

- **Synchronization:** Two or more processes have to be synchronized before a next step in the overall process can be taken. For example, before an application for a life insurance policy can be accepted, the client should be checked medically, and the client information should be checked against the existing records. Synchronization can be pooled, sequential, or reciprocal (Thompson, 1967) and may be achieved in several different ways, e.g. by planning, standards or mutual adjustment (see e.g. Mintzberg 1979).
- **Control:** A process may, by issuing instructions, control other processes. For example, the loss claims department of an insurance company could send a damage assessor, in order to arrange for a correct payment. Control can be implemented by plans, by organizational structure and by ad-hoc decisions (see e.g. Kieser and Kubicek, 1977).

Both synchronization and control assume that processes (and the actors that carry them out) are able to exchange information about their work.
Designing Organizational Coordination

In order to synchronize, a process needs to ‘know’ when its partner is finished. In order to control, processes must be able to send and receive instructions. Actors may have to inform each other about the starting or finishing times of their tasks in order to be able to synchronize their work. Actors may issue instructions to other actors in order to get things done. Thus, actors must be able to communicate about their actions and tasks. The next section therefore elaborates on the subject of communication between actors that carry out business processes.

1.3 Communication in business processes

Communication processes in organizations are ‘the processes for gathering, processing, storing, and disseminating the information that enables an organization to function’ (Farace et al., 1977). These communication processes are needed to coordinate processes within and between organizations. Communication and communication technology can play an important role in the coordination of activities and decisions. Keen (1988), for example, argues that telecommunication technology has evolved from an operations tool through an internal utility, into a coordinated business resource (see also Hammer and Mangurian, 1987). In this section we extend our framework for coordination design, focussing on communication and communication technology. We start with an example from the insurance world to show how communication plays a main role in business processes at the micro, meso and macro level.

1.3.1 An example: changing car insurance

We take the example of a person who wants to transfer his car insurance from one insurance company to another. This is a transaction that happens frequently, as competition between car insurers is quite strong in the Netherlands.

Our driver, Mr. Peace, was advised by his insurance advisor to turn to another insurance company because of cheaper rates and better con-
Organizational coordination: a design approach

ditions. Mr. Peace has driven safely and without damage for the last four years and receives a no-claims bonus of forty percent from his current company. His advisor, an insurance intermediary, will arrange the transfer to the new company. To effectuate the change, he first sends a message to the old company to end the insurance contract at a specific date. The old company forwards this message to the acceptance department which changes the records for Mr. Peace and calculates the premium that still has to be paid or refunded. They send a message to the payment office that states the amount due. The acceptance department also sends back a message to the intermediary, to notify the change of records and the amount due. With this message they send a bonus declaration stating that, during the last four years, Mr. Peace did not make any claims.

In the meanwhile, the intermediary has phoned the new company, to apply for an insurance for Mr. Peace. He will send a written application later, but he gets a verbal assurance from the new company that Mr. Peace will be insured starting on the date his old contract expires. This verbal statement can be given after a quick scan of the applicant's data and an inspection of the national lists of defaulters and suspicious cars. When the advisor receives the bonus declaration of the old company, he will send a copy of it, together with the written application, to the new company. The acceptance department of the new company then processes the form, calculates the premium based on the four 'claim free' years and sends the policy form to the intermediary, who passes it on to Mr. Peace. Mr. Peace pays the new premium to the payment department of the new company.

From this example, we observe the importance of communication in interdependent business processes. Especially when dealing with more than one organization, communication becomes a key factor in effectively and efficiently performing business transactions. In the following section, we will enrich our coordination framework with these communicative aspects. The framework will then serve as the starting point in chapter 3 for developing methods and tools for the design of organizational coordination.
Designing Organizational Coordination

1.3.2 Extending the framework

Our initial framework for designing organizational coordination expresses three different perspectives: the macro-perspective, the meso-perspective and the micro-perspective. Streng (1994), regards these perspectives as organizational levels, or layers, and identifies communication within and between these layers, supporting the business processes. In our example we observe three different organizations; the old company, the new company and the insurance advisor. The insurance companies both have two departments involved: the payments department and the acceptance department.

At the micro-level, we see that, in their communication tasks, employees of the different organizations may be supported by information and communication technology (see e.g. Picot and Reichwald, 1984, Keen, 1988). The employee at the acceptance department of the new company, for example, has to communicate with the system that contains the list of defaulters and suspicious cars. At the meso level, we identify the communication between the business processes within one organization. The acceptance department of the old company has to communicate with the payments department in order to arrange payments and refunds. At the macro level, communication takes place between the various organizations. In this case, the bonus declaration is communicated between the two companies via the intermediary.

The communication processes are mostly part of the business processes themselves. The design of communication processes is therefore part of business process design. In case of the macro level of the above example, the process can only be effective, if all insurance companies agree on the bonus declaration system and the lists of defaulters and suspicious cars.

Information and communication technology may be implemented to support the designed communication processes. Our framework for communication design focuses on both process and support elements. Taking the organizational levels as a starting point, this leads to the following extension to the framework:
Organizational coordination: a design approach

<table>
<thead>
<tr>
<th>Communication</th>
<th>Process design</th>
<th>Support design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macro level</td>
<td>Transaction protocols</td>
<td>Communication infrastructures</td>
</tr>
<tr>
<td>Meso level</td>
<td>Communication procedures</td>
<td>(Group) Information systems</td>
</tr>
<tr>
<td>Micro level</td>
<td>Communication tasks</td>
<td>Personal communication</td>
</tr>
</tbody>
</table>

Table 1.2: The extended framework

At the macro level, the design of transaction protocols and communication infrastructures for electronic data interchange (EDI) have been important recent topics, especially in the transport and financial sectors (see e.g. Bolter et al., 1990, Schrijver, 1993, Streekg, 1993 and Berroggi, 1995). At the meso level, communication procedures between organizational processes are becoming crucial for effective operation (Keen, 1988, Keen, 1991). These processes are now being supported by computer networks that operate groupware and workflow software (Meer, 1994). At the micro level we observe that communication gets more important for task execution, in ill-structured or type-II activities as well as in structured or type-I activities. Communication tasks can be easily supported by many kinds of technology, from a simple telephone to electronic mail software, personal voice mail and Internet software (Stallings and Van Slyke, 1994). For the design of coordinated business processes, the design of supporting communication processes and technology can be restricted to the upper OSI layers (see Stallings, 1985, for an overview of the OSI layers). We elaborate on communication and communication technology in chapter 3.

1.4 Research scope and objectives

To support organization designers in solving the problems we described in the above sections, we propose a set of structured methods and tools for the analysis and design of coordinated business proc-

---

1 Panko and Sprague (1984) make a distinction between type I and type II office work. Type I or structured activities are characterized by a focus on efficiency, high volumes of transactions and well-defined procedures and data. Type II or ill-structured activities are characterized by a focus on effectiveness, a low volume of transactions with no explicit procedures or data available.
Designing Organizational Coordination

esses within and between organizations. In the development of these methods and tools, we acknowledge the important role that modern information and communication technology can play in achieving coordination and improving organizational performance.

Our research deals with the development of an inquiry system for the design of coordinated business processes. We follow Sol (1982) in defining an inquiry system:

**Definition 1.2:** An inquiry system is a structured set of instruments which can be used as a context in problem solving activities. It expresses a methodology in view of a problem area (Sol, 1982).

This definition of inquiry systems is somewhat wider than the definition of inquiring systems, as proposed by Churchman (1971), who merely investigates research approaches and their philosophies. We give specific comments on the issue of inquiring systems in section 1.5.

For the sake of clarity, we prefer to use the term 'design system', rather than inquiry system, when we mean the product of this research. To get a clear picture of what we are driving at, we elaborate on the elements of the above definition:

**Structured set of instruments:**

Instruments are concrete or abstract objects, used to perform or support problem-solving activities. Instruments can be for example (parts of) methods, tools, heuristics, procedures, all pointing at the same goal which is solving a problem.

**Problem solving activities:**

Our design system will be aiming to solve problems, that arise from uncoordinated processes in organizations. We regard the development of coordinated organizational processes as a sequence of problem solving activities.
Organizational coordination: a design approach

Expressing a methodology in a certain problem area:

In other words, we are looking for a methodology, supported by tools, for the organization designer that is focused on process improvement by means of better coordination. The problem area can therefore be found in organizations that encounter performance problems with respect to their processes. From this point of view, we state our research objective as follows (see also Eijck, 1992):

To develop a comprehensive set of instruments and methods (a design system) that supports the design\(^{2}\) of coordinated business processes in organizations.

This ‘toolbox of the organization engineer’ must include instruments and methods relating to the following aspects:

- The analysis and description of existing processes, in order to find the relevant coordination and communication problems.
- The detection of this kind of problems from the given description.
- The generation of alternatives for improving performance.
- The estimation of the effects of a specific alternative.
- The process of introduction of an alternative.

Sol (1984, 1985) gives an analytical framework for design methodologies. With this framework, it is possible to characterize design methodologies by their mode of thought, or ‘Weltanschauung’, their modeling constructs, their working method and their management approach. We refer to these characteristics as:

- The way of thinking.
- The working method.
- The modeling approach.
- The management approach.

\(^{2}\) In our opinion, design is strongly related to improvement. Even when designing from scratch, the designer uses a frame of reference, a set of objectives which his design should accomplish. We further discuss this issue in chapter 3.
Designing Organizational Coordination

The *way of thinking* provides the philosophy on which the methodology relies, it gives a 'Weltanschauung', a perspective on the problem domain, and the way in which problems can be solved. The *working method* provides the subsequent steps that, according to the methodology, should be carried out in order to arrive at a superior situation. We note that these steps do not necessarily need to be sequential. Some steps require the use of abstractions from reality, models of the problem domain. The concepts that establish these models are brought together in the *modeling approach*. The *management approach* provides the managerial components of the design system, including the planning and programming of the design steps. Figure 1.1 gives an overview of the elements of the analytical framework.

![Diagram](image)

**Figure 1.1: An analytical framework for design methodologies**

The way in which the methodology is supported by automated tools, can be reflected in a set of support tools\(^3\). The support tools may incorporate elements that support several characteristics of the methodology and are no part of the methodology itself. The combination of the methodology and the support tools define our design system.

---

\(^3\)With respect to organization design, Simon (1973) states that 'one of the most important uses of a computer is to *model* complex situations and to infer the consequences of alternative decisions, to overcome bounded human rationality'.

18
Organizational coordination: a design approach

In the analytical framework, the core elements are the modeling approach and the working method. The working method determines the process to be carried out whereas the modeling approach defines the products (the models) that are to be constructed. The management approach coordinates the processes and the products. We would like to highlight two elementary characteristics of the framework that relate to the modeling approach and the working method:

- The framework implies the use of models.
- The framework can be used as a basis for a problem solving design method, i.e. a design system.

These two observations enable us to come up with a design system that looks upon problem solving as a chain of consistent transformations of models (see Sol, 1982, p.5). As we will show in chapter 4 and 5, models play a central role in our design system.

The above framework is successful in describing design methodologies and design systems (Wijers, 1991, Verhoef, 1993'). However, Meel (1994) argues that it can also be used in a more prescriptive sense, meaning that a design system that addresses the elements of this framework in a coherent manner will have a good chance of being successful. In chapter 4 we use the framework to describe the methodology, and we present the support tools in chapter 5.

1.5 Research philosophy and research approach

The research objective of developing a design system implies studying practice. As the objects of study are organizations, it is hardly possible to design repeatable experiments and formulate (or test) hypotheses. We are not focussing on the generation of large datasets, in order to draw statistically valid conclusions. We submit that the development of a design system for coordinated business processes requires a spe-

' Verhoef (1993) defines it as a 'framework for understanding information systems development'.
Designing Organizational Coordination

cific research philosophy and a specific research approach. The following two subsections describe the philosophy which we adopt and the approach which we follow respectively.

1.5.1 Research philosophy

The objective of this research is the development of a design system. A design system can be defined by the elements of Figure 1.1, combined with a set of support tools. The prime requirement of a design system is that it works, not in theory, but in practice (Checkland, 1981, refers to this as ‘fitness for purpose’). To reach this goal, it is not enough to just investigate the object of study in a structured way, and draw conclusions from such analyses. We should be able to derive relevant problems from the object of study, in order to be able to add elements to the design system that solve these problems. In other words: design systems do not only support the understanding of the object of study, but also the development of alternative ways of arranging it (see Simon, 1969). A design system will therefore always rely on a design theory, which is in turn based on the worldview of the developer of the design system (see Sol, 1988, Wierda, 1991). This implies that there is no ‘best’ methodology or design system (Wierda, 1991), one can only strive for improvement and progress.

The practitioner of science also makes use of a methodology, i.e. a scientific method. Churchman (1971) takes a problem-solving perspective and refers to these methods as inquiring systems. The practitioner of science will choose an inquiring system that results in the best fit between his own worldview and the one on which the inquiring system is based\(^4\). Thus, practicing science, too, relies on worldviews implying that there is no ‘best’ way to practice science. One should merely make use of the appropriate elements of existing methods. Meel (1994) refers

---

\(^4\)Inquiring systems as defined by Churchman (1971), specify certain stages in a problem solving process. Churchman notes Leibnitzean (formal, deductive), Lockean (experimental, inductive, consensual, empirical), Kantian (formal and empirical), Hegelian (synthetic design, based on conflict) and Singerian (synthetic, interdisciplinary) inquiring systems.
Organizational coordination: a design approach

to this observation as the 'more pluralist' view on science. Mitroff et al. (1974) did the same observation and identified $3.4 \times 10^8$ different kinds of inquiry, whereas the contemporary 'work in the philosophy of science has tended almost exclusively to pursue one of two forms'. They argue that major emphasis has been given to the formal, deductive, conventional form of inquiry, denoted by Churchman (1971) as the Leibnitzean inquiring system.

This leaves us with the choice of the inquiring system that comes closest to our view on science. Taking a design perspective, we believe that the Singerian inquiring system, also advocated by Churchman (1971), Mason and Mitroff (1973) and Sol (1982), is most effective for the development of a design system for coordinated business processes. We think so for several reasons:

1. The inquiring system has the purpose of creating the capability of choosing the right means for the desired ends (Churchman, 1971, p. 200), and is therefore teleological or goal-oriented.

2. The inquiring system allows the use and combination of several different disciplines for the purpose of progress in creating knowledge, opening up a possibility for interdisciplinary research (see also Sol, 1982).

3. The environment of the inquiring system is a cooperative one, 'where A wants that goal which will help B in attaining his goals' (Churchman, 1971, p.200). We like to identify A as the researcher, B as the organizations in need of coordinated business processes and the goal of A as the design system for organizational coordination.

Inquiring systems can be categorized by model cycles expressing a framework for the inquiring method to be applied (Mitroff, 1974, Sol, 1982). According to Bosman (1977) and Sol (1982), the Singerian inquiring system implies the inductive hypothetical model cycle. This model cycle is central to our research approach and is, therefore, discussed in the next subsection.
1.5.2 Research approach

The inductive-hypothetical model cycle is advocated by Sol (1982) for the following reasons:

- It emphasizes the activities of conceptualization and problem specification, underlining the specification and testing of premises in an inductive way.
- It opens up possibilities for interdisciplinary problem specification.
- It enables the generation of more than one solution, starting with an analysis of the existing situation.
- It regards the phases of analysis and synthesis in solution finding as interdependent.

We would like to add that the inductive-hypothetical model cycle strongly emphasizes an environment which is cooperative in the sense of our third reason for choosing the Singerian inquiring system. Furthermore, the model cycle allows an empirical starting point of the research as opposed to the hypothetico-deductive one (see Mattessich, 1978, ch. 5), which takes a theoretical point of departure (i.e. a hypothesis). From our design point of view this is extremely important, because the problems that have to be solved using our design system must be drawn from organizational practice and are, in this study, not themselves the subject of hypotheses. It is merely the design system itself that is tested for effectiveness: the design system is supposed to give appropriate results in specific empirical situations. We like to emphasize that a completely unbiased, empirical starting point is impossible; one always wears the colored glasses of the object of study (in our case, coordination in business processes). The inductive-hypothetical model cycle is described in figure 1.2.

The model cycle starts in the upper left corner, with the construction of one or more descriptive empirical models. These models describe a perceived situation in a specific area of interest (e.g. distributed organizations) from a specific point of view (i.e. coordination in business processes). This is usually done by conducting explorative case studies (see below).
The models may contain elements from practice as well as theory. The models are abstracted into a single descriptive conceptual model, describing the essential aspects of the area of interest from the chosen perspectives. The relationship between the empirical models and the conceptual model is, in a weak sense, similar to the object instance - object class relationship (see e.g. Booch, 1994).

The descriptive conceptual model is the starting point for the development of the design system, represented by the prescriptive conceptual model. The design system is prescriptive in the sense that it gives design guidelines for solving the problems formulated in the descriptive conceptual model. Both the descriptive and the prescriptive models are influenced by the worldview (or 'Weltanschauung', as Churchman calls it) of the developer of the design system. The design system is finally applied in practice, usually also in case studies, resulting in prescriptive empirical models that can be evaluated. Evaluation of the test cases will pose the question whether the design system can, or indeed needs to be, improved or extended. The answer to this question may lead to the start of a new cycle.
Designing Organizational Coordination

Even though the cycle states a starting point, it does not specify an end. The prescriptive empirical model seems a logical end point, but the answer to the improvement question will practically never be that the design system is perfect. This implies that the inductive-hypothetical model cycle relates to a process of continuous improvements to the design system, which closely follows the goal of the Singerian inquiring system: progress (see Churchman, 1971).

The model cycle emphasizes empirical research as a point of departure and as a means of evaluation. The choice of an instrument for empirical research is influenced by the following observations:

- We want to investigate contemporary events.
- The area of research needs to be studied in its natural setting.
- We are interested in ‘how’ and ‘why’ questions.

These observations are recognized by Yin (1989) and Benbasat et al. (1987) as conditions under which a research project may benefit from case studies. According to our model cycle, case studies can be used in the first and final phases of the research project. The generation of descriptive empirical models may greatly benefit from a case study approach, as the goal is to describe and explore the area of research (the organization in its natural setting) from a specific point of view (coordination in business processes).

The generation of prescriptive models may also benefit from case studies, but will require the possibility of changing the situation by the researcher, in order to be able to evaluate the design system. In terms of Clark’s typology (Clark, 1972), the former case studies would be evaluation studies (to assess some aspect of organizational performance) and the latter applied studies (aiming to solve a problem by applying appropriate knowledge).

To summarize, we use explorative (descriptive) case-study research for the generation of descriptive empirical models and applied (prescriptive) case studies for the generation of prescriptive empirical models.
Organizational coordination: a design approach

The value of the design system can only be systematically assessed by reflecting on its application in practice and deriving further improvements. The limitations of time available for this research project allowed us to apply the design system in only a few organizations, implying a single round of the model cycle. Evidently, this leaves questions unanswered and opportunities to be seized. However, we assert that the design system as proposed in chapter 4 and 5, is an important contribution to the design of coordinated business processes. This is shown in chapter 6 and 7 where its application in five case studies is described and reflected on. The unanswered questions and the improvements to be made are elaborated upon in chapter 8. The next section presents the practical application of the inductive-hypothetical model cycle to this research project.

1.6 Research outline

This book follows the application of the model cycle in the order in which it has actually been carried out. In chapter 2, we first investigate the results of four explorative case studies held in large, distributed organizations. Their purpose is to generate the descriptive empirical models. The case studies concern a large shipping company, a large bank, a telecommunications service provider, and an insurance company. The case descriptions establish the first phase of the model cycle.

In chapter 3, we define our descriptive conceptual model, based on the interviews and case studies reported on in chapter 2, together with the fundamental thoughts offered in sections 1.2 and 1.3. We investigate four main topics: business processes, organizational coordination and coordination problems, possible indicators for organizational performance, and information and communication technology as coordination support mechanisms. With the descriptive conceptual model as a starting point, we present the design system in chapters 4 and 5. Chapter 4 presents the methodology along the lines of the analytical framework of section 1.4 and subsequently describes our way of thinking, the working method, the modeling approach and the management approach.
Designing Organizational Coordination

Chapter 5 describes several support tools that can be used within the design system. Chapter 6 and 7 lead us to the generation of prescriptive empirical models, presenting in chapter 6 the results of applying our design system to two case studies reported in chapter 2. The application of the design system to previous explorative case studies proves that the requirements set out by these case studies are actually met. Nevertheless, chapter 7 provides three new test cases to which the design system is applied. These cases provide a more independent test of the design system and its practical effectiveness. Finally, in chapter 8, we present our evaluation and conclusions concerning the research project in general, and the design system in particular.
2. Organizational coordination in practice

'The Messenger would hang his head and feel sad and foolish that he had not realized what a tough and complex place the real world was and what difficulties and paradoxes had to be embraced if one was to live in it.'

- Douglas Adams

2.1 Choosing a research environment

For the development of a design system that aspires to practical effectiveness it is important to make a study of problems occurring in practice. We have selected organizations that provide information about the elements we have introduced in the framework of chapter 1 and give insight in the type of problems we want to solve. The four organizations we describe in this chapter all have distinct but interdependent business processes that need to be coordinated. We investigated processes that are interdependent across more than one organization, to get a clear picture of coordination from the macro perspective. We studied processes that are interdependent across more than one location to understand communication processes and the possible use of information and communication technology. All of the investigations took place in large Dutch companies. The case studies concern a large shipping organization, a large bank, a telecommunications service organization and an insurance company and are reported in sections 2.2 to 2.6. The different organizations are selected in such a way that all three perspectives of our initial framework are represented. In all case studies, we were able to use more than one source of evidence\(^1\). We used inte-

\(^1\)Yin (1989) finds multiple sources of evidence essential in case studies: he defines a case study as an empirical inquiry that (1) investigates a contemporary phenomenon within its real life context; when (2) the boundaries between phenomenon and context are not clearly evident; and in which (3) multiple sources of evidence are used.
Designing Organizational Coordination

views, archival sources, project documents, articles and other publications, as well as working experience in some of the case organizations.

Table 2.1 shows the relation between the perspectives of the framework of chapter 1 and the organizations that provided the respective information. All case studies are characterized by their relevant business processes and organization structure, the aspects concerning coordination and communication and conclusions that can be drawn from the descriptions. For reasons of confidentiality, we have anonimized the names of some of the organizations.

<table>
<thead>
<tr>
<th>Macro-perspective</th>
<th>Shipping organization, Bank, Insurance company</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meso-perspective</td>
<td>All organizations</td>
</tr>
<tr>
<td>Micro-perspective</td>
<td>Telecom service organization, Insurance company</td>
</tr>
</tbody>
</table>

**Table 2.1: The research environment in perspective**

The first two cases provide general descriptions of large organizations. The studies are meant to get insight into the coordination problems that large service organizations have. The shipping company case study concerns two open interviews on the topic of organizational coordination with general managers and an investigation into documents related to the company, like annual reports. The case of the large bank concerns an interview with a strategy advisor of the central bank organization, company related documents and personal working experience at several departments of a local bank office. The rough descriptions of the first two cases are followed by a delineation of the type of problem which the design system should be able to handle. Based on this, we discuss the two other case studies and formulate the requirements that arise out of this type of problem. These requirements provide the first clues for the process of abstraction in chapter 3.

### 2.2 Case 1: A large shipping company

The first organization concerns a shipping company called Nedlloyd Lines which is part of a major logistics corporation (Nedlloyd Group) in the Netherlands. The main activity of Nedlloyd Lines is the opera-
Organizational coordination in practice

tion of a worldwide logistical network of shipping services with container-, Ro/Ro- and multipurpose freight vessels. In 1991, the year before this study was carried out, the organization shipped 500,000 TEU\(^2\) of containers and 422,000 tons of conventional cargo of which 65% was shipped with company-owned vessels (Nedlloyd, 1991).

### 2.2.1 Organizational processes and structure

Nedlloyd Lines consists of two sub-organizations: a Container Slot Management organization (CSM) and a Container Flow Management organization (CFM). The Rotterdam based organization CSM, operates the network and performs fleet management and maintenance. In effect, CSM supplies container slots\(^3\).

![Diagram of the Nedlloyd Lines organization](image)

*Figure 2.1: The Nedlloyd Lines organization*

---

\(^2\)Containers are constructed in different sizes. The most popular sizes are 20 ft and 40 ft long. The amount of shipped cargo is measured relative to the first of these two sizes: TEU means: Twenty feet Equivalent Units

\(^3\)A container slot is one cell in a container vessel. CSM also supplies shipping capacity for conventional cargo.
Designing Organizational Coordination

CFM is the sales organization responsible for the flow of cargo (especially containers). CFM arranges the transfer of cargo from the shipper to the consignee (when this transfer involves the crossing of an ocean or sea). CFM is a network of 14 commercial sales offices, distributed worldwide. The sales offices are all responsible for their own turnover and profit. The organizational chart is given in figure 2.1.

The process of shipping cargo is depicted in figure 2.2 and proceeds roughly as follows: A shipper wants to ship a container to a consignee. If necessary, CFM arranges land transport to get a (Nedlloyd) container at a container depot and transfers it to the shipper. When the container is filled up, CFM arranges transport to the port where the container is placed in a CSM vessel. CSM ships the container across the ocean where CFM takes over and arranges land transport to the consignee. Eventually the container may end up in a container depot again.

![Figure 2.2: Division of activities between CFM and CSM](image)

2.2.2 Coordination and communication aspects

For both suborganizations, world wide issues are coordinated centrally from Rotterdam. The sales offices are authorized to coordinate local operations only. Coordination between the different parts of the organization or with other organizations (e.g. stevedore companies, customs, port authorities and many other actors) can be difficult because of geographical distribution and functional separation. For example, coordination between CFM and CSM should take place via Rotterdam because of the central nature of the CSM organization. Sales offices can coordinate mutually, but for important or difficult matters they must rely on the central organization in Rotterdam. Sales representatives can communicate freely with their clients, but for important or big clients,
Organizational coordination in practice

general management will assist and the representative must travel to Rotterdam. To cope with coordination problems that are caused by functional separation, Nedlloyd authorizes sales offices to coordinate operations that directly relate to individual clients.

Specific coordination problems for this organization are the time zones and the differences in prices between countries (because of exchange rates and local price standards). For price coordination, there is a separate office in Hong Kong in order to keep Nedlloyd prices at the same level world-wide.

The business processes of this organization involve intensive information flows between the many parties that take part in these processes. In the first place, in order to perform their tasks effectively, the individual employees have to coordinate their work processes. Second, departments that are geographically separated have to coordinate their business processes so as to serve their clients effectively. Third, the organization itself will have to adjust to other organizations that are frequently contacted (e.g. clients, port authorities and stevedores).

Communication is often still performed by means of telephone. There is also a lot of traveling involved. The organization is using Electronic Mail, Telex, Fax and a range of EDI applications as means of business communication. Large information flows occur between the various sales offices, as well as between the sales offices the central offices in Rotterdam and Hong Kong. The use of information- and communication technology is very intensive but the systems are still centrally organized and operated from Rotterdam. The expenses for IT increase as the shipping network gets more dispersed.

2.2.3 Conclusions

The description of this organization concentrates on issues that highlight the macro- and meso-perspective. The above outline shows the coordination problems that a large distributed organization has to deal with when doing international business. Transactions between Nedlloyd and others have to be coordinated, involving central and distributed ac-
Designing Organizational Coordination

tors. Business processes of the different sub-organizations have to be adjusted to get containers from a shipper to a consignee. All this involves a lot of communication, that has to be supported by information and communication technology. This results in high IT investments and high costs of operation. Coordination and communication issues are therefore an important factor in the design of business processes. Relevant characteristics of this organization are summarized in table 2.2.

| Part of the organization is dispersed (CFM), part is central (CSM) |
| Coordination between distributed and central parts is difficult and costly |
| Information systems are centrally organized and IT expenses are high |
| Differences between locations (e.g. price and time-zone) are a problem |
| Important or difficult matters are managed centrally |
| Several means to support coordination are available: |
| Communication technology (Fax, Telephone, E-mail, EDI) |
| A price coordination office |
| A lot of traveling (with Rotterdam as a central meeting point) |

*Table 2.2: Summary of the Nedlloyd Lines situation*

2.3 Case 2: A major Dutch bank

The second organization concerns a major financial organization in the Netherlands. The organization started in 1972 as a merger of two cooperative bank organizations. The bank traditionally operated on the agricultural market but has developed into an all round financial organization.

2.3.1 Organizational processes and structure

Because of its cooperative nature the organization structure is somewhat different from other banks. The organization consists of three major enterprises: A central bank organization, an insurance firm and an association of approximately 750 local autonomous banks (that are associations themselves). The central bank is responsible for wholesale banking, international banking and information systems development.
Organizational coordination in practice

The local banks are all retail banks and serve as points of sale for the insurance company.

Figure 2.3: The bank organization

The main organizational goal is to supply loans to its clients at an interest rate that is as low as possible. The borrowers become members of the local bank association. The members choose a board committee. The committee decides about the management and staff of the local bank. The committees of all local banks choose a board committee for the central organization. The central bank organization serves as an advisory firm for all the local banks. It also takes over activities that cannot be handled by one or more local banks (wholesale- and international services and IS development). The organizational structure is depicted in figure 2.3. With this structure, the organization can combine the advantages of small- and large-scale banks. We show this by an example:

When a local bank only has a few clients that hold stock accounts, it can rely on the advisory services of the central organization. But when a fairly large portion of the clients is holding shares, the bank may decide (without central intervention) to hire a stock market consultant of their own in order to provide better service and to attract more clients.
Designing Organizational Coordination

With this structure, the bank is able to attain short response times, because it operates close to the market and because responsibilities are widely dispersed. This also implies that processes and services can be adjusted to the local market, without having to coordinate with a central office.

The price that has to be paid for these advantages is the fact that company wide policy making can pose a serious coordination problem. Locally, the problem of bigger (geographically distributed) clients sometimes has to be solved by means of a joint action by two or more local banks.

The local banks have sole responsibility for their local processes. However, many processes may involve more than one bank, and there needs to be adjustment. Except for some formal meeting arrangements and working groups, there is no dedicated technical support for solving these problems. An electronic mail network was implemented but did not succeed. The informal network between local bank managers is very important with respect to overall policy development and local joint ventures.

2.3.2 Coordination and communication aspects

This case also demonstrates coordination problems existing on several levels of the organization. Local bank employees must coordinate their work processes with individuals at other local banks or with the central organization.

For example: when a client moves to another town, he may as well want to transfer his accounts and possibly some other financial services to the bank in his new place of residence. The local bank employee must contact the other bank to establish the services at the new bank against the same conditions.4

At the level of local banks, there are other coordination problems. Local banks may establish joint activities for several purposes, like promotion

4Service conditions may differ locally because of the autonomy of the local banks.
or serving local big clients that are too large for only one bank (joint activity may involve all 750 banks which requires organization-wide coordination). The banks and the central organization also need to coordinate with other organizations (e.g. clients).

Coordination within the organization can be problematic because it may involve as many as 750 autonomous organizations. However, this kind of coordination only concerns adjustments of general policy, like promotional activities, marketing and general price policy. Once a year, a meeting of all board committees is organized. Once every quarter there is an official meeting for local bank managers of a region (for regional cooperative activities). These regional networks choose representatives for the quarterly organization-wide meetings of bank managers. Coordination between two or more local banks is usually no problem because of the informal networks. Coordination of individual work processes is supported in many ways (telephone, telex, fax, data retrieval services and on-line datacommunication). When they occur closer to the operational level, it seems that coordination processes are more informal. Furthermore, there are technical committees for product development. Local banks can participate in these committees by supplying a product expert. In this way, market demands can be transformed into products.

Communication occurs at several levels. At the workplace, local bank personnel can communicate with the advisory services of the central organization via several means of communication (data inquiry by terminal, phone, fax or personal contact). A local bank communicates with another bank or the central bank via several transaction networks. A videotape is sent every month to all local banks in order to inform them about general organization-wide topics and some regional highlights. Also a satellite television network with the same purpose is in a test phase (broadcasts twice a day, every day except Monday). Communication with other organizations (other banks, travel and insurance companies) is possible via telephone, fax, telex or datacommunication. It is recognized that information systems development is carried out in a classical manner, based on a centralized organization structure. As the idea is to serve the customer at a transaction basis, the development of information systems should proceed from this philosophy. However, the
Designing Organizational Coordination

current information systems are organized on a batch basis, communicating daily with the central computing center\textsuperscript{3}. According to Creemers (1992, 1993), information systems in this sector should be developed to support \textit{transaction-oriented} instead of \textit{batch-oriented} processes.

2.3.3 Conclusions

The discussion of this organization shows that coordination between distributed parts of the organization can have strong impact at the micro-level, as well as at the other levels. It also shows that coordination is strongly related to processes. Sometimes, even the most flexible organizational structure cannot solve coordination problems, leading to informal structures. Relevant characteristics of this organization are summarized in table 2.3. The bank case study is also discussed in Eijck and Attema (1993).

\begin{itemize}
  \item Most of the organization is distributed (750 banks), a small part is central
  \item The dispersed parts of the organization are autonomous enterprises
  \item Information systems are centrally organized and IT expenses are high
  \item Organization-wide policy-making is a slow process
  \item Several means to support coordination are available:
    \begin{itemize}
      \item Communication technology (Fax, Telephone, Datacommunication)
      \item Formalized meetings
      \item Technical committees
      \item An important informal network
    \end{itemize}
\end{itemize}

\textit{Table 2.3: Summary of the bank situation}

2.4 Delineation of the problem area

The previous two case studies had the objective of showing the reader that coordination is in fact an important issue in large organizations in the service sector. The cases showed a strong relation between coordination and the way in which organizational processes and struc-

\textsuperscript{3}In the Netherlands, transactions between different banks are communicated via a special banking datacommunications network and a central computing center.
Organizational coordination in practice

tures are implemented. They also showed a relation between the processes and the structure. However descriptive the case studies may be, for the development of the design system, we need some problem situations that indicate clearly the typical environment in which such a design system should function.

We argued in chapter 1 that our design system should:

- solve problems.
- address coordination through communication.
- focus on processes before focusing on structure.

Having presented a general view of the importance of coordination in the first two case studies, the following two case studies show, in a more detailed fashion, problem situations that our design system should be able to handle. The first case concerns a service department of a telecommunications network provider. The case material was provided by the members of the project team in which we were able to participate. Furthermore we obtained project documentation, held several interviews with employees involved in the case and participated in problem-analysis and problem-solving meetings. The second case study was carried out in two phases. The first phase concerned a broad introduction to a large insurance company, by means of interviews with IT managers, account managers, line managers and office workers. The second phase concerned the participation in a specific problem in the general insurance company, where we were able to analyze and investigate coordination problems.

In the conclusions of each case study, we indicate the characteristics that are a potential subject for our design system. These characteristics are related to the three requirements mentioned above: problem solving, coordination and communication, and processes and structures. Finally, in section 2.6, we give a short introduction into the four key aspects that are discussed in chapter 3, so as to extend and fully complete the conceptual groundwork for our design system.
Designing Organizational Coordination

2.5 Case 3: Dutch Telecom service engineers

This case intends to show an organization of which the processes are both dispersed and interdependent at the same time. The case study concerns a service department for business clients of Dutch Telecom. Service departments are regional organizations of engineers who react to clients having trouble with their phone system. In the next sections, we describe the problem situation and the subsequent coordination issues within this department.

2.5.1 Organizational processes and structure

A service engineer solves problems that occur at clients of the telecommunications network. The work that service engineers do is of a dispersed nature; they travel from client to client and communicate with the office by phone to get new assignments. When an engineer starts his day, he phones to the office and collects one or more problems that have occurred at clients and drives to the first client. He tries to solve the problem and notes the amount of used materials and time on a service notice. The client signs it and gets a copy of it. If the engineer has no more assignments to work on, he calls the office again to collect new jobs. After this he goes to his car to drive to the next client. Once in two weeks, the service engineer drives to the office to get a new supply of materials and to drop off the service notes at the administrative department which then proceeds to billing the clients.

The task of the people at the phone desk at the office is to coordinate the distribution of assignments over the twelve service engineers that are active in this specific region. They get service requests from a computer that is connected to the service desks throughout the country. These service desks can be reached by a dialing a special number. The service requests are evaluated and dispatched by the phone desk according to four criteria:

- Required response time, related to the type of service contract.
- Required knowledge and expertise of the service engineer.
Organizational coordination in practice

- Distance between the client and available service engineers.
- Number of assignments already dispatched to specific engineers.

The processes were not without problems. First of all, the engineers’ phone calls to the department took very long, resulting in service engineers being confronted with occupied lines, taking too much of the client’s time. Second, the service notices were regularly misunderstood by the administrative department, because of the poor handwriting of the engineers. This resulted in issuing erroneous bills, and sometimes in the inability to send a bill at all.

2.5.2 Coordination and communication aspects

In the process of serving a client, four distributed groups of people do the work and they have to engage in coordination:

- The people at the service desks to whom clients turn with their problems.
- The people at the phone desk who dispatch the assignments.
- The service engineers who fix the problems.
- The administrative people who do the billing.

The coordination between the processes of the service desks and the phone desk is efficiently managed by a database of problems reported by clients. Communication is unnecessary, except on some very rare occasions. Coordination between the phone desk and the service engineers requires a lot of communication and is neither effective, nor efficient. Coordination between the service engineers and the administrative department is also ineffective, as bills for a specific engineer can only be generated once every two weeks and are regularly misinterpreted.

These problems have led to the desire to be able to understand and redesign the current processes and to be able to estimate the effect of possible courses of action. Of course there were ideas for the improvement of the process, but there existed neither blueprints of the current
Designing Organizational Coordination

situation, nor formal descriptions of possible improvements. Dutch Telecom thought of several alternatives, but could not estimate their respective value. Service engineers could for example be equipped with portable phones, to be able to phone from the car instead of from the client. Service engineers could be equipped with a portable PC, a modem and a printer, to communicate with the phone desk. The phone desk would then only have to enter the assignments into the computer. One could even try to link the modems directly to the computer of the service desks, completely bypassing the phone desk.

2.5.3 Conclusions

Problem-solving

From this case study, we learn that, in order to solve the problem, a design system for organizational coordination should be able to give an accurate description of the current situation, to be able to test the value of alternatives for change. It should be able to describe the business processes carried out by all actors that take part in it. This means that it should be possible to model the workload of people as well as other resources, like computers, databases and means of communication. To summarize, the design system should at least support the following steps:

- **Analyzing** current processes, implying the choice of the right performance measures for the problem at hand, and measuring the performance of the current situation accordingly.
- **Generating** alternatives, involving creative thinking based on the description of the current situation and the performance measures.
- **Estimating** effects of alternatives, involving performance measurement of the alternatives according to the measures as agreed upon, and the comparison of the alternatives with the current situation.
Coordination and communication

In the case study, we see that communication and coordination are very strongly related. Coordination problems can occur through poor communication. With the help of the design system, we should therefore be able to:

- Analyze communication within and between organizations.
- Draw conclusions about coordination, based on this analysis.
- Derive solutions that solve the coordination problems found.
- Design communication processes that support coordination.

We should take into account the new and constantly improving means of communication technology, which may solve problems that would otherwise remain. An example of this is the recent large-scale introduction of mobile phones, enabling people to coordinate their work processes at moments they were unable to before (e.g. in their car).

Structure and processes

Although the service department does not have a very complex organizational process and structure, problems did occur. The problems were formulated in terms of response times, queue lengths and waiting times. These are clearly logistic keywords and lead to the idea of analyzing processes and structures by using logistic concepts. Logistic concepts relate to coordination problems for production processes, so why not for administrative processes?

We investigate these concepts in chapter 3, and incorporate them in our design system. The case study was discussed in a case report (Eijck, 1992), and two scientific papers (Eijck and Attema, 1993 and Eijck and Sol, 1993).
Designing Organizational Coordination

2.6 Case 4: A large insurance company

This case study concerns a major Dutch insurance company. The company is referred to as Insuraco. Insuraco is an international insurance company with 12,000 employees worldwide, divided in the areas Netherlands, Europe and United States. Insuraco is active in all kinds of insurances. Insurances can be categorized into life insurances, general insurances and special insurances. We focus on the area of the Netherlands. Insuraco is reorganizing into business units that are fairly autonomous. We analyze the situation that will exist after the reorganization (end 1992).

2.6.1 Organizational processes and structure

The organizational structure of Insuraco is depicted in figure 2.4. The organization will be divided into several business units, each responsible for a special kind of insurance. The organization is supported by several staff units. In the insurance world, there is a distinction between individual insurances, business insurances and collective individual insurances. This distinction is reflected in the business units. This case study focuses on the primary process of selling and maintaining insurance policies, carried out by life insurances, contract agents and general insurances. Insuraco works with approximately 17,000 independent insurance agents. The greater part of the policies are sold by 3,000 agents that do frequent business with Insuraco. The agents are autonomous and paid on a provision basis. Agents can be divided into two types: regular agents and agents that have a commitment by contract to Insuraco. The committed agents are handled and supported by a special business unit, 'Contract Agents'.

The business process involves four interdependent subsystems:

- Clients.
- Insurance agents (contracted / not contracted).
- Account teams (an account manager and several specialists).
- Insuraco (headquarters).
Organizational coordination in practice

Figure 2.4: The Insuraco organization

The process of selling an insurance policy consists of four activities:

- Making an offer to a client.
- Accepting the insurance.
- Producing and administrating the policy (invoicing etc.).
- Renewing the policy (annually) and handling claims.

These activities and the related tasks must be distributed over the given subsystems which are geographically dispersed. These tasks must subsequently be coordinated in order to be effective. Tasks can be directly related to the primary process (e.g. handling a claim) or have an indirect relation to the process (e.g. marketing). The division of these tasks and the coordination between these tasks will be regarded as a coordination problem. As the businesses of the three business units show many differences, we shall describe them separately.
Designing Organizational Coordination

Life insurances

The Life Insurances Business Unit always deals with independent insurance agents. This means that the agent may propose different insurance companies to his clients based on the clients' needs. Furthermore, a life insurance is not a product that is sold in half an hour. The client must have certain confidence in the insurance company as well as in the agent. Transactions are usually made face to face contact between the agent and the client. Under specific circumstances, a life insurance may involve a medical test.

The organization is adjusted to the nature of the product. The unit is dispersed into 11 areas, each consisting of an area manager and 10 or 11 inspectors. Together they serve about 10,000 agents of which 3,000 are frequently visited. The inspectors are experts in the life insurance business: they have thorough knowledge of mortgage, private and collective life insurances (some big clients for collective life insurances are served from headquarters by special account managers).

Agents may have direct contact with Insuraco headquarters. However, the inspector usually is responsible for contact between Insuraco and the agent. Operational information exchange (like policies and price lists) usually takes place via surface mail. Large agents may have their own contact manager at headquarters if they reach a certain level of importance for Insuraco.

Agents may need support of experts in a certain field (e.g. financial or legal) for a specific client. These experts are available via their inspector. (However, Insuraco must find sufficient reason for supplying such a specialist.) The life insurance business unit does not support agents by any kind of information- or communication technology except for fax and phone. Inspectors use a PC with possibilities for on-line communication with headquarters. A management information system supplies information about the agents. Furthermore the inspectors use fax and phone. They coordinate with each other and with the area manager by monthly face to face meetings. The inspector remains in contact with the agent by means of regular visits. However, effective
meeting time between agent and inspector takes only 30% of the inspector's time.

**General insurances**

Insuraco general insurances is a business unit of Insuraco that sells general insurances as a mass product. The insurance agent is free to suggest any insurance company except direct writing companies to his clients. However, to be able to sell, the agents should receive support from the companies. Out of 3,000 agents, Insuraco chooses a hard core that does frequent business with Insuraco general insurances. These agents are supported by a support team, consisting of a liaison manager and an inspector. The liaison manager deals with the marketing efforts of the agents. When it concerns daily business, contact between agent and insuraco headquarters is direct, and involves operational workers. They contact the liaison manager about problems or marketing and management issues. An inspector deals with the longer term planning of the agents. He or she regularly visits the agents to talk about entrepreneurship, management and future planning. Where necessary, the support team can rely on specialist knowledge that is available at Insuraco. The specialists do not participate in a support team but are functioning as consultants. Technical support of an agent lies in the range of no support at all to full Insuraco software and hardware support for on-line transactions. Technical support is related to the level of responsibility of the agent. This responsibility varies from only advising a client to buy a certain policy to a complete license to sell and accept policies and the treatment of claims.

**Contract agents**

The business unit concerned with contracted insurance agents sells all kinds of insurance policies. The unit tries to commit young, starting insurance agents to Insuraco. In return for this commitment the agent may expect thorough support from Insuraco, varying from automation to legal help and insurance expertise. The agents can be divided into three classe, A, B and C.
Designing Organizational Coordination

From the start, an insurance agent grows to a certain point of maturity and professionalism. This point is usually reached after three to four years. At this point, three things may happen. Some agents prove to be very successful and have a potential to grow very big; these are the class A agents. Other agents are satisfied with their line of business and carry on at more or less the same level of professionalism for the rest of their career; they are the class B agents. The third group loses more and more of their market share and will eventually leave the business (class C agents).

Insuraco concentrates on contracts with class A and B agents. These groups need full support from Insuraco to run their business. They are supported by an agent support team, consisting of several Insuraco employees, adjusted to the needs of the insurance agent. The agent support teams consist of a sales manager, one or more inspectors, one or more liaison managers, an automation assistant and a financial assistant. A team usually supports about 30 agents and is bound to a geographic area and a specific class of agents (A, B or C). The sales manager is in charge of a team and is responsible for support in management, marketing and sales.

The inspectors are responsible for support with specific insurance expertise. The automation manager supports the agent in the implementation of the necessary information systems. The liaison managers are the contacts between the agent and the Insuraco headquarters and are in charge of operational business.

The agent as well as the members of the support team are equipped with several means of communication. They all use telephone and fax for message parsing and they might use a personal computer for online contact with Insuraco Headquarters. Several software packages support on-line contact, other use downloading to update software and data. The workplace for liaison managers is at Insuraco headquarters. The base for distributed workers is at home. However, they often have face to face contact with agents or other members of their support team.
2.6.2 Coordination and communication aspects

Although they show differences in products, organization structure and business processes, the three business units have some aspects of coordination and communication in common. For example, they all share the fact that they sell insurances through insurance agents. It is important that the processes of the agent and Insuraco are well coordinated. This applies to operational, or primary business processes such as the process of accepting an insurance policy, as well as secondary processes like organizing a marketing campaign. We give an impression of the scale of coordination problems that arise from the insurance agent as a sales channel, by describing a specific problem arising in the individual general insurances business unit.

An example: Insuraco individual general insurances

The business of general insurances in the Netherlands is turbulent. Competition between insurance companies is very strong and market shares are dropping for companies that focus on the insurance agent as a sales channel. Many insurance companies try to sell directly (and therefore more cheaply) to the consumer by phone, or by phone and other means of communication. Next to this, clients tend to claim more and more losses. To summarize, the following market tendencies have undermined the traditional belief that turnover means profit:

- The number of loss claims is increasing strongly.
- Consumers demand more flexibility for less money.
- Damage insurances are becoming mass products.
- The agent as a sales channel is becoming too expensive.

For Insuraco, these tendencies have led to an estimated loss share of 60 percent, a commission share of 20 percent and an internal cost rate of 20 percent of the premium turnover. This does not leave much space for profit. The level of commission is strongly determined by the market, so this is a rather fixed figure. The share covering losses cannot be influenced directly (to do this, one should reduce the number of accidents or
Designing Organizational Coordination

fires), so the only possibility is to improve the internal cost rate. This focus on internal cost levels urges Insuraco to rethink and redesign the business processes, which obviously has consequences for the coordination between Insuraco and the agents. To give an idea of the processes and the required coordination, we describe the processes of Insuraco general insurances.

Processes

Early in the morning the mail bags arrive at the mail room. Together with some employees of the Dutch Mail, the mail for the different departments is sorted. This can be complex because insurance agents tend to put all mail for Insuraco into one envelope. Many agents use their own forms, so it is hard to figure out to what department a specific piece of mail should be sent. After sorting, the mail can be sent to the departments or to individual employees. Mail is sent to the acceptance and administration department and the loss claims department.

At the acceptance department, the incoming mail is sorted by one of the employees according to whether it concerns fire insurances or motor insurances. After this, it is sent to the acceptors at the respective subdepartments. The batches are processed here by the acceptors. When an acceptor is not able to accept an order, he sends it to one of the liaison managers who contacts the agent. If the liaison manager reaches agreement with the agent, the order is sent back to the acceptor for system input. However, this does not mean to say that an acceptor never communicates with the agents. He may write or phone an agent to correct errors or omissions on the application form.

Incoming phone calls for the acceptance department are received by a special phone department. Many applications and questions can be handled by phone. The phone desk employees have the same expertise and authorization as the acceptors. Phone desk employees may also send an application to a liaison manager but this almost never happens because they use the same information system as the liaison managers.
At the loss claims department, the claims are sorted (by the two heads of department) into complex claims and non-complex claims. The claims are sent to the appropriate employees (employees that can handle complex claims may also handle non-complex claims). Complex claims require frequent contact with the insurance agent. This is usually in writing but there is also frequent phone contact. The process time of a complex claim may be more than one year. The administrative flows are schematically depicted in figure 2.5.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{administrative_flows}
\caption{Administrative flows at Insuraco}
\end{figure}

The liaison managers play an important role in controlling the primary processes. Not only do they handle the exceptions, but they also monitor the behavior of insurance agents. They do this by registering output performance indicators (like loss amount, number of new applications etc.), by supporting the agent in his commercial activities and by visiting agents, together with an insurance inspector. The purpose of this is to recruit agents that perform according to Insuraco standards, and to keep them performing this way.

\footnote{Complex claims may involve injury, high losses or complex juridical arrangements.}
Designing Organizational Coordination

Problems

The way in which the business processes are organized results in several problems. We identified five types of problems:

- **Disturbances of the business processes by phone calls.** The loss administration process is frequently disturbed by phone calls from agents who want to be informed about the processing of their specific cases. The loss administration process is a rather complex job that mostly involves legal matters and needs concentration. The acceptors did not have this problem because a special telephone desk was installed for acceptance phone calls.

- **Ineffective and inefficient internal coordination.** Due to the relatively low level of expertise of acceptors and their very limited responsibility, they have to communicate frequently about daily matters with their liaison manager. This has a negative influence on the efficiency of both jobs.

- **Performing identical processes by different nodes.** The mailroom, the loss department and the acceptance departments all sorted documents according to their own criteria. In fact, the mailroom used the same information as the acceptors for their search and sort activities. Better organized, the mail room could do the job for the other departments. However, the acceptance and loss departments were not aware of this problem.

- **Inefficiency of communication between agent and Insuraco.** The acceptors have frequent contact with the agents by phone and mail. This is mainly about daily affairs and as a result of frequent errors on application forms, filled out by the agents.

- **Frequent administrative errors.** The fact that letters are still typed manually by people in a typing room, instead of by the acceptors themselves, results in a lot of internal communication about errors in outgoing mail.

Unfortunately, these problems resulted in much backlog and very long throughput times. According to Insuraco, too much time of the administrative employees was spent on communication, internally and with
Organizational coordination in practice

the agents, in order to correct errors. Next to these problems, the management of Insuraco wanted the cost level of the business process to drop.

Insuraco clearly needs a way to get insight into their rather complex business processes and communication structures. Insuraco wants a tool that gives them insight into the key performance indicators of the business processes, not only their own, but also the processes of the agents and into the relation and coordination between the two. They need a method to analyze, design and improve the interdependent processes of agent and insurer. They want to be able to remove bottlenecks and double procedures from those processes. This would help them to improve throughput times, reduce the workload and reduce the costs of the business processes involved.

2.6.3 Conclusions

Problem-solving

From this case study we learn that, on closer examination, some problems that at first glance seem to be internal, turn out to be inter-organizational. It shows that, where necessary, our design system must take more than one organization into account, investigating the effects of problems and solutions from the macro-perspective as well as from the micro- and meso-perspective.

The case study clearly shows that the three perspectives discussed in chapter 1, cannot be analysed and designed independently (e.g. the frequent phone calls between the agents and the loss department are a problem at both the micro and the macro level). The design system should therefore treat the three perspectives as distinct but interdependent levels of analysis.

Coordination and communication

The case study shows that putting extra effort into coordinating processes (e.g. the frequent communication between acceptor and liaison
Designing Organizational Coordination

manager) may actually be counter-effective. It also shows that special task forces for these kind of explicit coordination tasks can have a positive influence on the efficiency of the regular workflow. We observed that administrative errors (made by the agents as well as Insuraco) can result in a need for extra coordination efforts (feedback coordination, see Kieser and Kubicek, 1977, 1983). In the design system, we should be able to identify these special efforts to achieve coordination, and to distinguish between coordination tasks and tasks related to the regular workflow.

Organizational structure and processes

As in the previous case, we observe a problem statement in terms of logistic concepts like workloads, throughput times and waiting times. We also observed interdependent processes that are organized independently (e.g. mail might be sorted once in the process but is sorted three times), resulting in a repetition of processes. We observed that Insuraco operates more processes than actually necessary to do the job (e.g. the mail typed in the typing room, could be generated by the acceptors themselves, using standard letters and a mailmerge facility of the information system they use). A design system providing a clear insight in the interdependence, performance and output of processes, would be very helpful in understanding the situation at hand (which was not the case with Insuraco) and the development of effective solutions. The case study is discussed in a case report (Eijck and Attema, 1993), and two scientific papers (Eijck and Attema, 1994, and Eijck and Sol, 1993).

2.7 Extending the conceptual groundwork

The next chapter presents our views on four key aspects that develop and complete the conceptual basis of our design system. These key aspects are derived from the case experiences as presented in chapter 1 and 2 and from the theories of coordination as discussed in chapter 1.
Organizational coordination in practice

Together with additional theories from various fields, these elements form the conceptual fundamentals on which the design system is to be built. The key aspects that need to be sorted out are the following:

- **Business processes.** The theory and the case studies give rise to a specific view on organizational processes, their location and their interdependence. We borrow some concepts from logistics that can be applied to business processes.

- **Coordination and coordination problems.** We argued in chapter 1 that coordination can be analyzed from three organizational perspectives: macro, meso and micro. We discuss coordination and coordination problems from these three perspectives.

- **Measuring organizational performance.** We experienced more than once that organizations have difficulties in defining the right performance measures for their business processes. As they show the results of improved business processes, and thereby give an indication of the effectiveness of the design system, performance measures are key to the design system.

- **Coordination and technology.** From the case studies, we recognized that information technology can play a crucial role in the coordination of administrative processes. We analyse the role it may play from the macro, the meso and the micro perspective.
Designing Organizational Coordination
3. A conceptual framework for coordination

'When analytic thought, the knife, is applied to experience, something is always killed in the process. This is fairly well understood, at least in the arts. ...

But what is less noticed in the arts -- something is always created too.'
- Robert Pirsig

3.1 Introduction

In this chapter, we lay the conceptual basis for our design system, using our previous investigations of chapter 1 and 2. We do this by touching four subjects on which, in our view, the design system should focus. First of all, if we want to design coordinated business processes, we should have a thorough and clear understanding of what business processes actually are. In section 3.2 we therefore sharpen our view of business processes, based on the impressions of the first two chapters. Armed with this deeper understanding, we return to the subject of coordination and fit it into our view of business processes in section 3.3. In section 3.4, we describe how the performance of a business process can be measured. This is of great importance because it also gives us a way of measuring the effectiveness of our design system in practice. Finally, section 3.5 is devoted to technology that supports the coordination of business processes.

3.2 Business processes

The design and redesign of business processes has received much recent attention from different fields (Meel, 1994). Since the release of Hammer's paper and book on the subject (Hammer and Champy, 1993), the term Business Process Re-engineering (BPR) leads a life of its own in management and information systems literature. Especially
Designing Organizational Coordination

the combination of business process design and the role of IT in organizations has received much attention (Davenport and Short, 1990, Davenport, 1993, Dennis et al., 1993, Hammer, 1990, Fried, 1991, Grover, 1993, Kaplan and Muirdock, 1991, Meel, 1994). Meel (1994) argues that, although they are all successful, the proposed methods differ greatly in definition and even in name. The way in which business processes themselves are defined and treated also lacks coherence.

In this section, we propose to look at business processes in a logistic manner. As our design system has the purpose of arriving at better coordinated business processes, a logistics approach could be helpful. Ballou (1992) states the mission of logistics as follows:

*The mission of logistics is to get the right goods or services to the right place, at the right time, and in the desired condition, while making the greatest contribution to the firm* (Ballou, 1992).

A business process has, in fact, the same goal. With this goal in mind, we define a business process as follows:

**Definition 3.1:** A business process is a prescribed sequence of organizational activities.

A business process is aimed at getting goods or information from the locus and time of its first activity to the locus of its last activity at the right time, and in the desired condition, while making the greatest possible contribution to the organization or organizations involved.

Organizational activities do not stand on their own: they are carried out by resources. We have mentioned in chapter 1 that, according to Bots and Sol (1988), these resources and the processes that are carried out can be observed from three perspectives. In the following three sections, we elaborate on our view of business processes from each of these three perspectives: we identify tasks and decisions from the micro perspective, processes from the meso perspective and transactions from the macro perspective.
3.2.1 The micro-perspective: tasks and decisions

From the micro-perspective, the task and decisions are the atomic units of activity. A task is carried out by a resource in order to arrive at a situation that is preferred (not necessarily by the resource itself) to the situation that existed before the task was executed. Next to tasks, we observe decisions, expressing the choice between two or more courses of action that a resource has at its disposal. Tasks and decisions related to one resource may be grouped into compound tasks, standing for specific sequences of their constituent tasks. The compound task then represents the possible sequences of tasks that can be carried out by a resource. These possible sequences are referred to as task structures (for an extensive discussion, see Bots, 1989 and Dur, 1994). When executing a task, a resource is usually operating on, or handling something or someone. These, what we like to call items, may be anything: products, pieces of information, or even people. For example in the Insuraco case study, an acceptor is executing several different tasks on an insurance application. The task is carried out in order to transform the item into the desired shape, to transport the item to a desired location, to produce a new item or to destroy the item. Executing a task takes time. Furthermore, a resource can only handle a restricted number of items per task execution (a service engineer can only repair one switchboard at a time). The fact that a task takes time results in process time, the fact that a resource has only limited capacity results in waiting times and queues of items in front of a resource (on virtually every acceptors’ desk, we observe a pile of insurance applications). As a result of this view on tasks and task execution, we come to the following definitions:

**Definition 3.2:** An Item is a construct, representing information\(^1\), one or more products, or one or more people.

**Definition 3.3:** A Task is a time-consuming action carried out by a specific resource, either creating an item, destroying an item, changing the condition of an item, or changing the location of an item.

\(^1\) Examples are: documents, memos, files and messages.
**Designing Organizational Coordination**

**Definition 3.4:** A *Decision* is the choice among two or more courses of action which can be followed by a resource, based on conditions specified by the situation the resource is in.

Tasks and decisions play an important role in the analysis and design of business processes. Part of our design system as described in chapter 4 is based on the above definitions.

### 3.2.2 The meso-perspective: processes

When observing an organization from the meso-perspective, we get an overview of several resources, carrying out their tasks. Resources are usually grouped into departments, that carry out specific processes. A business process of a department can be viewed as a chain of tasks of several resources, linked together to achieve a certain goal. The linking of processes of several departments constitutes an organization-wide business process. An organization can have several different business processes that may share resources. For example, in the Insuraco case study we observe a process for accepting insurance applications, a process for administrating policy mutations and a process of administrating and paying loss claims. In all three processes, the mail room and the typing room are used as shared resources. Other departments (e.g. the loss claims department) are dedicated to one business process only. In this example, the three different processes are clearly distinct. However, sometimes several business processes are intertwined and cannot be recognized separately at first glance.

A way to discover and distinguish between the mix of business processes within an organization is to identify the different kinds of items that flow through the organization and follow each of them through the processes that handle them (see also Streege, 1993, p. 90). For example, Insuraco handles applications, policies and loss claims. The business processes are all organized around these three item types: a process for handling insurance applications, a process for handling mutations and a process for handling loss claims.
A conceptual framework for coordination

The meso-perspective is an important perspective for analyzing coordination, because from this perspective, the interdependence between the different business processes can be made transparent. On a lower level of abstraction this concerns the coordination between tasks of resources (e.g. the acceptor and the liaison manager in the Insuraco case) and on a higher level of abstraction between processes of departments (e.g. the phone desk, the service engineers and the billing department). Definition 3.3 can now be extended by the definition of an organizational sub-process or activity, which is part of the business process:

Definition 3.5: An Activity is the combination of a resource or a group of resources and a task or a group of tasks.

3.2.3 The macro-perspective: transactions

From the macro perspective we observe two or more organizations that do business and therefore have to mutually adjust their business processes. The adjustment of processes is usually achieved by making agreements about how things should work. The example in chapter 1 of changing a car insurance to another insurance company shows that a few general agreements about how processes should connect (i.e. in the form of a bonus declaration), coordinate the processes of two insurance companies.

In a sense, there often is no more difference between inter-organizational processes and intra-organizational processes than the fact that an item crosses the organizational boundaries. However, to avoid problems, the how and why of this crossing is usually embodied in a business agreement or governed by law. We define the actual realization of such an agreement as a transaction. A transaction involves at least two resources, one of each organization involved. The resources will connect the business process by communicating through the exchange of items, like in a normal intra-organizational business process. So each time an item crosses an organizational boundary, a transaction takes place. We define a transaction as follows:
Designing Organizational Coordination

**Definition 3.6:** A Transaction is the execution of an agreed procedure concerning two activities, each being part of a business process of a different organization, involving the exchange of items in order to mutually adjust the two business processes.

### 3.2.4 Design system requirements

From the above perspectives on business processes, we can derive requirements for our design system. In order to obtain a clear insight into a business process from each of the three perspectives, we should identify these requirements for each perspective.

The **micro-perspective** provides a focus on tasks and decisions of individual resources. Tasks and decisions depend on other tasks and decisions, even on those from other resources. We assume that tasks are carried out in a certain amount of time and that decisions require no time. To model a resource and its individual business process, the design system should present the following information about the set of tasks and decisions of each resource:

<table>
<thead>
<tr>
<th>Task</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements for executing the task</td>
<td>Requirements for taking the decision</td>
</tr>
<tr>
<td>Accomplishments when executed</td>
<td>Results of the decision</td>
</tr>
<tr>
<td>Time required by the task</td>
<td>Previous task or decision</td>
</tr>
<tr>
<td>Previous task or decision</td>
<td>Next task or decision</td>
</tr>
<tr>
<td>Next task or decision</td>
<td>Interdependence with other tasks</td>
</tr>
</tbody>
</table>

*Table 3.1: Modeling requirements from the micro-perspective*

From the **meso-perspective**, the focus is on processes, carried out by groups of resources. Usually, a business process relates to a specific kind of item, handled by the business process. A model that gives insight from the meso-perspective should provide a description of the resources that relate to the process and the activities that accomplish the entire business process for a specific item. This leads to the following required information about resources and activities:
A conceptual framework for coordination

<table>
<thead>
<tr>
<th>Resource</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task set for the resource</td>
<td>Resources involved</td>
</tr>
<tr>
<td>Links to other resources</td>
<td>Process requirements</td>
</tr>
<tr>
<td>Resource capacity</td>
<td>Process interdependencies</td>
</tr>
<tr>
<td></td>
<td>Previous activity</td>
</tr>
<tr>
<td></td>
<td>Subsequent activities</td>
</tr>
</tbody>
</table>

Table 3.2: Meso-perspective requirements

From the macro-perspective, the focus is on transactions, activities that involve more than one organization. Our design system should provide insight in the resources involved and the procedures that are carried out when a transaction is executed. This leads to the following information about transactions being required:

<table>
<thead>
<tr>
<th>Transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizations involved</td>
</tr>
<tr>
<td>Resources involved</td>
</tr>
<tr>
<td>Business processes involved</td>
</tr>
<tr>
<td>Transaction requirements</td>
</tr>
<tr>
<td>Transaction results</td>
</tr>
</tbody>
</table>

Table 3.3: Macro perspective requirements

Our total view of a business process is schematically depicted in figure 3.1. A business process may start with a transaction, proceed through a sequence of activities constituted by tasks and decisions, carried out by resources and end up in another transaction, crossing an organizational border and proceeding into the next business process. We use this view on business processes to arrive at a way of modeling in chapter 4.

3.3 Coordination and coordination problems

In the case studies we have observed unsatisfactory situations that need improving. To solve this kind of problem, we should have an idea about how it emerges and what causes this kind of problem. In this section we present our view on coordination problems in general and from the three perspectives of our framework in particular.
Designing Organizational Coordination

![Diagram of business processes in perspective]

**Figure 3.1:** Business processes in perspective

### 3.3.1 Analyzing coordination problems

The problems that are dealt with in this study are organizational or business problems. We adopt the definition of Dur of a *business problem* (Dur, 1992, see also Sol, 1982):

**Definition 3.7:** A business problem is a situation in an organization that meets the following conditions: One or more problem owners 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), they have alternative courses of action available, implying changes in the business processes or technologies applied in the organization, the choice made can have significant effect, and they are in doubt as to which alternative should be selected.

This definition is similar to the definition of a problem by Ackoff (1981) and by Sol (1982). We like to emphasize that the problem owner
A conceptual framework for coordination

may not be aware of all possible courses of action, and that the process of solving a problem may include the design of a desirable alternative (see also Bots, 1989). Combining our notion of coordination as elaborated in chapter one, and of a problem as defined above, we arrive at our definition of a coordination problem:

**Definition 3.8:** A coordination problem is a situation in an organization that meets the following conditions: One or more problem owners are dissatisfied with the performance of one or more activities in the light of the adjustment of these activities to other activities, inside or outside the organization. The problem owners have alternative courses of action available, each of which has significant effect, and they are in doubt as to which one to choose.

Operational adjustment of processes can, in our view, only happen by means of communication between these processes. Every time communication occurs, three activities have to be carried out. Before communication takes place, the right information should be gathered for transmission. Then the information must be transferred between the processes, and finally the information should be processed in the receiving process. Each of these communication-related activities may cause coordination problems. First of all, errors can be made in gathering the right information for transmission. An example of this can be taken from the Insuraco case study. Applications for new policies are sent in to Insuraco by insurance agents who have to fill in an application form for their clients. This application form is generic for many types of insurances and the agent just fills in the entries required by a specific insurance. Inexperienced agents tend to forget to ask the client for specific data or to fill out some entries. This usually causes a lot of extra communication between Insuraco and the agent to get the record of the client right and to produce the proper insurance policy. Because many agents use their own forms, this happens often, slowing down the application process. Problems may also arise in the actual transmission of information. This happens when a piece of information sent fails to arrive at the receiver in the correct format. There are three possible reasons for such problems:
Designing Organizational Coordination

- **Translation problem:** the actual content of the message differs from the intended content (e.g. when the price in a sales transaction between Belgium and France is specified in ‘francs’, the receiver may be confused about what currency is being used, thus necessitating communication later on. To overcome this problem, official abbreviations for each currency have been developed).
- **Interpretation problem:** the content of the message was not understood correctly (e.g. when a service engineer misunderstands the address of a client when calling the phone desk, he will arrive at the wrong address).
- **Transmission problem:** the quality of communication is insufficient to transfer the message correctly (usually a technical problem, the distorted communication between a Ph.D. student and his professor over an analog mobile phone, used in dense traffic, may serve as an example).

If none of the above problems occurs, it is still possible that the information is handled incorrectly by the receiving process. This happens when a wrong activity is initiated, or when the receiver does not know which action to undertake based on the incoming information (for example, an insurance acceptor, having received an application for a car insurance, may enter the client’s data in the policy system, forgetting to look up the car in the registration system for stolen cars).

These examples show that communication problems can be a major factor in causing coordination problems. However, there are more causes for coordination problems in organizations that concern communication. Organizations consist of people, and people may have problems communicating with other people. This may have consequences for coordination between parts of the organization. We refer to these kinds of problems as human interaction problems, ranging from simple misunderstandings to severe arguments. However, these problems are beyond the scope of this thesis.

Coordination problems can be observed from each of our three perspectives. We investigate the specifics of coordination problems in each perspective in the following three sections.
3.3.2 Micro-coordination problems

From the micro-perspective, we focus on the resource and its tasks and decisions. We analyze possible coordination problems from this perspective and relate them to the concepts we proposed in section 3.1.1.

Two things can happen with a resource: a resource may be triggered by incoming stimuli to perform a task, or the resource may need information in order to perform a task properly. An example of the first situation is the acceptance employee in an insurance company, receiving an application form he has to process. An example of the second situation is the same acceptance employee receiving an application form that is not yet filled in completely. Next to these situations there may be time and capacity constraints: if the stimuli are not delivered on time, or if the resource is busy, problems may occur. All types of coordination problems may arise from the above two situations. For our design system this implies that we should be able to model the conditions under which tasks can and will be executed. An analysis of these conditions may reveal possible coordination problems and their causes. We note that these conditions are dynamic and the delivery of stimuli is time dependent. Therefore, our design systems should include a time dimension.

3.3.3 Meso-coordination problems

From the meso-perspective, we analyze business processes and try to locate coordination problems that involve several resources and activities. From this perspective, we are able to analyze the interdependence of processes. For example: the acceptance employee has to get permission of his superior to accept a certain policy, but his superior is on holiday, so the policy cannot be accepted and the client finds himself another insurance company. Interdependent processes have to exchange stimuli in order to interact. So, if we analyze the communication between processes, we may find coordination problems and their possible causes. For this, we need to understand the communication links between resources and the interdependence of processes which we have included in our modeling requirements in section 3.1.2.
Designing Organizational Coordination

3.3.4 Macro-coordination problems

From the macro-perspective, coordination problems can be found in the agreements between organizations. Either these agreements themselves cause the problems or the execution of these agreements does. When a transaction itself causes a problem, the requirements or the results of the transaction are probably not specified with enough clarity. When the execution of a transaction results in a problem, one or more of the business processes concerned are probably the cause, and one should investigate them from the meso-perspective. An example of the first situation is a simple purchase of goods for which no clear price has been agreed previously. The second situation occurs when, in the example of section 1.3.1 (change of insurance company), the insurance agent forgets to send the bonus declaration to the new insurance company.

3.3.5 Design system requirements

Our design system should enable a business process analyst to identify coordination problems from several perspectives. We have investigated briefly in this section where such an analyst may start looking. From a logistic point of view, business processes, and this also applies to coordination in business processes, are meant to achieve two things:

1. Delivering the right items to the right resource at the right time.
2. Carrying out the right activities with the items that were delivered.

From the micro perspective this means that resources should have a clear view of which items can be delivered and what has to be done with them. This means that the design of individual task structures is of crucial importance for achieving coordination. From the meso-perspective, one should analyze and design the structure and sequence of processes to achieve coordination in a whole business process. This also requires thorough design of exception handling, as exceptions usually cause coordination problems. From the macro-perspective, this requires the design of clear agreements or transactions, again including procedures for possible exceptions.
From all perspectives, the time dimension is very important and our design system should therefore include ways to analyze and design the dynamics of business processes. We use these insights in constructing our design system, specifically as regards the working method and the modeling approach.

3.4 Measuring organizational performance

To improve whatever needs improving, including coordination in business processes, one must have at least a frame of reference in order to be able to estimate the effects of measures taken (see Rummler and Brache, 1990). Change is always measured by means of a comparison with the situation before the change has taken effect. In this section we investigate four dimensions along which performance in organizations can be measured. We distinguish between cost, time, capacity and quality. Each dimension has its own characteristics, and before anything can be changed, the people who are responsible for the change should reach agreement about which dimension or dimensions they want to change, and how. Our design system should include these dimensions of performance and offer instruments for effective measurement and comparison. The following sections relate the dimensions of measurement to the analysis and design of organizational processes.

3.4.1 The cost-dimension

Business processes are costly and therefore need to be organized efficiently. From the case studies in chapter 2 we learned that insight in process-costs is strongly desired. Mostly, organizations are used to either relate costs to products or to resources (Horngren, 1982, Linden, 1991, Shank and Govindarajan, 1989, Theeuwes, 1990). However, for our design system we want to measure the costs of individual activities and processes as well. We require performance measures that relate more closely to the processes themselves. Thinking along the lines of our view of business processes of section 3.1, we would like the design system to incorporate three different ways to account for costs:
1. The costs related to individual resources that carry out the processes (e.g. a retail bank wants to know the difference in costs between a cashier behind a counter and an automatic teller machine).
2. The costs related to individual items that are handled by these processes (e.g. Insuraco wants to know the costs related to a specific kind of insurance policies).
3. The costs related to individual activities and processes (e.g. Dutch Telecom wants to know what it costs to repair a phone system).

3.4.2 The time-dimension

In the business processes we encountered in the case studies, we noticed that time plays an extremely important role. Insuraco, for example, was very concerned about the total time which the acceptance and loss claims processes were taking. The satisfaction of the insurance agents about the performance of the company was greatly determined by the average response time. In the service-engineers case, too, response time was a crucial factor, and waiting times for engineers had to be reduced. In our view, there are three ways in which time can be related to a business process:

1. By the processes or activities themselves (Process time).
2. By queueing until a process can be executed (Queueing time).
3. By routing between activities (Transfer time).

Total throughput time of an item through a business process can therefore be determined as process time + queueing time + transfer time. If we want to identify performance values in the time dimension, this leads to three basic requirements for our design system:

- The measurement of process time and queueing time per individual activity.
- The measurement of process time, queueing time and transfer time per item that is handled by the business process.
- Useful aggregations of the above measurements.
3.4.3 The capacity-dimension

The performance of business processes is strongly influenced by the capacity of the resources that carry out these processes\(^1\). The insurance acceptors and the service engineers of the case studies in chapter 2 are confronted with this on a daily basis. Whenever one of their colleagues has time off, they see the number of assignments increasing on their to-do lists. Apart from the performance measures as given under the time dimension, the capacity of a process can be measured by the following two criteria:

- Process speed (number of items leaving the resource per time unit).
- Workloads or queue length (number of items waiting for a process).

These two criteria are strongly related and their values depend on the type of queuing system that can be identified for a set of resources. The capacity of a queue / resource combination can be investigated by acquiring three variables of the process (see Grimmett and Stirzaker, 1982):

1. The inter-arrival function (representing the time between the arrival of two items).
2. The serving time function (representing the time spent by the resource per item).
3. The number of resources that serve one queue.

The design system should enable the modeller to incorporate these three types of variables in models of business processes.

\(^1\)Goldratt and Cox's 'The Goal' (1986) is an excellent novel, vividly illustrating the problem of capacities in production processes, equally applicable to administrative business processes.
Designing Organizational Coordination

3.4.4 The quality-dimension

The final dimension of performance measurement is the quality-dimension. To make the measurement of quality feasible, it should always be measured relative to previously defined goals or requirements. Crosby states that quality is conformance to requirements (Crosby, 1980, Crosby, 1984). In production environments, quality is often measured as the probability of producing a defective product. Zero defects is the common term for absolute quality with respect to the production process. Crosby (1984) even argues that zero defects is the performance standard.

We take a slightly broader approach and define the number of failures in a business process as our quality indicator. A failure is defined as any situation occurring in a business process that does not meet previously agreed standards. These standards can be defined using the above measures of a more quantitative nature. A business process of absolute quality is then defined as a business process where no failure situation occurs. For our design system this means that we need instruments that enable the modeller to identify goals and failure situations in business processes, from the micro-, the meso- as well as the macro-perspective.

3.5 Coordination and technology

Many recent publications have addressed the subject of technological support for coordination (for an overview, see Vreede, 1995). Most of these publications have been in the field of collaboration and group support systems (Crowston et al., 1987, Flores et al., 1988, Malone and Crowston, 1990, Opper and Fersko-Weiss, 1992), with a special emphasis on electronic meeting systems. These systems primarily focus on decision-making processes in organizations. However, for support-

---

3Pirsig (1972) presents an excellent treatment of the subject of quality in his novel 'Zen and the Art of Motorcycle Maintenance'. We avoid taking the same approach, but we could not let his work go unnoticed.
A conceptual framework for coordination

ing coordination in primary business processes, other technologies draw our attention. In this section we give a short overview.

Again, we use our micro-, meso-, macro- classification to describe the technologies that have recently emerged to support business processes. From the micro-perspective we take an interest in the latest developments in workplace support and personal computing. From the meso-perspective, we analyze the recent public fascination with workflow automation and workflow management. The macro-perspective confronts us with the field of electronic data interchange. All three types of information systems have the aim to improve the effectiveness and efficiency of business processes, and can be viewed as electronic support systems for coordination in business processes.

3.5.1 The micro-perspective: personal computing

Electronic support of tasks and personal coordination in administrative environments is currently done by means of personal computers. With the advent of relatively cheap local area networks and the concept of client server computing, tasks of resources can be easily coupled and coordinated. Nowadays, many software applications exist to support the information worker in his daily activities. Many desk workers do not have any problems operating database systems, spreadsheets, word-processors and many other personal tools. The tools can be nicely grouped and given the same look under operating systems that contain modern graphical user interfaces. Panko (1988) gives an excellent overview of possibilities for supporting desk workers. The design, however, of applications that support desk workers in the business process they are part of, does not come to an end by simply putting standard software on their desktop machines. It usually involves the design and implementation of task specific software. This software should be developed, based on descriptions of tasks that are to be carried out in a business process.

'Sol and Bots (1993) refer to ‘information workers’ as ‘workers in information intensive organizations’
Designing Organizational Coordination

Hirschheim (1986) argues that methodologies for office systems that are more participative in nature, are more likely to succeed in developing effective task support.

Although in most cases conventional PC's are sufficient, sometimes the hardware is also task specific (e.g. the service engineer of the Telecom case study in chapter 2 requires personal communication devices to receive assignments and send in the results). For our design system, this means that a clear description of tasks of resources is essential for the development of workplace systems, software as well as hardware. Based on an optimally designed task description, software and hardware requirements can be established, resulting in a work environment that suits the tasks that have to be executed.

3.5.2 The meso-perspective: workflow management

The process view of organizations immediately relates to the products or information that flow through these processes. For this process view, we have added the item and process concept in section 3.2. The flow of products and documents through organizations has received much attention in practice and academia recently. Information systems that support the flow of products is currently a major focal point in the information systems field (see Deurvorst, 1995, Joosten en Aussem, 1994, Meer and Verbraeck, 1995, Uijlenbroek, 1995).

A workflow management system is described by the Gartner Group as 'the automation of work among users where the system is intelligent enough to act based on the definition of work types, users, tasks and the recognition of dynamic processing conditions' (Anderson, 1993).

Workflow management is in fact itself a system that coordinates work. An electronic document (that is, the object of a sequence of activities) can be routed through the organization, and along several computer applications, by means of a workflow system. Based on conditions of the process (e.g. queues before a resource) or the document itself (e.g. entries already filled out), the workflow system may decide on the next resource that gets the document and who gets a copy. A schematic but
A conceptual framework for coordination

general view of workflow systems is given in figure 3.2: resources A through D are using applications 1, 2 and 3 to handle the incoming items, based on the condition of the item. The workflow management system coordinates the execution of the applications for the processes that have to handle an item.

Uijlenbroek (1995) distinguishes between workflow systems that manage type 1 tasks and systems that also involve type 2 tasks (see also Panko and Sprague, 1984). Systems that involve type 2 tasks let the end-user decide on the eventual routing of the electronic document. In type 1 systems, within certain boundaries, the routing of a document is rather rigidly defined. Again, we argue that, before any workflow system can be implemented, the business processes it should support should itself be looked at. The processes can then be defined in a workflow system that is interfaced with the primary applications that support the business processes. Mostly, only parts of the business processes can be supported by a workflow system.

![Figure 3.2: Schematic view of workflow systems](image_url)
Designing Organizational Coordination

For our design system, this means that process descriptions should be clear and easy to convert into the process definitions of workflow management software packages. Item routing decisions should be made explicit in order to be able to establish a conversion.

3.5.3 The macro-perspective: electronic data interchange

Business processes frequently do not end when a product or information leaves an organization. We have recognized this and added the notion of transaction to our conceptual basis in section 3.2. At the end of the eighties and the beginning of the nineties, the information systems field was characterized by a strong focus on solving inter-organizational problems. Journal and conference papers and dissertations concentrated on topics like 'Interorganizational Information Systems' and 'Electronic Data Interchange (EDI)' (see e.g. Benjamin et al., 1990, Choudhury, 1988, Eijck, 1992, Streng et al. 1992, Streng, 1993, Wierda, 1991 among many others). Most publications were mainly focused on the potential strategic value of EDI applications and the competitive advantage that EDI should bring to its users.

However, we would like to take another angle and see EDI systems as support systems that coordinate business processes via transactions. Ediforum (1991) defines EDI as 'the electronic exchange of formatted, transactional information between computers of different organizations'. We use a slightly different definition, one that includes the supportive nature of EDI systems:

**Definition 3.9:** Electronic Data Interchange is electronic support for the coordination of business processes of different organizations via transactions.

From this definition, it is easy to understand that EDI systems can be significant instruments for improving coordination from the macro-perspective. Benefits of EDI can be expressed by the performance indicators of section 3.4. We summarized some benefits as given in the literature in table 3.4 (see also Streng, 1993 and Eijck, 1992).
A conceptual framework for coordination

<table>
<thead>
<tr>
<th>Financial dimension</th>
<th>EDI systems save costs and stimulate more transactions.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time dimension</td>
<td>EDI systems increase transaction speed and decrease throughput time.</td>
</tr>
<tr>
<td>Capacity dimension</td>
<td>EDI systems can handle a higher workload than people; they reduce the need for human capacity.</td>
</tr>
<tr>
<td>Quality dimension</td>
<td>EDI systems reduce the number of errors made in transactions.</td>
</tr>
</tbody>
</table>

Table 3.4: Benefits of Electronic Data Interchange

To establish these benefits for the users of EDI systems, the development of these systems should be taken on by a group of participants that has a common goal (Wierda, 1991). This group has to decide on which information is going to be exchanged, the standards for the messages that establish the transactions, (e.g. the UN EDIFACT standard) and the infrastructure that is used for communication (e.g. a Value Added Network that uses mailboxes for message parsing).

However, Streng (1993) points out that ‘the general opinion concerning the technical infrastructure side of EDI is that it is not a bottleneck in EDI development’. Rather, for EDI to be successful, solving organizational and market problems is of vital importance. We argue that before developing an EDI system, the business processes of the participants should be well connected to each other and to the market. Key requirement for a good connection is that participants know how the business processes of other participants are designed and how market dynamics can be changed to obtain a competitive advantage.

For our design system, we submit that business process design should precede the design of systems that support interorganizational coordination. Furthermore, the designs of business processes and their subsequent support systems would benefit greatly by being understandable, not only by the owners of the business processes themselves, but also by other participants in the business chain, including consumers. In our opinion, this involves not only static elements such as information sets and task descriptions, but also dynamic aspects such as the timing of transactions between organizations.
Designing Organizational Coordination

3.6 Towards the development of a design system

In this chapter we have molded our findings in practice and in literature into a conceptual framework consisting of four elements: business processes, coordination, organizational performance and information technology. We conclude that coordination in business processes can be seen from a logistic perspective and that coordination problems occur at several organizational levels. Furthermore, it is important to measure organizational performance in order to be able to assess the effect of alternatives. Even when a process is designed from scratch, the designer should be able to express the desired performance of the business process to be able to assess the value of his design, relative to his or her own standards. From our three organizational perspectives, we have identified technologies for supporting coordination.

The object of this study is to construct a design system for organizational coordination, from the requirements as expressed in our framework. Chapter 4 presents a methodology that meets the above requirements, according to the lines set out in the analytical framework for design methodologies in chapter 1. Chapter 5 describes the support environment of the design system.
4. A methodology for coordination design

'A full set of rules is so massively complicated that the only time they were all bound together in a single volume they underwent a gravitational collapse and became a black hole. A brief summary, however, follows: ...'

- Douglas Adams

4.1 Introduction

In this chapter we develop the first part of our design system. It concerns the methodology that will support organization designers in designing better coordinated business processes.

The methodology is based on the dynamic modeling approach as proposed by Sol (1982). This approach, in various forms, has been applied successfully in many different environments and research projects (Sol 1992): for example in designing interorganizational information systems for international goods transport by rail (Wierda 1991), in developing an information system to support treasury management and in-house banking (Motshagen 1991), for scheduling support in a production environment (Verbraeck, 1991), for business engineering in information-intensive organizations (Dur, 1992), for the use of fleet-management systems supported by mobile data communication systems (Schrijver 1993), for supporting investment decisions concerning the use of Electronic Data Interchange in the port of Rotterdam, (Eijck, 1992, Eijck et al., 1993, Streng 1993), for the analysis of organizational change alternatives for the Amsterdam Police Force, (Meel 1994, Vreede, 1995) and for the assessment of change alternatives in a major Dutch hospital (Vreede, 1995). The variety of these projects lays the groundwork for the development of the methodology we propose in this chapter.
Designing Organizational Coordination

The methodology is described according to the four characteristics of design methodologies as given in the framework of chapter 1 (Sol, 1984, 1985). The core of the methodology is found in the working method and the modeling approach, as these parts give detailed guidelines of how to operate. The 'way of thinking' gives the overall context of the methodology and the philosophy from which it departs. The management approach focuses on checkpoints and documentation to be made when the methodology is applied.

4.2 Way of thinking

The way of thinking underlying a methodology provides the philosophy on which it relies; it gives a 'Weltanschauung', a perspective on the problem domain, and the way in which elements of the problem domain are interpreted. In chapter 1, we defined our problem domain as organizations that have to coordinate their business processes. We claim that our methodology supports organizations in solving the problem of designing well-coordinated business processes. We stated in chapter 2 that, in our opinion, the (re)design of business processes must be based on a thorough analysis of the existing situation, taking qualitative as well as quantitative issues into account. In the methodology, organizations and business processes are seen as dynamic systems, consisting of (inter)acting objects.

From these statements, we derive the following foundations of our methodology:

- **Worldview**: Organizations and business processes as dynamic systems of interacting objects.
- **Domain focus**: Interdependent business processes in organizations.
- **Methodological perspective**: Business process design as a problem-solving approach that takes the existing situation explicitly into account.
A methodology for coordination design

Our working method should therefore be problem-oriented, aiming to arrive at a better coordinated business process. Our modeling approach should take the dynamics of business processes and organizations explicitly into account and give a clear understanding of the individual objects as well as the system as a whole. The management approach should support the subsequent stages of the problem solving process.

Before entering the realm of the working method, we first elaborate on two main topics that establish the philosophy of our methodology: the object oriented worldview and the problem-solving perspective.

4.2.1 Systems, objects, attributes and actions

We adopt a systems approach as the basis of our methodology. We do so, not only because it has proven to be a useful approach in organization science (Kramer and De Smit, 1977), but also because we think that it can be used as a sound frame of reference for developing analysis concepts and modeling concepts (see e.g. Checkland, 1981).

We follow Sol (1982) in defining a system: ‘A system is a nested structure of entities’. However, we use the term ‘object’ instead of entity to prevent confusion with the term entity, as used in entity - relationship modeling, and also because the term ‘object’ is more generally used in the literature. Hence we define a system as a nested structure of objects, and submit that organizations and business processes can be seen as systems, acting to achieve a certain goal, in other words as purposeful systems (Ackoff and Emery, 1972, Leeuw, 1990). We note that, according to Sol (1982), a delineation of a part of the world as a system is a matter of choice, and that this part of the world can be described by the attributes and actions of its objects (see also Holbaek-Hanssen et al., 1985). This means that the analysis and design of a system (e.g. an organization) is relative to the analyst or designer. Our methodology does not, therefore, aim to arrive at the model of an organization or business process, but at a model that is appropriate for the analyst who uses the methodology.
Designing Organizational Coordination

The description of a system is given by a description of each of its objects. The properties of an object are described by its attributes and actions. The attribute part contains data elements describing the state of the object. The action part contains the tasks and decisions that reflect the possible behavior of an object.

We follow Bots (1989) and Dur (1992) who state the following advantages of the object-oriented approach in (conceptual) modeling:

- The elegance of defining things in terms of what they are, the attributes and what they do, their actions.
- The natural distinction between conceptual and empirical models by separating object classes from object instances.
- The possibility of modeling autonomous behavior within objects.

Furthermore, objects may inherit properties from other objects. In our modeling approach, we do not follow the other principles from object oriented programming, like polymorphism and encapsulation, as these properties strictly relate to programming and provide no added value in (conceptual) modeling.

Objects and coordination

In order to cooperate and coordinate in an organization, the constituting objects need to communicate. Hence, we should see an organization as a communicating system, a nested structure of communicating objects. The description of such a system is given by a description of each of its objects and each of the existing relations between objects. In our modeling approach, we thus have to describe the relevant objects in organizations, how these objects interact, communicate, and thereby coordinate their activities, and how they form part of the organizations’ processes. Consequently we add to our definition 3.1 of a business process that the sequence of activities is carried out by interacting organizational objects. We refer to Booch (1994) for further properties of objects in the object-oriented approach and defer further elaboration on the objects in our modeling approach until section 4.4.
4.2.2 Problem solving

Due to the many actors involved, the interdependence of processes and organizational units and the many tasks that have to be carried out in order to arrive at the desired result, improving business processes is a complex activity. Our methodology should support the designer, whenever possible, in reducing this complexity and coping with it when it cannot be reduced. We submit that improving coordination in business processes can be considered as solving an ill-structured problem. An ill-structured problem is defined as a problem that fails to meet one or more of the following requirements (Sol, 1982, p.5):

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

To support the designer in solving ill-structured problems, we propose a problem-solving cycle, derived from the paper "A Systems View of Different Varieties of Scientific Behaviour" as submitted by Mitroff et al. (1974). The cycle is depicted in figure 4.1.

The aim of our methodology is to arrive at a satisfactory solution rather than an optimal solution. This view of problem solving is referred to as procedural rationality, as opposed to substantive rationality where problem conditions are assumed to be perfect and an optimal solution can be arrived at (Simon 1981). Bots (1989) and Sol (1982) refer to procedural rationality as a special form of bounded rationality where the complexity of a problem is far greater than the computational powers available (Simon, 1981). Following Sol (1982), we submit that the bounded-rationality perspective on problem-solving is most appropriate for dealing with ill-structured, organizational problems.
Designing Organizational Coordination

The problem solving cycle consists of six activities, divided in an understanding phase and a design phase. The understanding phase starts with the first activity, concerning the conceptualization of the problem situation. The aim of this activity is to model the problem situation in such a way that a clear, qualitative understanding of the problem situation is arrived at. The main purpose of the second activity (specification) is to construct a descriptive empirical model of the problem situation that can be experimented with in order to get quantitative results for the analysis and diagnosis of the problem situation. The third activity is called correspondence test, or validation, and is carried out to reach a satisfactory correspondence of the empirical model with the problem situation. The validation activity concludes the understanding phase.

The fourth activity initiates the design phase and is concerned with finding solutions for the problem situation. Based upon the results of the problem diagnosis, several alternative solutions may be generated. The alternative solutions are worked out in detail in a number of prescriptive empirical models. These models can be experimented with in
order to study the performance of the alternatives in more detail. All solutions should be expressible in terms of the modeling concepts used in the conceptualization activity. This can be checked by a consistency check. The final activity concerns the actual choice and implementation of the preferred solution in the organization to actually solve the problem situation.

We elaborate on the different activities and steps in the problem-solving cycle in the next section.

4.3 Working method

In section 4.2.1, we introduced a problem-solving cycle on which the working method is based. In this section we give a detailed elaboration of the six activities that constitute the cycle. The activities are strongly related to the modeling approach as proposed in section 4.4, and the requirements of the design system as discussed in chapter 3.

4.3.1 Conceptualization

The aim of conceptualization is to provide models of the problem situation, giving a clear qualitative understanding of it. In line with Bots (1989, p. 34), we state that a conceptual model should consist of the following elements:

- A delineation of the problem area, stating the boundary between the problem situation and its environment.
- Models of the problem situation, providing insight in the objects that constitute the problem situation, and the relations between these objects.
- A problem statement which identifies the problem to be solved in terms of the objects as identified in the models.

From chapter 3 we conclude that the models needed for the analysis of business processes should provide information about the resources
Designing Organizational Coordination

that carry out the activities, as well as about the activities themselves. The resources should be modeled in terms of the tasks they perform and the decisions they take. Furthermore, from the macro perspective, the transactions with other organizations should be identified. In section 4.4 we discuss in detail three model types that provide this information:

- The network model, specifying the resources in the organizational network, from the macro-, meso- or micro-perspective, in terms of objects and their interrelations. The network model also includes a description of the organizational boundaries where transactions are executed.
- The process model, specifying the business process in terms of activities (of resources) and transactions (activities that cross organizational boundaries).
- The actor model, specifying the structure of tasks and decisions for one specific active resource.

The above requirements and model types lead to the following steps to be taken during conceptualization:

1. Identify the object classes present in the problem situation and its environment, using the object definitions of section 4.4.
2. Construct the conceptual model in terms of one or more network models, process models and actor models as specified in section 4.4.
3. Formulate a problem statement, using the object classes as defined as the references to the problem situation.
4. Write conceptualization report (see section 4.5).

4.3.2 Specification

Bots (1989, p. 34) argues that 'Problem specification is essentially no more than a translation of the conceptual model', but admits that 'the actual construction is not without difficulties'. An empirical model should, in our opinion, have the following features (Bots, 1989, Sol, 1982):
A methodology for coordination design

1. It must show a high degree of qualitative as well as quantitative correspondence with the problem situation as perceived in reality.
2. It should be possible to experiment with the model, i.e. it should be executable on a computer in order to obtain numerical results of the model of the existing situation, as well as of the alternatives.
3. It should incorporate the dynamics of the business process under consideration.

A translation from the conceptual model (i.e. the identified object classes) requires the use of a specification language. The above requirements can be met by using a simulation language for specification. With some simulation languages it is possible to stay very close to the objects identified during conceptualization. Furthermore, simulation models naturally reflect the dynamics of the business processes under consideration. Simulation models are also very convenient for testing and comparing alternatives (see Meel, 1994). In section 4.5 we introduce a translation of the elements of the conceptual model into blocks of simulation code. In chapter 5 we introduce a simulation tool that greatly facilitates this translation.

To arrive at a simulation model of the problem situation, the following steps should be taken (see also Meel, 1994):

1. Specify output variables. Output variables are specified to enable a comparison between the empirical model and reality and between the empirical model and models of alternatives. They measure the organizational performance as displayed by the model (see ch. 3).
2. Specify model reductions. Model reductions are introduced to reduce the complexity of reality as displayed in the conceptual model. An example is the introduction of stochastic variables instead of deterministic ones. In fact any model is itself a reduction reflecting the specific interests and perspectives of the modeler. It is, however, important to specify these reductions, in order to prevent major model inaccuracies.
3. Collect empirical (input) data to instantiate the object classes of the conceptual model. To build a model that shows good correspondence, input data that stems from reality is of great importance. Sometimes it can be very hard to obtain this information, and it usually takes more time to collect it than expected.

4. Construct a simulation model using the tools of chapter 5.

5. Construct an animation of the simulation model when necessary. Modern simulation environments like Siman/Cinema™ and Arena™ feature animation facilities which can serve as communication vehicle, verification device and presentation tool (see Vreede, 1995, for an extensive discussion on the use of animation facilities).

6. Specify initial treatment for validation purposes. A fully specified treatment enables the modeler to run the simulation model\(^1\). A treatment of a simulation model consists of (Ören and Zeigler, 1979, Sol, 1982):
   - The specification of input data.
   - The collection of input data.
   - The initialization conditions.
   - The run control conditions.
   - The specification of output data.

7. Write specification report.

4.3.3 The correspondence check\(^2\)

When the simulation model is constructed and the initial treatment is specified, the model can be checked against the expectation of the modeler and against reality. This checking is actually done in three different steps called 1) verification, 2) replicative validation and 3) structural validation. After these three steps, the problem can be diagnosed in terms of qualitative and quantitative values. As a result, the specification report should be adapted and actualized.

---

\(^1\) Runs under the same treatment but with different random number streams for the stochastic variables are called replications. A simulation experiment is the total set of replications for one specific treatment.

\(^2\) We restrict ourselves here to a rather limited description. Specific verification and validation techniques are discussed extensively in chapters 6 and 7.
Verification

Verification is carried out to ensure that the simulation model operates as intended (Pegden et al., 1990, Pegden et al., 1995). This means that the generation of input data, the operation of the model itself and the calculation of the output data should be checked. This can be done in several ways. Meel (1994) mentions software engineering techniques like structured walkthroughs (Yourdon, 1978), tracing and stepping (Pegden et al, 1995). Modern simulation environments like Arena™ have standard verification tools built in.

Replicative validation

Replicative validation is carried out to test the correspondence of output variables as specified in the simulation model to the values as found in reality. This can be done by different types of statistical tests. Usually, the output values resulting from the replications of a treatment are tested against a representative sample of values from reality, using a $\chi$-square test or a t-test of comparison. The results of these tests can be recorded in confidence intervals (see Soest, 1985).

Sometimes it is very difficult or even impossible to gather all the empirical information required for a replicative validation. Often organizations are not used to measuring processes in terms of throughput times, queue lengths and capacities, so the information can only be retrieved by special, and very time-consuming, surveys. In this case, several alternatives are available. These are discussed below, under 'structural validation'.

Structural validation

In structural validation the model is compared to the experience of experts, either people that are part of the modeled situation or people outside the modeled situation, but having a deep understanding of it, or of similar situations. Structural validation can be carried out in several ways, varying from multiple sessions with questionnaires that
Designing Organizational Coordination

survey opinions, to a simple meeting where the model and the model outcomes are shown to the experts (see Herik, 1994). Structural validation becomes more crucial where replicative validation is impossible, or where decision makers have not much confidence in model building. Meel (1995) argues that, where replicative validation is difficult or impossible, stability analysis can be used as an extra kind of structural validation. A stability analysis tests the sensitivity of the model to small changes of the input data, in order to obtain insight into the robustness of the assumptions and estimates built into the model (see also Motshagen, 1991). Animation of a simulation model can play an important role in structural validation, as it provides the means of matching the model to the visual image an expert has of a problem situation (Vreede, 1995). It may also serve as a medium that facilitates the discussion about the structure of the model, the assumptions that have been made, and the model outcomes (Wierda, 1991).

4.3.4 Solution finding

Solution finding breaks down into two activities; the actual generation of alternatives and the estimate of the effectiveness and efficiency of these alternatives. The results of these activities should be documented in an 'alternatives report', discussed in section 4.5.

Generating alternatives

In this phase, solutions are generated to the situation as diagnosed using the empirical model. Solution finding can be done in many different ways and by many different participants in the process. Meel (1995, p. 117) gives an overview. A very promising method is to organize a brainstorm session in which the model of the existing situation is thoroughly explored by a group of participants (see Bouwman, 1995, Kuijlaars, 1995). To speed up the process of brainstorming and to effectively record the results, a Group Decision Room (GDR) which facilitates electronic support for brainstorming and idea generation can be used (for an extensive discussion, see Vreede, 1995).
A methodology for coordination design

Evaluating alternatives

For the evaluation of alternatives, the simulation model can be used. The alternatives can be first conceptualized using, the model types of section 4.4, and then programmed into the simulation model of the existing situation, resulting in a new treatment. Sometimes this means no more than changing the value of an input variable or constant. Sometimes parts of the model must be reprogrammed. The results as described by the output variables can be compared to the results of the existing situation. When necessary, differences between alternatives can be analyzed by means of statistical tests. The results of the evaluation of the alternatives may form the basis of a cost / benefit analysis. Eventually, a recommendation for a decision as regards the implementation of one or more of the alternatives can be made.

4.3.5 The consistency check

The goal of a consistency check is to prevent the models of alternatives going beyond the scope of the problem. Consistency can be checked by analyzing the solutions in terms of the concepts used in the conceptual model. The concepts forming part of an alternative should always be expressible in terms of the conceptual model. If this is not the case, it is very likely that the boundaries of the problem are exceeded, requiring new conceptual and empirical models.

4.3.6 Implementation

It is unwise to give general guidelines for the implementation phase of problem-solving. The way in which implementation is carried out depends on many factors, ranging from financial to social. We confine ourselves here to remarking that the conceptual and simulation models of alternatives provide blueprints for change and can be used for training purposes. When eventually the implementation is completed, it is advisable to perform an 'ex-post evaluation'.
Designing Organizational Coordination

4.3.7 Discussion

The above description of the working method suggests a sequential execution of the activities. However, the model cycle as displayed in figure 4.1 shows that each activity can be repeated in an iterative manner. Conceptualization and specification are strongly related to the modeling approach, discussed in the next section. In turn, the conceptual and empirical models greatly determine the way in which one works. The cases as discussed in chapter 6 illustrate this.

4.4 Modeling approach

The proposed working method requires the use of abstractions from reality, models of the problem domain. The concepts that establish these models are brought together in the modeling approach. According to our working method, we distinguish between two types of models: Conceptual models, constituting the results of conceptualization and Empirical models, constituting the results of specification. We discuss the models in two separate sections, but we first elaborate on the relation between the different model types, as well as on the important role that objects play in the modeling approach.

4.4.1 Model types and their relationships

In sections 4.4.3 and 4.4.4 we establish aspect models relating to the conceptual model and the empirical model. An aspect model is a collection of certain model elements, highlighting specific model characteristics (Vreede, 1995). The relationships between conceptual model, empirical model and aspect models is schematically depicted in figure 4.2.

In section 4.4.3 we propose object definitions through which the problem situation can be ‘filtered’ in order to arrive at a conceptual model. These object definitions relate to the organization’s structure as well as to the organization’s processes, and can be visualized by means of three aspect models. The first is called network model and focuses on the overall
A methodology for coordination design

structure, the organizational network in which a business process is carried out. The activities that are carried out and that constitute the business process are highlighted by the second model type, the process model. This aspect model gives an overview of the processes that are subject to analysis and design. The third aspect model relates to the actual tasks of an active object (an actor) that contributes to the overall business process and is called the actor model. The process model combines the two other aspect models. The processes in the process model are constituted by the activities, as displayed in the actor models, and the objects, specified in the network model, that carry out these activities. For each model type we propose a set of schematic design techniques that enables the designer to visualize the conceptual models.

![Diagram](image)

**Figure 4.2: Framework for the modeling approach**

In section 4.4.4, we introduce the concepts of a simulation language as a 'specification filter' needed to construct the empirical model from the conceptual model. Apart from simulation language concepts, the specification filter includes reduction operators for the aggregation and conversion of empirical data into stochastic distributions. The empirical model is the combination of empirical data, simulation code and an
Designing Organizational Coordination

experimental design, defined by the modeler to perform an analysis of the model. We present two aspect models of the empirical model, one called *experiment model*, focusing on the empirical models' treatments and the experimental results, the other, called *animation model*, focusing on the dynamic behavior of the empirical model.

### 4.4.2 The conceptual model

The essence of modeling organizational coordination lies not only in modeling what activities are performed by actors, but also and in particular in modeling how they cooperate and *influence* each other's activities. In this respect we want to be able to model important organizational phenomena that relate to coordination like: parallel processes, synchronization of activities, and control (management) of activities. We can formulate two requirements that have to be met by models describing organizational coordination:

1. They have to capture time-aspects, i.e. they have to be dynamic models.
2. They have to capture the communication between the actors involved in the coordination process.

The literature reports on several modeling approaches that seem to meet these requirements. The Petri-net formalism (Petri, 1980, Reisig 1985) is one of the first approaches to modeling concurrency and synchronization. Typical application areas are communication protocols, computer systems, and distributed systems (see Aalst 1992). Based on the Petri-net formalism, a substantive number of modeling approaches have been formulated (see e.g. Sol and Hee, 1991, Sol and Crosslin, 1992). However, two reasons make Petri-nets less suitable for modeling organizational coordination. First, they are poorly communicable to people that are not trained in reading them. Second, where the modeled situation gets slightly more complex, Petri-nets tend to grow very large and incomprehensible.

A mechanistic view of organizational coordination is offered by Diplans, a formal graphical language for expressing plans (Holt 1988).
A methodology for coordination design

The strong characteristic of Diplans is that they elucidate the coordination effort in a certain situation. The disadvantages are similar to those of Petri-nets: Diplans are poorly communicable, and they quickly grow very complex and incomprehensible.

Task structures as proposed by Bots (1989) offer a means of modeling the activities (tasks) of individual actors and their relationships. However, coordination that involves two or more actors at the same time cannot be modeled straightforwardly, although the formalism offers a construct for modeling situations where coordination with other actors is required. Task structures have proven to be well communicable.

Dur (1992) proposes task structures from a different perspective: in his task structures, tasks can be performed by different actors, but all involve the same item (e.g. a form or product being processed). Because of this focus on items, it is not possible to describe coordination tasks in which no item is being processed. Furthermore, it is not possible to use single task structures to describe the coordination of business processes that involve more than one class of items. Like Bots' task structures, they have proven to be well communicable though.

Object classes

To model organizations, business processes and coordination, we need objects that carry out activities (and mutually communicate) as well as objects that are passive (and can be communicated between active objects). Passive objects are called items and active objects are called item processors. The description of an activity in terms of attributes can also be seen as an object definition.

The objects item, item processor and activity are combined in our first level of abstraction, called basic classes. The item class is defined by a list of identifying attributes (called identifiers), to be filled in at the lowest level of abstraction. The item class has an empty action part. We use the term 'item' instead of 'message' because an item may represent information as well as physical matter. The item processor class is defined by its identifiers, and an action part.
Designing Organizational Coordination

The action part may receive, process and send items. The activity is described by its identifiers, a list of items required to accomplish the activity, a list of pre- and postconditions and a list of potential subsequent activities. Each activity, when executed, will transform a situation in a business process from the state as defined by the preconditions into a state as defined by the postconditions. Pre- and postconditions may refer to attributes of any object in the conceptual model. The formal object definitions of the basic classes are given in table 4.1.

<table>
<thead>
<tr>
<th>object class ITEM</th>
<th>object class ITEM PROCESSOR</th>
<th>object class ACTIVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>attributes</td>
<td>attributes</td>
<td>attributes</td>
</tr>
<tr>
<td>list of Identifiers</td>
<td>list of Identifiers</td>
<td>list of Identifiers</td>
</tr>
<tr>
<td></td>
<td>list of Status attributes</td>
<td>list of required items</td>
</tr>
<tr>
<td>actions</td>
<td>receive item</td>
<td>list of preconditions</td>
</tr>
<tr>
<td></td>
<td>send item</td>
<td>list of postconditions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>list of next activities</td>
</tr>
</tbody>
</table>

Table 4.1: Object definitions of the basic classes

Within each organization we identify two types of item processors: nodes and links. Nodes and links are together called network classes and both inherit from the item processor object class. Nodes are constructs that constitute places where items can be processed or stored. The construct that connects the nodes in such a way that they can exchange items is called a link. A link is specified by its attributes and may carry out actions related to the kind of technology it represents. It routes items from node to node. It may also change attributes of items (e.g. a link may misbehave and cause distortion). The configuration of the nodes and the links in the organization corresponds to the perception of the modeler of the situation subject to analysis. Together, the objects of the network class form the basic elements of network models. A formal description of the network object classes is given in table 4.2.

Nodes can be further specified on the final level of leaf classes. A node can be either an actor (e.g. an office worker or, on a higher level of abstraction, a whole department) or a repository (e.g. a database).
A methodology for coordination design

<table>
<thead>
<tr>
<th>object class NODE</th>
<th>object class LINK</th>
</tr>
</thead>
<tbody>
<tr>
<td>inherits from ITEM PROCESSOR</td>
<td>inherits from ITEM PROCESSOR</td>
</tr>
</tbody>
</table>

attributes
- list of Other attributes

actions
- store item
- retrieve item

attributes
- starting node
- ending node
- uni- or bi-directional
- list of Other attributes

actions
- route item

Table 4.2: Object definitions of network classes

Both repositories and actors inherit the properties of the object class node. When a node is an actor, its action part is not empty and it can establish state changes in the system. When a node is a repository, its action part is empty (apart from inherited actions) and it contains items that can be examined or changed by actors. Repositories may serve as a coordination mechanism: a database can be the node between workers that have to adjust their processes. The case studies in chapter 6 give some examples.

An object that has a non empty action part can change its own attribute values or those of the items it possesses. Altering attribute values of other active objects must be done by means of communication, i.e. sending an item. An item contains information (in its attribute values) that triggers an action (or more precise: a reaction) in the receiving object; it does not carry out any actions itself. The formal definition of the node classes are displayed in table 4.3.

<table>
<thead>
<tr>
<th>object class ACTOR</th>
<th>object class REPOSITORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>inherits from NODE</td>
<td>inherits from NODE</td>
</tr>
</tbody>
</table>

attributes
- list of Other attributes

actions
- list of Activities

Table 4.3: Object definitions of node classes

95
Designing Organizational Coordination

In modeling organizations it is convenient to introduce a treefold distinction into messages, products and persons. A message transfers information, a product transfers physical matter and a person transfers himself. The object definitions are given in table 4.4.

<table>
<thead>
<tr>
<th>object class</th>
<th>MESSAGE</th>
<th>object class</th>
<th>PRODUCT</th>
<th>object class</th>
<th>PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td>inherits from</td>
<td>ITEM</td>
<td>inherits from</td>
<td>ITEM</td>
<td>inherits from</td>
<td>ITEM</td>
</tr>
<tr>
<td>attributes</td>
<td>Information contents</td>
<td>attributes</td>
<td>Product properties</td>
<td>attributes</td>
<td>Person properties</td>
</tr>
</tbody>
</table>

**Table 4.4: Object definitions of item classes**

Each item processor is capable of performing actions, modeled as activity objects. An activity can be either a choice between two or more courses of action or the performance of a task. We distinguish between decisions and tasks. The combination of tasks and decisions into a more complex activity is called a task structure (see Bots, 1989, Lochovsky, 1983). When an activity contains subactions, we denote the activity as complex. Activities are passive objects, in the sense that they cannot execute themselves, they need an item processor. We propose the object classes in table 4.5 as our activity classes.

<table>
<thead>
<tr>
<th>object class</th>
<th>TASK</th>
<th>object class</th>
<th>DECISION</th>
</tr>
</thead>
<tbody>
<tr>
<td>inherits from</td>
<td>ACTIVITY</td>
<td>inherits from</td>
<td>ACTIVITY</td>
</tr>
<tr>
<td>attributes</td>
<td>delay</td>
<td>attributes</td>
<td>list of decision rules</td>
</tr>
<tr>
<td></td>
<td>list of Attribute assignments</td>
<td>list of Subdecisions</td>
<td>list of Subtasks</td>
</tr>
</tbody>
</table>

**Table 4.5: Object definitions of activity classes**

All the object classes that are used in a specific problem inherit from one of the eight object classes that constitute the lowest possible level of abstraction: the actor, the repository, the link, the message, the product, the person, the task or the decision. We note that, if a person is active in the sense that that person can process items, then he or she is modeled as an instance of the actor class. However, if a person is not active, then he or she is modeled as an instance of the person class. Summarizing, the object definitions can be displayed as in the object hierarchy of figure 4.3.
A methodology for coordination design

Figure 4.3 Object hierarchy for conceptual modeling

A conceptual model which consists of object descriptions alone, is hard to communicate and very difficult to construct. Vreede (1995) argues that it is hardly possible to describe the complexity of organizational processes and their coordination using just object definitions. The aspect models as discussed below provide three perspectives on organizational coordination. Between them they constitute a firm basis for the analysis and improvement of coordination in business processes. The modeler may of course add models of his own, based on the object descriptions, but we feel that, together, the network model, the process model and the actor model make for a rather comprehensive conceptual model.

The Network model

A way to understand and assess organizational relationships is to describe the formal and informal communication and interaction between organizational actors that constitute a business process (Lundquist and Huston, 1990, March and Simon, 1958, Qureshi, 1995). The network model accommodates such a description by means of a diagram displaying organizational nodes, links and items which express the infrastructure for accomplishing a business process. We propose the icons of figure 4.4 for constructing network models.
Designing Organizational Coordination

**Figure 4.4 Icons for the network model**

The network model provides an overview of the item flow between the nodes that constitute a business process.

**Figure 4.5 Example of a network model**
A methodology for coordination design

It is not dynamic in the sense that it displays the flow of items in time. Rather, it just depicts the relationships between nodes that exist for the sake of the business process.

An example of a network model is shown in figure 4.5, which depicts the network model representing the purchase of a house. In this example we want to focus on the dealings between buyer and seller, including the estate agents and the actual closing of the deal by the notary. One could, for example, also include the bank of the buyer, which might provide a loan. In this specific example, we see that the network model may go across organizational boundaries and, if only they are used for that purpose, may even include organizations as repositories. However, this falls short of giving an adequate view of the dynamics of the processes involved. To visualize these processes, we propose the process model.

The process model

The process model seeks to give insight into the processes as carried out to achieve a certain goal. The process model shows a prescribed sequence of activities, carried out by the objects of the network model. To provide such insight, we propose the icons of figure 4.6 for the construction of the process model, as based on Dur (1992).

![Figure 4.6 Icons of the process model](image-url)
Designing Organizational Coordination

A small bomb appears in the upper right corner of tasks and decisions when the element is compound, consists of subtasks or subdecisions. This means that the element can ‘explode’ into more detailed structures. An example of this is visualized in figure 4.7 in which we present the process model of buying a house. Apart from deterministic decision outcomes, the flow direction may be labeled with stochastic outcomes, denoted by probability percentages or the names of stochastic variables. To indicate the use of items stored in repositories, tasks and decisions may be connected to repository symbols by a dotted line.

Figure 4.7 Process model of ‘buying a house’

In the example of figure 4.7, the task ‘arrange transaction’ is a compound task. This means that this task consists of more than one element. To illustrate the possibility of creating hierarchies in process models, we display the ‘exploded’ view of the ‘arrange transaction’ task in figure 4.8.

100
The process models are based on Dur’s task structures. An important difference, however, between our process model and Dur’s (1992) task structures is the fact that we allow all defined item classes to flow through the model instead of only one. Furthermore, we defined separate icons for decisions that include a reference to the concerned actor.

**The actor model**

The actor is the focus of attention in our third aspect model, the actor model. When the conceptualization activity requires a detailed analysis of the tasks of specific actors, these can be described in actor models. The actor model is based on Bots’ (1989) task structures and stresses the activities of a single actor, acting in the business process under consideration. The actor model emphasizes the actor’s personal coordination of tasks. We propose the icons of figure 4.9 for the construction of actor models.

101
Designing Organizational Coordination

Figure 4.9 Icons of the actor model

The elements of the actor model are identical to those of the process model, except that in the actor model, the name of the actor at issue is left out for obvious reasons.

Figure 4.10 Actor model of the buyer of a house
A methodology for coordination design

The main difference between Bots’ task structures and our actor models is the inclusion of compound decisions, as well as the fact that actor models do not show a difference between structured and unstructured tasks. An example of an actor model is given in figure 4.10.

4.4.3 The empirical model

In this section we demonstrate how the modeling concepts as discussed in the previous section can be translated into an empirical model. We submitted in figure 4.2 that an empirical model consists of the following elements (see also Sol, 1982):

- **Simulation code.** We suggested that simulation languages are convenient for the analysis of dynamic processes. We present a generic translation of the conceptual model elements into the Siman IV simulation language (Pegden et al., 1990) as an example.
- **Empirical data.** A simulation model can run only when empirical data is available. For example empirical data may concern handling times, resource capacities and quantifications of item flows. The attributes and actions in the defined object classes provide indications as to the values to be found in, or derived from, practice.
- **Experimental design.** An experimental design aims at testing hypotheses about the model by means of exposing the model to a number of treatments (Sol, 1982). The design of the experiment depends on these hypotheses. The experiment aspect model deals with the exact details of the experimental design.

In this section, we first elaborate on the characteristics of the simulation language used for the empirical models. Second, we will provide the translations of the conceptual modeling elements into this language and, finally, we discuss the animation and experiment aspect models.

**Concepts of the simulation language**

Simulation models can be used to analyze an organization’s business processes and the coordination required for these processes. A simula-
Designing Organizational Coordination

tion model is constructed in three steps, each relating to one of the three aspect models of the conceptual model: the network model, the process model, and the actor model. We start this section with a brief description of the simulation language we used and continue by presenting translations of the three aspect models.

The translation of the conceptual model is done using the Siman IV simulation language, a process oriented language that allows processes to be executed in parallel (see Pegden et al. 1990). Strong features of Siman are its ability to separate empirical data from the structure of the simulation model, its support for statistical analysis and its animation facilities (see also Streng 1993).

<table>
<thead>
<tr>
<th>Block</th>
<th>Operands</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSIGN</td>
<td>Variable = Value</td>
<td>Assigns attribute and variable values</td>
</tr>
<tr>
<td>BRANCH</td>
<td>Maximum, Randomstream,</td>
<td>Controls flow of items through code</td>
</tr>
<tr>
<td></td>
<td>IF, Condition, Label, Primary,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WITH, Probability, Label, Primary,</td>
<td></td>
</tr>
<tr>
<td>CREATE</td>
<td>Batchsize, Offset,</td>
<td>Creates items</td>
</tr>
<tr>
<td></td>
<td>Interval, Maxbatches</td>
<td></td>
</tr>
<tr>
<td>DELAY</td>
<td>Duration, StorageID</td>
<td>Delays an item</td>
</tr>
<tr>
<td>DISPOSE</td>
<td></td>
<td>Disposes of an item</td>
</tr>
<tr>
<td>QUEUE</td>
<td>QueueID, Capacity, Balklabel</td>
<td>Queues up items before a resource</td>
</tr>
<tr>
<td>RELEASE</td>
<td>Resourcename, Quantity</td>
<td>Releases a resource from an item</td>
</tr>
<tr>
<td>REMOVE</td>
<td>Queuelocation, QueueID, Label</td>
<td>Removes an item from a queue</td>
</tr>
<tr>
<td>ROUTE</td>
<td>Duration, Destination</td>
<td>Transfers an item between stations</td>
</tr>
<tr>
<td>SEARCH</td>
<td>QueueID, Startpos,</td>
<td>Searches a queue for an item</td>
</tr>
<tr>
<td></td>
<td>Endpos, Condition</td>
<td></td>
</tr>
<tr>
<td>SEIZE</td>
<td>Priority, Resourcename, Quantity</td>
<td>Lets an item capture a resource</td>
</tr>
<tr>
<td>SIGNAL</td>
<td>Code, Totallimit</td>
<td>Signals WAIT to release entities</td>
</tr>
<tr>
<td>STATION</td>
<td>StationID</td>
<td>Implements a logical unit</td>
</tr>
<tr>
<td>WAIT</td>
<td>Code, Waitlimit</td>
<td>Holds entities until signaled</td>
</tr>
</tbody>
</table>

**Table 4.6 Siman IV modeling blocks**

A simulation model written in Siman can be thought of as a collection of items, described by attributes, flowing through a piece of simulation code. Items can be created and disposed of. Items can be sent to other pieces of code by conditional branch statements. Code that is restricted
to one physical location can be assigned to a station. Items may be routed between stations. Stations may contain queue/resource combinations that act on items. For the construction of a simulation model from our modeling concepts, we use the Siman statements (called blocks) given in table 4.6. There are many more blocks in the language, but only the ones stated in table 4.6 are necessary for our purposes (For a more elaborate discussion of the Siman language, see Pegden et al. 1990; Systems Modeling 1989).

The basic structure: the network model

For each object in the network model we propose a conversion to the Siman IV language. As items can be simply translated into Siman entities with attributes, we do not specify the item construct.

Actors can be modeled with the STATION statement and a QUEUE / RESOURCE combination. An actor is seized by an incoming product, message or person and performs (part of) its activities. An actor has only limited capacity, so, if the actor is busy, the items are queued up. This results in the following code for an actor:

```
STATION, Actorname;
QUEUE, ActornameQ;
SEIZE: ActornameR;
BRANCH, 1: ALWAYS, ActornameAL;
ActornameRA RELEASE: Actorname;
BRANCH, 1: ALWAYS, ActornameLI;
```

In this code, Actorname stands for the name of the actor. This name can be extended with a Q to denote the queue, with an R to denote the resource, with AL to denote the address where the code for the actor’s activities is located, with RA to denote the address where the item returns after the activity has been accomplished and with LI to denote the address of the link to which the actor sends the items after handling them. This construct becomes more complex when an actor shows autonomous behavior while there is no item to handle. The actor should then be kept ‘alive’ by a control item or ‘trigger’, (see Eijck and De Vreede 1994).
Designing Organizational Coordination

The repository can also be modeled by a station. The only task of a repository is to receive, store and supply items. This can be done by means of the following code:

```
STATION, Repname;
BRANCH, 1: IF, Store, RepnameSTO
         IF, Retrieve, RepnameRTR;
RepnameSTO QUEUE, RepnameQ: DETACH;
RepnameRTR BRANCH, 1: IF, NQ(RepnameQ) == 0, RepnameLI;
                    ELSE, RepnameRTRA;
RepnameRTRA SEARCH, RepnameQ: Searchcondition;
         REMOVE, J, RepnameQ, RepnameLI: DISPOSE;
```

An item is received automatically by the station block. Because items are not placed in any order in a queue, the queue must be searched first to find the item that is requested. The RepnameLI address points at the link interface to which the repository sends its items.

After having completed a list of activities, the actor may send items away to link objects. An actor, however, may have more than one link. To be precise, the actor will have one for each actor or repository he or she is connected to. In one way or another, the destination of the item must be determined. This can be done anywhere, so the easiest way is to reserve an attribute in the item to store its destination. This attribute may be altered during an activity or somewhere else. It may also be stored in a global variable or determined instantly when sending the item. The following code can be used for sending an item to a link object:

```
ObjectnameLI ROUTE: 0, Destinationexpression;
```

Since any time delay in transport is arranged for by the link object, the duration is zero. The destination can be determined from the item, from a variable or local expression or can have a standard value. The destination must be a link object.

The link object is modeled by the following code:
A methodology for coordination design

LinknameST  STATION, Linkname;
BRANCH,1:  IF, Active, LinknameGO:
BRANCH,1:  IF, NOT Active, LinknameNOGO;
LinknameGO ROUTE, LinknameDELAY, Actor/Repname;
LinknameNOGO DELAY, LinknameDELAY2;
BRANCH,1:  ALWAYS, LinknameST;

This construct simulates the possible working, or its failure to do so of
the link object. The LinknameDELAY variable stands for the time it
takes to send an item to its destination. (Note that a link has only one
origin and one destination.) If a link is inactive, the item waits
(LinknameDELAY2) and tries again. (This construct can be replaced
by other mechanisms, e.g. the item can be disposed if it is not sent.)
The actual failure of links can be generated by stochastic generators.

With these concepts it is possible to simulate the basic structure of the
network model. We have not yet specified the way in which the tasks
of actors are performed. This is done in the following section.

Coordination in business processes: process and actor models

The business process to be modeled is reflected in the tasks that are
carried out by actors. This means that the activities of the relevant
actors should be specified. In an actor, the item is sent to the initial
task or decision of an activity by the following code:

ActornameAL  BRANCH,1:  ALWAYS, Initial_task_or_decision;

This may also be a conditional branch, determining the initial task
from the attribute values of the incoming item. It might even be that
items have to be copied to accomplish an activity.

We now need simulation code for task and decision objects. Tasks have
two properties: they take time and they may influence attribute and
variable values. Decisions also take time and they may influence the
sequence of tasks to be performed. A task is modeled by the following
piece of simulation code (note that this code always belongs to an actor
STATION):
Designing Organizational Coordination

Taskname ASSIGN: Variable1 = Value1:
                VariableN = ValueN:
                Attribute1 = Value1:
                AttributeN = ValueN;
DELAY: TasknameDUR;
BRANCH,1: ALWAYS, Next_task_or_decision;

The assignment of attributes may also include a destination attribute that determines the flow of the item after the activity is completed. In case of a task that has no successor tasks, the last line of code in this construct is replaced by:

BRANCH,1: ALWAYS, ActornameRA;

A decision can be modeled by the following piece of simulation code:

Decisionname DELAY: DecisionnameDUR;
BRANCH,N: IF, WITH and/or ALWAYS constructs;

In the branch statement, items may jump to other tasks and decisions according to the decision rules or probabilities specified in the decision object. N represents the maximum number of items that exit the decision for one incoming item; this means that the item can be copied by the branch statement.

The decision rules are specified by a structure of conditional branch statements. Note that tasks and decisions are specified on the level of leaf elements. Compound tasks and decisions are constructs that simplify graphical modeling, but have to be expanded (or exploded) when the simulation model is constructed.

The network model and process model describe an implicit form of coordination: the items that flow between actors and repositories. In order to coordinate the tasks and decisions carried out by single actors, explicit coordination in terms of pre- and postconditions has to be defined. We now describe the different ways of implementing coordination issues in the empirical model.
A methodology for coordination design

For establishing coordination that cannot be achieved by item routing, we have to specify the pre- and postconditions for the tasks and decisions in their activities. Preconditions may be expressions that operate on global variables or attributes. There are two ways of implementing this. The first is by assigning values to attributes and variables that may be tested later on (the conditions are implicitly specified by the values of attributes and variables). In this case, the code for the task and decision object needs to be extended with code for precondition testing by using a simple conditional branch statement like the following (note that this code precedes the specified code for tasks or decisions, so the labels Taskname and Decisionname are replaced by TasknameA and DecisionnameA):

<table>
<thead>
<tr>
<th>Task/Decname</th>
<th>BRANCH,1:</th>
<th>IF, Preconditions met, Task/Decname:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>IF, NOT Precond met, Task/DecnameWT;</td>
</tr>
<tr>
<td>Task/DecnameWT</td>
<td>DELAY:</td>
<td>Duration of test cycle;</td>
</tr>
<tr>
<td>Task/DecnameA</td>
<td>BRANCH,1:</td>
<td>ALWAYS, Task/Dec.name;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rest of code for task or decision</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(see above)</td>
</tr>
</tbody>
</table>

The second way of implementing pre- and postconditions is by using the WAIT and SIGNAL construct. The above can be replaced by the following:

<table>
<thead>
<tr>
<th>Task/Dec.name</th>
<th>WAIT:</th>
<th>Precondition expression (integer);</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task/Dec.nameA</td>
<td>.......</td>
<td>Rest of code for task or decision</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(see previous section)</td>
</tr>
</tbody>
</table>

Any time a condition is met, the WAIT statements should be signaled by the following statement:

Somewhereincode SIGNAL: Precondition expression (integer);

We prefer the first construct because we don't need to use integers here and because the code for testing a precondition is only located at one position in the simulation model (while the SIGNAL statement may be far away from the WAIT statement, making the model quite complex and hard to debug).
Designing Organizational Coordination

The animation model

The first aspect model of the empirical model we discuss is the animation model. Vreede (1995) points out that most simulation environments provide animation facilities. An animation is the graphical representation of the empirical model, focusing specifically on the dynamics as modeled in the dynamic model. The main purpose of animation is to facilitate communication between the participants in a problem-solving process. We distinguish between problem owners, decision makers and problem solvers (see also Vreede and Verbraeck 1995).

Animation can be very useful for the problem solver as a verification instrument. When more than one problem solver is involved, the animation can serve as a communication instrument between the problem solvers, aiming at a joint and matching mental model of the problem situation. When the animation is used to communicate between problem solvers and problem owners, it provides an instrument for structural validation. The animation contributes to the acceptance of the empirical model as a representation of reality.

Finally, the animation model can be used as a presentation tool, in order to communicate with decision makers. The animation may highlight specific elements of the empirical model, displaying bottlenecks, throughput problems, and under- or overcapacity. However, the animation may never serve as a decision support tool in itself as it is incapable of showing statistically accurate model behavior. The experiment model, discussed in the next section, is a better basis for decision making.

The experiment model

Simulation experiments are run in order to obtain quantitative material concerning the situation under consideration. The second aspect model of the empirical model is called experiment model and serves the purpose of quantitative analysis. For this analysis to take place, the following elements must be present:
A methodology for coordination design

- One or more treatments.
- The collection of output data for each treatment.
- A specification of the analysis rules that are used to interpret the output data.

The experiment model consists of a collection of analyses ranging from the presentation of the output data itself (in graphs, charts or tables) to more sophisticated methods like confidence intervals, variance analysis (ANOVA) and regression analysis. The most common variables that are analyzed from a simulation experiment are throughput times, service times, queue lengths, capacity utilization and number of entities in the model. Examples are given in the case studies in chapter 6.

4.5 Management approach

A management approach supports a methodology in terms of checkpoints, documentation, decisions and time management. Although, for our methodology, any common approach can be used (see e.g. Turner, 1980 and Turner et al., 1989), we like to point out some specific details.

In the discussion of working method, we have argued that the approach is not linear in terms of steps to be carried out. In using the methodology and managing subsequent projects, we advise an incremental approach, starting from a ‘middle out’ point of view. This management philosophy results in the following guidelines:

- Delineate the problem situation as narrowly as possible, but in such a way as to ensure capturing the core of the problem.
- Extend problem boundaries only when absolutely necessary.
- Keep project teams small, extend with expertise when required.
- Report on essentials only, but do not leave out important details.

In section 4.3.2. we have defined a treatment as a specification of input data, a collection of input data, initialization conditions, run control conditions and the specification of output data.
Designing Organizational Coordination

As regards the working method, we mentioned the reports which result from each steps taken in the project. We repeat them here, and also state the purpose of the reports and the typical decisions that may be taken after the different modeling steps.

Conceptualization report

- **Goal**: to provide insight into the boundaries of the problem situation and the concepts that are going to be used to attack this situation in terms of object classes, network models, process models and actor models. The report also contains a problem statement.
- **Decisions**: if and when the participants agree on the conceptual models as displayed in the conceptualization report, one can decide to construct an empirical model of the current situation.
- **Time**: the conceptualization phase takes about 30 to 40 percent of total project time

Specification report

- **Goal**: to provide insight into the construction, verification and validation of the empirical model. The report contains information about the simulation model itself, but also about the animation model and the experiment model of the problem situation.
- **Decisions**: based on the results displayed by the experiment model, one can decide to generate alternatives for changing the situation.
- **Time**: the specification phase takes about 20 to 30 percent of total project time

Alternatives report

- **Goal**: to provide insight into possible solutions to the problem and the effects of each analyzed solution. The report contains descriptions of each solution in terms of the concepts of the conceptual model, and the experiment models of each solution. The report should also contain a discussion about the consistency check.
A methodology for coordination design

- **Decisions**: based on the results displayed by the experiment models of the alternatives, one can decide to implement one or more alternatives.
- **Time**: the alternative generation phase takes about 30 to 40 percent of total project time
Designing Organizational Coordination
5. Support tools for coordination design

'New technologies that enhance the ability to create and understand information have always led to dramatic changes in civilization.'
- Al Gore

5.1 Introduction

In chapter 4 we introduced a methodology for coordination design based on the dynamic modeling approach. The methodology comprises conceptual and empirical model types, to be constructed during a problem solving process. This chapter presents two software tools that support the construction of these models. The first tool supports the user in making the conceptual models as put forward in chapter 4. It is discussed in section 5.2. The second tool supports the specification phase and is discussed in section 5.3.

Many methodologies, especially in the field of information systems development, are supported by software packages for generating models. Support ranges from simple drawing packages to 4th generation modeling and programming workbenches.

During the development of our methodology, we have encountered several packages that support conceptualization or specification. We have not found any software tools that support both conceptualization and specification in the way required by our methodology.

1For conceptualization support, we encountered model construction tools like ABC flowchart, CorelFlow and VISIO. For specification support we investigated the Siman/Cinema and ARENA simulation environments. We have already discussed specification with Siman/Cinema in chapter 4.
Designing Organizational Coordination

During the project we encountered a model construction tool called VISIO® (see Shapeware, 1994) and a simulation workbench called ARENA™ (see Pegden et al., 1995) that both allow the construction of methodology-specific 'templates'. Such a template provides means to construct a conceptual model in terms of the elements of chapter 4, and subsequently a simulation model based on these model elements.

In section 5.2 we develop the requirements for a conceptualization support tool and give an example of a conceptualization template in VISIO. In section 5.3 we give the requirements for a tool that takes a more integrated approach towards specification and describe the specification template that was built in ARENA.

5.2 Conceptualization support

In section 4.3.1 we stated that our working method for conceptualization consists of the following four steps:

1. Identify the object classes.
2. Construct the conceptual model.
3. Formulate a problem statement.
4. Write the conceptualization report.

The second, model construction activity is the central issue of conceptualization. In this activity, the network models, process models and actor models as discussed in section 4.4.2 are constructed. The construction of these models can be facilitated by computer software. The following section gives the requirements of such software, based on the object class definitions and model types of section 4.4.2.

5.2.1 Requirements

Bockstael and Meinsma (1994) identify two main groups that will use a 'conceptualization support environment'. They identify a passive user group, consisting of problem owners and other project partici-
Support tools for coordination design

pants, and a more active group, the modelers themselves. The passive users need a recognizable, clear and easy to understand representation of their problem situation and therefore depend heavily on the graphic capabilities of the support system (Bockstael and Meinsma, 1994). The active group, on the other hand, should be able to create an environment for communication between the parties involved in conceptualization (Meinsma and Sol, 1993). To support these two groups in conceptual modeling, the system supporting our methodology should meet the following requirements:

1. It should be able to construct network models, process models and actor models.
2. It should be able to work with multiple levels of abstraction (e.g. tasks and subtasks) and from multiple perspectives (e.g. macro, meso and micro).
3. The models should be of high graphical quality in order to assure good communication between modeler and problem-owner.
4. In order to reduce the time of actually drawing the model, the software should include all modern graphical modeling features like cutting, pasting, copying, rotating, flipping, snapping, gluing, aligning and distributing model elements.
5. The models should be exportable to a word processor so as to be able to show them in conceptualization reports.

As Vreede (1995) points out, it can also be very useful to have a package that supports group modeling, for example in a Group Decision Room. He discusses a package called TeamGraphics™ that has group functionality but lacks the high graphic quality and special features of products like ABC FlowCharter™, CorelFLOW™, and VISIO™. However, in the near future, we expect the stand alone software packages to incorporate group functionality and group modeling packages to increase their graphical standards.

For the conceptualization activities carried out in this research project, we chose VISIO as a tool. The main reason for this choice is the possibility of developing a template within the VISIO software, consisting of the model elements as defined in chapter 4.
Designing Organizational Coordination

Furthermore, VISIO allows multiple levels of abstraction, has high graphical quality and incorporates the modeling features mentioned above. The modeling results can be transferred to word processors (and other applications) that operate under the MS-Windows operating system. The next section gives an example of conceptualization support, using this software (see also Eijck and Sol 1996).

5.2.2 An example: the DOC/C template in VISIO 3.0

VISIO version 3.0 for windows offers the possibility to develop your own library of model elements and group them in a ‘template’. For each of the model elements as expressed in our methodology, we developed a ‘shape’ in the template. This results in a small toolbox from which shapes can be dragged on a drawing pad on the screen and from which the models as proposed in chapter 4 can be constructed.

![Figure 5.1: Screendump of the DOC/C template](image-url)
Support tools for coordination design

The shapes all have connection points to which arrows and lines can be glued. Text can be added to several area's of a shape (e.g. one can type the actor name and the process name in a process block). Shapes can be aligned to each other and to a grid, and can be copied, moved and deleted. To each shape, one can attach additional information (for example object definitions or process characteristics). Parts of drawings or complete models can be copied into other Windows applications and exported to other graphical formats. It is even possible to start an application by double clicking on a shape. An example of the template and a model in construction is given in figure 5.1. The template is shown on the left side of the screen and contains the icons as displayed in figures 4.4, 4.5 and 4.6, along with some additional features for connecting the model elements. The template is called DOC/C (Designing Organizational Coordination / Conceptualization).

5.3 Specification support

An example of specification support is given in chapter 4, where we presented a translation of our conceptual modeling approach into simulation code of the Siman IV simulation environment. For a more flexible support of our methodology, we developed a template in the ARENA simulation environment which provides a construction tool that conforms to our way of thinking and our modeling approach.

5.3.1 The ARENA simulation environment

The ARENA simulation environment allows a modeler to construct a simulation model from pre-defined building blocks. The complexity of these building blocks ranges from simple statements of the simulation language to very complex constructs like servers, conveyor belts and automatically guided vehicles. The building blocks are presented as modules in a template, much like the template we discussed in the previous section. A simulation model is then constructed by dragging the modules from the template onto the modeling screen, connecting them and entering the right data in the user interface of the module.
Designing Organizational Coordination

With the ARENA professional edition, it is also possible to define your own specific modules in a template that can be attached to the modeling screen. This user defined template can be used in conjunction with other predefined templates. The so-called ‘template builder’ allows us to develop a modeling template that stays very close to the modeling concepts as discussed in chapter 4. It is possible to develop a module for each object as defined in our methodology\(^2\). The user interface and simulation logic of the modules are based on attributes and actions of the objects of chapter 4.

The notion of pre-defined modules also conforms to the distinction, mentioned before, between object class (i.e. an ‘empty’ module consisting of logic and ‘empty’ attributes) and object instance (i.e. a copy of a module, containing specific data as attribute values). An example of this is the server module of the standard ARENA template. When selected, the server is empty and represents the object class server. The server is instantiated by giving it a name (e.g. Employee_01) and by entering empirical data into the input fields of its user interface.

Based on the concepts of chapter 4, we developed a template that supports the specification activity of our methodology. In the next subsection we derive the requirements for specification support from the concepts as presented in chapter 4. In section 5.3.3, we present the functional specifications of each of the modules in the template, based on the requirements.

The template is a working prototype and was built with the help of two undergraduate students (see Bouwman and Kuijlaars, 1995).

5.3.2 Requirements

The requirements for a specification support tool can be categorized according to the steps that are carried out in the specification process. We discuss the support for each step in the specification process.

\(^2\)For the item object, there is no need to build a module. It is however necessary to build a module for creating items and assigning initial attribute values to them.
Specify output variables

According to our discussion of performance indicators in chapter 3, it should be possible to specify the following types of output variables:

- Throughput times between specified time stamps or events.
- Process times of specific objects.
- Queue lengths.
- Capacity utilization.
- Values of attributes and global variables.

It should be possible to measure the average value, the variation, the maximum value, the minimum value, the end-value and a frequency distribution of these output variables.

Specify model reductions

Sol (1982) describes the reduction problem as: ‘... what attribute parts and what action parts of which entities (objects, DvE) are to be described to what degree of detail.’ This means that reduction operators refine and extend the attribute and action parts of the objects as identified in the conceptual model. As to the attribute parts, specification support should allow reduction operators of the following kinds within the resulting simulation model:

- Operators that replace deterministic attribute values by stochastic values (e.g. the duration of a specific task can be modeled by a normal distribution).
- Operators that coarse the range set of attribute values.

As to the action part, specification support should allow reduction operators that replace repeating tasks or event sequences by functions or procedures that can be invoked by one or more objects. Furthermore, the specification method should allow the reduction of relevant objects that lie outside the system under consideration. This can be done, for example, by means of item generators and item disposers.
Designing Organizational Coordination

We do not find it imperative that model reductions are specified as a separate list within the simulation model, but they should be part of the specification report.

Collect input data

We do not find it useful to provide support for the activity of collecting input data in our specification support tool.

Construct simulation model

The main objective of the specification support tool is to help the modeler in constructing a simulation model that is based on the conceptual model. The support tool should comprise building blocks that are based on the conceptual building blocks of chapter 4. The specification support tool should provide constructs that specify the attribute and action parts of the objects as identified in the conceptual model, taking the reduction operators as specified above into account.

The specification support tool should provide an environment in which simulation models can be constructed in much the same way as the conceptual models are constructed: placing individual objects on the screen, and connecting them in the right way. Furthermore, the specification environment should not restrict the user merely to the use of the predefined building blocks. It should, for example, be possible to use simulation constructs already provided by other templates or to extend the building blocks for personal use.

Construct animation

Each building block should have an animation part that can be specified by the modeler. This animation component should include a visualization of the behavior of the object itself, as well as the possible handling of items and routing of items to another object (e.g. the animation should display the functioning of repositories as well as the reception, storage and routing of items).
Support tools for coordination design

Specify initial treatment

Within the simulation model it should be possible to specify the input data, the initialization conditions and the run control conditions. Together with the earlier mentioned specification of output data, the model can then be used for verification and validation purposes. Furthermore, it should be possible to expose the simulation model to different treatments and perform sensitivity analyses with it.

Write specification report

It should be possible to incorporate parts of the simulation model and the animation facilities in the specification report. If this can be done by cut and paste functionality of the operating system's user interface, there is no need for specialized functionality within the prototype.

5.3.3 Functional specifications

In our requirements, we stated that the specification support tool should provide building blocks for constructing a simulation model. These building blocks should be natural extensions of the conceptual building blocks of chapter 4. The core of the conceptual models are the actors who perform tasks and the repositories that store items. So the basic structure of the conceptual model, is reflected in the network model. In our view, therefore, the construction of a simulation model should, start with a specification of the network model as defined in chapter 4. This involves the following building blocks:

- The actor.
- The repository.
- The link.
- The item.

Although passive objects, the items flow through the organizational network and, in a way, keep the simulation model 'alive'.

123
Describing Organizational Coordination

To be able to specify the activities an actor performs, we developed the following building blocks:

- The activity (for compound tasks).
- The task (for simple, or subtasks).
- The decision.

At the system boundaries, items are generated and disposed of. The generation and disposal of items are in fact reduction operators. We need two constructs to achieve these reductions in a simulation model:

- The item generator.
- The item disposer.

When the above nine constructs are specified in terms of ARENA modules, the elements of our conceptual modeling approach appear in the template as individual building blocks, from which a simulation model can be constructed. In this section, we describe the functional specifications of each of the nine constructs. Each construct is developed as a module in the template. From now on we denote our modeling constructs as modules, as they will be implemented as such in ARENA. The collection of modules that compose the specification support tool, is referred to as the template.

The functional specifications are numbered and start with the first letter of the module name, followed by the type of specification. Specifications relate to the basic functions that a module performs, (denoted by an F), the different ways in which a module can be used apart from its basic functionality, called options (denoted by an O), the way in which performance indicators related to the module are specified, called statistics (denoted by an S) and the way in which the module appears on the screen when the simulation model is running, i.e. the animation (denoted by an A). The tables in the following subsections give the functional specifications of each module.

124
Support tools for coordination design

The actor

The actor is the central object in the methodology. The actor is active in the sense that this object can carry out tasks and take decisions. However, these tasks and decisions can be the same for different actors, so within the actor module there should only be a reference to the tasks he or she may carry out and the conditions under which these tasks are executed. The actor module should be able to receive and send items and also fetch items from repositories when necessary. Table 5.1 shows the specifications of the actor module.

<table>
<thead>
<tr>
<th>Functions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AF.1</td>
<td>Taking items from one or more links, thereby seizing a resource.</td>
</tr>
<tr>
<td>AF.2</td>
<td>Giving items access to activities, keeping the resource seized.</td>
</tr>
<tr>
<td>AF.3</td>
<td>Sending items to one or more links that take items to their destination.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Options</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AO.1</td>
<td>Selecting between activities, including externally defined activities.</td>
</tr>
<tr>
<td>AO.2</td>
<td>Generating copies of items.</td>
</tr>
<tr>
<td>AO.3</td>
<td>Handling prioritized items.</td>
</tr>
<tr>
<td>AO.4</td>
<td>Accessing repositories.</td>
</tr>
<tr>
<td>AO.5</td>
<td>Temporarily changing the capacity (breaks etc.).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statistics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AS.1</td>
<td>Keeping track of capacity utilization.</td>
</tr>
<tr>
<td>AS.2</td>
<td>Keeping track of the average number of items in queue.</td>
</tr>
<tr>
<td>AS.3</td>
<td>Keeping track of maximum and minimum number of items in queue.</td>
</tr>
<tr>
<td>AS.4</td>
<td>Keeping track of the average throughput time of items with actor.</td>
</tr>
<tr>
<td>AS.5</td>
<td>Keeping track of the minimum and maximum throughput time of items.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Animation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AA.1</td>
<td>Displaying the actor’s status.</td>
</tr>
<tr>
<td>AA.2</td>
<td>Displaying the queue in front of the actor.</td>
</tr>
<tr>
<td>AA.3</td>
<td>Changing the appearance of items.</td>
</tr>
</tbody>
</table>

Table 5.1: Specifications of the actor

The repository

The repository serves as a temporary storage of items, and can be accessed by actors. It should be able to receive and send items upon request. We distinguish between simple repositories that contain identical items, and just send the number of items requested randomly from
Designing Organizational Coordination

their internal queue, and more complex repositories that contain specific items, which can be requested individually, based on their attribute values. Table 5.2 shows the requirements of the repository module.

<table>
<thead>
<tr>
<th>Functions</th>
<th>Options</th>
<th>Statistics</th>
<th>Animation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF.1</td>
<td>RO.1</td>
<td>RS.1</td>
<td>RA.1</td>
</tr>
<tr>
<td>RF.2</td>
<td></td>
<td>RS.2</td>
<td>RA.2</td>
</tr>
<tr>
<td>RF.3</td>
<td></td>
<td>RS.3</td>
<td>RA.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RS.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RS.5</td>
<td></td>
</tr>
</tbody>
</table>

Taking items from one or more links.
Storing an item.
Sending items to one or more links that take items to their destination.
Taking requests from actors for specific items.
Keeping track of the average number of items in the repository.
Keeping track of maximum and minimum number of items.
Keeping track of the number of requests for items.
Keeping track of the average time an item stays in the repository.
Keeping track of the minimum and maximum time an item stays.
Displaying the repository's status
Displaying the queue in front of the repository.
Changing the appearance of items when stored or sent away.

Table 5.2: Specifications of the repository

The Link

The link actually is a very simple module and just transfers items between actors and repositories or between actors. Table 5.3 shows its requirements. For the prototype, the link has no options. In later versions one could think of options that generate link failures.

<table>
<thead>
<tr>
<th>Functions</th>
<th>Statistics</th>
<th>Animation</th>
</tr>
</thead>
<tbody>
<tr>
<td>LF.1</td>
<td>LS.1</td>
<td>LA.1</td>
</tr>
</tbody>
</table>

Transferring one or more items.
Keeping track of the number of items transferred by the link.
Keeping track of the capacity utilization of the link.
Showing the transfer of items.

Table 5.3: Specifications of the link
Support tools for coordination design

The item

The item is a passive object and performs no actions of its own. Therefore, it has no functions or options. It has statistics requirements and animation features only. Table 5.4 shows these requirements.

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS.1</td>
<td>Average throughput time of an item between preset events.</td>
</tr>
<tr>
<td>IS.2</td>
<td>Maximum and minimum throughput time of items.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Animation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IA.1</td>
<td>A visualization of the movement of an item between active objects.</td>
</tr>
<tr>
<td>IA.2</td>
<td>A visualization of the item itself and the changes it may undergo.</td>
</tr>
</tbody>
</table>

**Table 5.4: Specifications of the item**

The activity

When it is compound, the activity module encapsulates several tasks and decisions that an actor may carry out. When it is simple, it assumes the functionality of a task. In an actor module there may be references to several activity modules, each reference is then related to specific conditions. An activity can be used by one actor or by several actors at the same time. Because the letter A was reserved for the actor, the requirements code of the activity module starts with a Y.

<table>
<thead>
<tr>
<th>Functions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YF.1</td>
<td>When compound, give item access to underlying tasks and decisions.</td>
</tr>
<tr>
<td>YF.2</td>
<td>When simple, assume functionality of the task (see below)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YO.1</td>
<td>An activity can be either compound or simple</td>
</tr>
</tbody>
</table>

**Table 5.5: Specifications of the activity**

The task

Tasks have the property that they transfer specified pre-conditions into specified post-conditions by changing attribute values of the item that is being processed or the values of global variables. Meanwhile they keep the concerned actor busy during the specified time delay.
Designing Organizational Coordination

<table>
<thead>
<tr>
<th>Functions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF.1</td>
<td>Transfer specified pre-conditions into specified post-conditions</td>
</tr>
<tr>
<td>TF.2</td>
<td>Keep the concerned actor busy during the specified time delay</td>
</tr>
</tbody>
</table>

*Table 5.6: Specifications of the task*

The decision

Decisions route an item through a sequence of tasks, based on a condition. This condition can relate to the item’s attribute values or to a probability distribution. The decision includes the option to make copies of items.

<table>
<thead>
<tr>
<th>Functions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DF.1</td>
<td>Route items to a next task or decision, based on a condition</td>
</tr>
<tr>
<td>DF.2</td>
<td>Keep the involved actor busy during the specified time delay</td>
</tr>
<tr>
<td>DO.1</td>
<td>Generate one or more copies of an item</td>
</tr>
</tbody>
</table>

*Table 5.7: Specifications of the decision*

The work generator

The work generator establishes a part of the system boundary. It generates items and initializes their attribute values. One or more work generators can be used in a simulation model.

<table>
<thead>
<tr>
<th>Functions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WGF.1</td>
<td>Generating items.</td>
</tr>
<tr>
<td>WGF.2</td>
<td>Assigning values to attributes of items.</td>
</tr>
<tr>
<td>WGF.3</td>
<td>Sending items to one or more links that take them to their destination.</td>
</tr>
<tr>
<td>Options</td>
<td></td>
</tr>
<tr>
<td>WGO.1</td>
<td>Generation of items according to a probability distribution.</td>
</tr>
<tr>
<td>WGO.2</td>
<td>Generation of items after a keystroke.</td>
</tr>
<tr>
<td>Statistics</td>
<td></td>
</tr>
<tr>
<td>WGS.1</td>
<td>Keeping track of the total number of items generated.</td>
</tr>
<tr>
<td>Animation</td>
<td></td>
</tr>
<tr>
<td>WGA.1</td>
<td>Show the generation of items.</td>
</tr>
</tbody>
</table>

*Table 5.8: Specifications of the work generator*
Support tools for coordination design

After an item is generated by a work generator, it is sent to a first destination (i.e. an actor or a repository) across a link. Items can be generated according to a probability distribution, or after a keystroke. The keystroke generation option is created for verification and gaming purposes.

The item disposer

The second module that establishes part of the system boundary is the item disposer. It disposes items and keeps track of the number of items disposed.

<table>
<thead>
<tr>
<th>Functions</th>
<th>Statistics</th>
<th>Animation</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDF 1</td>
<td>IDS.1</td>
<td>IDA.1</td>
</tr>
<tr>
<td>Generating items.</td>
<td>Keeping track of the total number of items disposed.</td>
<td>Show the disposal of items.</td>
</tr>
</tbody>
</table>

**Table 5.9: Specifications of the item disposer**

The user interface

The user interface of a module involves the following three elements:

- The dialog with the user.
- The icon of each module.
- The names of each module.

Each of the modules will be presented to the user as a dialog box in which specific values of the elements can be entered. These dialog boxes present entries for all values needed by the module to function properly. The boxes use textual entries, selection buttons (also called 'radio buttons') and check boxes. Based on selections made, other entries may appear and even new dialog boxes can show up. In the next section we present some of the dialog boxes.
Designing Organizational Coordination

5.3.4 A prototype: the DOC/S template in ARENA 1.23 PE

In this section we present the prototype of the simulation template we constructed based on the above requirements and functional specifications. The prototype is implemented in version 1.23 of the ARENA™ Professional Edition (see Pegden et al., 1995). The prototype is called the DOC/S tool (Designing Organizational Coordination / Specification) and consists of the following seven modules, grouped in a template that can be attached to the ARENA simulation environment:

- Actor module.
- Repository module.
- Activity module.
- Task module.
- Decision module.
- Work generator module.
- Exit point (item disposer) module.

Next to these modules, the regular ARENA modules can also be used. We defined no module for the link object. The begin- and end points of a link are implemented as stations within the actor and the repository. The connection between these begin- and end points is established by taking a route from the animation template which is part of the simulation environment.

The item object is not implemented as a module. Items and their attributes are provided by the simulation engine itself, and are generated by the work generator module.

The activity module does not need to be attached to one single actor. This provides a way of using one and the same activity definition for several actors. Each actor then has a reference to that specific activity. Several activities may be specified within one actor. Each activity is executed according to the condition specified with the reference in the actor. When more than one condition is valid, the related activities are executed sequentially, i.e. one actor cannot perform tasks in parallel.
Support tools for coordination design

The DOC/S template and the ANIMATION template are the only templates that are required for simple models. For more complex models, it is possible to attach other templates, provided by the ARENA environment, that contain lower-level simulation statements or higher-level building blocks.

In this thesis, we do not present the implementation of all modules of the template. We give an explanation of how the activity module is implemented in terms of module structure, user interface and source code. For a full explanation of the technical implementation of the template, we refer to Bouwman and Kuijlaars (1995).

For the implementation of each module in the template, five steps have to be carried out:

- The construction of a user interface. The functionality as specified in the functional specifications can be implemented by specifying the windows and window entries that should be presented to the user. A window may contain sub-windows, text or numerical entries, selection lists and radio- and check buttons.
- The implementation of the module's simulation logic. For the logic of a module, the language constructs of the simulation engine can be used. Within the simulation statements, references to entries in the user interface can be specified.
- The specification of the module's options (called switches). Specific options for a module can be specified in the user interface by means of check boxes and radio buttons. The Boolean variables that contain the options that are selected in a module have to be specified separately. These Boolean variables can also be used to switch on or switch off specific parts of the simulation logic.
- The construction of the module's animation features. Each module has a basic appearance for animation purposes, which can be changed by the user.
- The construction of the module's icon. To visualize the module in the template window, left of the workspace, an icon should be specified.
Designing Organizational Coordination

In this section, we show the construction of the *activity module*, following the above five steps. Furthermore, we demonstrate the prototype by showing the specification of a small example, using the template. We choose the activity module because it is relatively simple and straightforward.

**The activity module**

The construction of the module's user interface requires the specification of the names, types and hierarchy of the input fields. Figure 5.2 displays the *operand window* of the template development environment, in which the input fields of the user interface can be specified and mutually connected.

![Activity Module Diagram](image)

**Figure 5.2: Operand window of the activity module**

The addition of the letters SW and a name between brackets means that the input field will only appear in the dialog box when the related Boolean expression yields the true value. In case of the activity module, the user should choose between a single task or a more complex structure of tasks and decisions. In the latter case, two entries of the dialog box are switched on and two entries are switched off. Table 5.10 shows the names and types of the input fields given in figure 5.2.
Support tools for coordination design

<table>
<thead>
<tr>
<th>Input field</th>
<th>Field type</th>
<th>Displayed when activity is</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Name</td>
<td>Text</td>
<td>single or compound</td>
</tr>
<tr>
<td>2: Select type</td>
<td>Radio button</td>
<td>single or compound</td>
</tr>
<tr>
<td>3: TASK. Duration</td>
<td>Numerical Expression</td>
<td>single</td>
</tr>
<tr>
<td>4: Attribute Assignments</td>
<td>(Sub)list of assignments</td>
<td>single</td>
</tr>
<tr>
<td>5: Structure_start</td>
<td>Start address</td>
<td>compound</td>
</tr>
<tr>
<td>6: Structure_end</td>
<td>End address</td>
<td>compound</td>
</tr>
</tbody>
</table>

*Table 5.10: Fields and field types of the activity module*

The fields as structured in the operand window lead to the two possible dialog boxes shown in figure 5.3.

*Figure 5.3: Dialog boxes for simple and compound tasks*

The *simulation logic* of the activity can be divided into two parts. One part is the logic for simple tasks, consisting of a delay and a set of attribute assignments. The other part is the logic for compound activities, consisting of references to the first and last task of the compound activity. The logic is constructed in the *logic window*. The logic for the activity module is given in figure 5.4. The figure shows a dotted box with the text ‘tasks’ in it, referring to the tasks and decisions that are entered in the dialog box as ‘structure start’ and ‘structure end’.

133
Designing Organizational Coordination

![Diagram](image)

**Figure 5.4: Logic window of the activity module**

The two lines departing from the station denoted as ‘TASK_Name’ are the two paths that relate to the activity being either compound (the upper part) or simple (the lower part).

The *options* are constructed in a so called ‘Switches window’, in which special Boolean expressions can be entered that influence the dialog boxes, the logic, the actual appearance and the animation of the module. To influence a piece of code or a field in a dialog box the expression, referred to as a *switch*, must be connected to that piece of code or to that dialog box. The activity module has two such expressions: SW_not_compound, denoting the fact that an activity is simple, and SW_compound, denoting the fact that an activity is compound. The switches are each other’s inverse.

The *appearance* of a module, including its animation features are constructed in the *user view* window of the ARENA template builder. An activity is denoted as an A, bounded by a rectangle, and has no specific animation features. When an activity is compound, two connection points appear. Tasks and decisions can be attached between these points. The two possibilities of the module’s appearance are shown in figure 5.5.

134
Finally, the icon that is displayed to denote the activity module in the ARENA modeling environment is also an A, bounded by a rectangle. The icon is displayed in figure 5.7, where we give an impression of the template in action. This concludes the discussion about the steps that were carried out to construct the activity module. The template consists of seven modules that were all created likewise. A complete specification of the implementation can be found in Bouwman and Kuijlaars (1995).

An example

We use an example of three service engineers who request service assignments from a database by means of electronic communication and carry out these assignments accordingly. A secretary answers phone calls from clients (clients are modeled as a work generator), determines the problem and assigns a service engineer to the job. The job is put in a database for later (electronic) retrieval by the engineer. When an engineer has finished his assignment, he sends a report to the service desk which is modeled as an exit point.

The network model of this example is given in figure 5.6 and the implementation using the template is shown in the scrrendump of figure 5.7. One of the benefits of the template is the possibility to arrive at a simulation model that strongly resembles the conceptual model.
Designing Organizational Coordination

![Network Model Diagram](image)

**Figure 5.6:** Network model of the service engineers example

![Screenshot](image)

**Figure 5.7:** Screendump of the template in action
In modeling the activities of actors, too, the simulation model stays close to the conceptual model. To show this resemblance between the conceptual model and simulation model, we displayed the actor model of an engineer and its specification in figure 5.8. The activity 'fix system' is specified as a compound activity. Within each service engineer, references to this and other activities (like the retrieval of new jobs) are incorporated. The engineers alternate between the 'retrieve job' and 'fix system' activities when they are active.

![Diagram](image)

**Figure 5.8: An actor model and its specification in the template**

The actor models as developed in a conceptual model can be converted into activities and underlying tasks and decisions. With the template, it is possible to model an actor model once and apply it to multiple actors. This greatly facilitates the specification of groups of actors who carry out the same task. In the example of the service engineers, this means that the actor model of an engineer is specified only once. Within each instance of an engineer, there is a reference to its actor model.

137
Designing Organizational Coordination

5.4 Conclusions

The description of the conceptualization tool and the specification tool concludes the presentation of our design system. Already, some preliminary conclusions can be drawn ahead of the practical uses of the design system in the next section. Below we present our findings with respect to the use of templates in general, the conceptualization support tool, and the specification support tool.

5.4.1 The use of templates

For both tools presented in this chapter, we were able to use software that allows the construction of a template that can be attached to it, in order to support our methodology. The building of the tools as discussed in this chapter was greatly facilitated by the concept of templates for the following reasons:

- A template can be tailor-made and fully adjusted to needs. Adjustments can be made quickly, even within the time span of a single project, making a template not only methodology-specific, but even project-specific.
- The construction of a template can be done much faster than the development of a complete software package. There is no risk of reinventing the wheel because the basic functionality is already programmed in the environment for which the template is built.
- The modeling environment itself is not restricted to one specific methodology. When necessary, concepts from other methodologies can be borrowed and incorporated in the template.

A drawback of the template concept is the possibility that different users in the same team can adapt the template to their own habits, resulting in a loss of generality in models of different team members. However, we find that this problem is solvable and that the template concept can be extremely useful for problem-solving support tools.
Support tools for coordination design

5.4.2 Conceptualization support

After developing it, the DOC/C conceptualization support tool was used by the author and several modelers who had to make the conceptual models as discussed in chapter 4. The general opinion was that the tool was flexible and fast. Models can be made much faster than by using traditional drawing packages. Models can also be arranged faster into a good layout by means of the built-in arrangement tools. Adjustment and adaptation of the models as well as the template itself are easy and flexible. Incorporation of models made with the template into word-processor documents proved to be no problem.

The models made with the template were judged by modelers and people from the problem domains as clear, easy to read and understand and of high graphical quality.

All conceptual models shown in the case studies of chapter 6 and 7 were made using the template, so as to test the latter thoroughly. The results of this testing are discussed in chapter 8.

5.4.3 Specification support

Due to the fact that the ARENA template development package was not available when this research project started and due to the time that template development takes, the DOC/S specification support tool is tested only in the case studies of chapter 7. We refer to chapter 8 for a final evaluation.

Not all of the functional requirements were eventually implemented. The statistics functionality is already strongly supported by generic ARENA building blocks, so the implementation of statistic support was not necessary in a first prototype. Table 5.11 shows which elements of the functional specification are implemented in the template. A ✓ represents the fact that an element of the functional specifications was implemented and an @ represents the fact that an element of the functional specifications is already provided by ARENA.
Designing Organizational Coordination

Due to time constraints, we did not yet implement two of the options of the actor. These are the handling of prioritized items (AO.4) and temporary capacity changes (AO.5). Both options should be implemented in the next version of the template.

<table>
<thead>
<tr>
<th>Functions</th>
<th>Options</th>
<th>Statistics</th>
<th>Animation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actor</strong></td>
<td>AF.1</td>
<td>✓ AO.1</td>
<td>✓ AS.1</td>
</tr>
<tr>
<td></td>
<td>AF.2</td>
<td>✓ AO.2</td>
<td>✓ AS.2</td>
</tr>
<tr>
<td></td>
<td>AF.3</td>
<td>✓ AO.3</td>
<td>✓ AS.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>✓ AO.4</td>
<td>✓ AS.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>✓ AO.5</td>
<td>✓ AS.5</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>RF.1</td>
<td>✓ RO.1</td>
<td>✓ RS.1</td>
</tr>
<tr>
<td></td>
<td>RF.2</td>
<td>✓</td>
<td>✓ RS.2</td>
</tr>
<tr>
<td></td>
<td>RF.3</td>
<td>✓</td>
<td>✓ RS.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>✓ RS.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>✓ RS.5</td>
</tr>
<tr>
<td><strong>Link</strong></td>
<td>LF.1</td>
<td>@</td>
<td>LS.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LS.2</td>
</tr>
<tr>
<td><strong>Item</strong></td>
<td></td>
<td></td>
<td>IS.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IS.2</td>
</tr>
<tr>
<td><strong>Activity</strong></td>
<td>YF.1</td>
<td>✓ YO.1</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>YF.2</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td><strong>Task</strong></td>
<td>TF.1</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TF.2</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td><strong>Decision</strong></td>
<td>DF.1</td>
<td>✓ DO.1</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>DF.2</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td><strong>Work generator</strong></td>
<td>WGF.1</td>
<td>✓ WGO.1</td>
<td>✓ WGS.1</td>
</tr>
<tr>
<td></td>
<td>WGF.2</td>
<td>✓ WGO.2</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>WGF.3</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td><strong>Item disposer</strong></td>
<td>IDF.1</td>
<td>✓</td>
<td>IDS.1</td>
</tr>
</tbody>
</table>

*Table 5.11: Implementation of the specification tool*

### 5.4.4 Other support and integration

Next to the conceptualization and specification tools as described in this chapter, one can think of support tools for other specific activities of the methodology or other forms of conceptualization and specification. Several examples of these developments are (see also Sol, 1993 and Sol, 1995):
Support tools for coordination design

- Support for statistical analysis and validation. Support for these activities can be found in spreadsheet software or specific numerical and statistical analysis software like SPSS (Norusis, 1990).
- Support for experimental design (Meel, 1994).
- Support for group conceptualization and specification (Vreede 1995).

One could also argue for more integration between several support facilities. However, there is a risk that the software then results in more dedication to one single way of working and will therefore become less powerful. We argue for the development of more software packages that have template functionality, and for the development of more templates in those packages. One could think of interconnections between templates of different packages. With the advent of operating systems that provide standardized graphical user interfaces like MS-Windows for DOS, MS Windows-95 and X-Windows for UNIX, this will become more feasible. In this way, support can remain specialized for one specific activity and can still be powerful enough to interconnect with other support tools or other methodologies.
Designing Organizational Coordination
6. Applying the design system: two cases revisited

‘They were not the same eyes with which he had last looked out at this particular scene, and the brain which interpreted the images the eyes resolved was not the same brain. There had been no surgery involved, just the continual wrenching of experience.’
- Douglas Adams

6.1 Introduction

Chapter 6 and chapter 7 provide the ‘proof of the pudding’ and, according to the English, the proof of the pudding is in the eating. Therefore, these two chapters serve a five-course meal of case studies. Two of them the reader has already tasted (chapter 6), the other three are new (chapter 7).

The case studies are presented in the order in which they were carried out. The design system is applied to each of the case studies, and the results of this application are discussed. Some of the case studies highlight specific issues of the design system such as validation and simulation (case 1 and case 3), some of the case studies are relatively straightforward (case 1 and case 4), whereas case 5 is very complicated. All case studies concern administrative organizations in need of improvement of their business processes in terms of coordination. Of each case study, we describe the conceptualization phase, the specification phase, the verification and validation phase, the solution finding phase and we discuss the results.

As indicated, this chapter presents two case studies we have already discussed in chapter two: the Dutch Telecom case and the Insuraco case. Both cases were specifically used to obtain the requirements of the design system. In this chapter we will show that the design system meets these requirements.
6.2 Case 1: Dutch Telecom service engineers (revisited)

The first case study, it will be remembered, concerns the service engineers of a large telecommunication service company. Before we enter the realm of conceptualization, we shortly sketch the problem situation.

The case study concerns a department of service engineers of Dutch Telecom. In this department, 12 service engineers work on operational problems of the clients' switchboard systems. Problems are reported by clients, using a special number which is attached to a call center. Problems are subsequently assigned to a service engineer by a phone desk employee who gets the problems reported from the call center's computer. To pick up assignments, the engineers usually call in from a client and get their assignments from one of the three phone desk operators who decide on contract issues, location of client and engineer, expertise of the engineer and number of assignments already dispatched to an engineer. With this situation, there are several problems:

- Phone lines of the assignment phone desk are often busy, resulting in throughput times that are too high.
- Billing is sometimes a problem due to poor handwriting of the engineers.

6.2.1 Conceptualization

In this section we present the conceptualization according to the steps of chapter 4. We first identify the object classes, then construct the conceptual model and finally formulate a problem statement.

Identify object classes

The boundaries of the problem situation are set by the point where work assignments for the engineers come in and where the bills go out: the clients. Object classes can be identified by following an assignment from problem to bill. When a problem is reported, it is registered in the
Applying the design system: two cases revisited

call center’s computer, the MM007 system. From this computer it is assigned by a dispatcher to a service engineer. The service engineer finishes his job at the client and sends a notification to the billing department. The billing department makes a bill and sends it back to the client. From this specification, we can derive the following object classes that inherit from the actor class:

<table>
<thead>
<tr>
<th>Object Class</th>
<th>Attributes</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIENTS</td>
<td>name</td>
<td>generate assignments</td>
</tr>
<tr>
<td>BILLING EMPLOYEE</td>
<td>name</td>
<td></td>
</tr>
<tr>
<td>ENGINEER</td>
<td>name, expertise</td>
<td></td>
</tr>
<tr>
<td>DISPATCHER</td>
<td>name</td>
<td>get problem from MM007 system</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object Class</th>
<th>Attributes</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>drive to client</td>
</tr>
<tr>
<td></td>
<td></td>
<td>fix switch board</td>
</tr>
<tr>
<td></td>
<td></td>
<td>phone dispatcher</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>interpret clients problem</td>
</tr>
<tr>
<td></td>
<td></td>
<td>assign problem to engineer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dispatch assignment</td>
</tr>
</tbody>
</table>

Table 6.1: Object definitions of actor classes

There is only one repository object in this problem: the MM007 system. The MM007 system has no other actions or attributes than a standard repository as described in chapter 4. We identify one item object: the work assignment. During the processing of an assignment, it changes into a bill when the billing department has completed its work. The assignment has an attribute type of contract which determines the maximum response time as agreed with the client (which can be 8 hours or 18 hours). The object definitions of the assignment and the MM007 system are given below. With these object definitions, we can construct the conceptual model according to the steps of chapter 4.
Designing Organizational Coordination

**Table 6.2: Object definitions of repositories and items**

**Construct conceptual model**

Based on the problem description and the object definitions, we constructed the network model of figure 6.1. In fact, the handling of switchboard problems is a cycle that starts and ends at the client. The client describes his problem, which is entered in the MM007 system, from where it is picked up by a dispatcher and dispatched to a service engineer. The engineer handles the problem, writes a billing note, which is sent to the billing department and transformed into a bill. The bill is sent to the client.

*Figure 6.1: Network model of the engineers case*
Applying the design system: two cases revisited

The process of handling switchboard problems is outlined in the process model of figure 6.2.

Figure 6.2: Process model of the telecom case

We do not give a further specification of the actor models as the process model is already detailed enough in this respect.
Designing Organizational Coordination

Formulate problem statement

The problem of the situation as sketched by the case description and the conceptual model concerns both quantitative and qualitative issues: performance of the system in terms of throughput times and waiting times is too low and the accuracy of billing is below standards. A problem statement can, therefore, be expressed as follows:

How can the performance of switch-board fixing in terms of throughput times, waiting times and billing accuracy be improved to a situation where contract promises (response times) can be kept in each situation, dispatching takes less time and billing can be 100% accurate? What is the minimum number of engineers needed to achieve this goal?

6.2.2 Specification

In this section, we present the empirical (simulation) model of the present situation, based on the conceptual model of the previous section. We do this by describing each of the steps as discussed in section 4.3.2.

Output variables

Conceptualization has revealed two problems: throughput time and billing accuracy. We propose the following performance indicators for measuring the system:

- Throughput time of assignments related to contracts that demand a maximum of 8 hours response time.
- Throughput time of assignments related to contracts that demand a maximum of 18 hours response time.
- Number of assignments waiting for an engineer.
- Capacity utilization of the engineers and the dispatchers.
- Billing accuracy, as a percentage of assignments correctly billed.

Of these variables we calculate the minimum, maximum, average and variance values.
Model reductions

After a thorough analysis of the empirical situation, the following model reductions can be specified:

- The clients and the MM007 system will be taken together as one work generator.
- Work is generated by an exponential distribution, with a mean value that changes each hour. Mean values could be retrieved from the empirical situation for each hour of the day.
- We assume that the system starts empty and builds up a certain stock of assignments to be carried out. We conclude that the system is non-terminating\(^1\) (this is a valid assumption according to empirical observations).
- There are no fluctuations in assignment generation, other than the hourly differences in the mean number of assignments generated.
- Handling times and traveling times are modeled as triangular distributions\(^2\), specified by minimum value, mode and maximum value.

Input data

The input data for the simulation model is specified in table 6.3. The times mentioned are in minutes. The distribution between assignments that demand 8 hour response time and those that demand 18 hours response time is 14 percent to 86 percent.

Other input data for the model are the fact that engineers work for 8 hours per day and the number of engineers and dispatchers that are available.

---

\(^1\)A system is considered to be terminating if the events driving the system naturally cease at some point in time. Otherwise the system is non-terminating (Pegden et al. 1990).

\(^2\)A triangular distribution is a stochastic distribution of which the distribution function has the form of a triangle between a minimum and a maximum value. The height of the triangle is determined by the relative frequency at mode value. The distribution function has three parameters: minimum, mode and maximum.
Designing Organizational Coordination

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driving time to next client</td>
<td>tria(10,15,20)</td>
</tr>
<tr>
<td>Distribution 8 / 18 hour assignments</td>
<td>disc(0.14,8,1,18)</td>
</tr>
<tr>
<td>Time for dispatching an assignment</td>
<td>tria(5,6,5,8)</td>
</tr>
<tr>
<td>Time for trying a call by engineer</td>
<td>tria(1,1,5,2)</td>
</tr>
<tr>
<td>Time per bill (billing department)</td>
<td>tria(1,5,9)</td>
</tr>
<tr>
<td>Chance of bad handwriting</td>
<td>tria(0.05,0.10,0.15)</td>
</tr>
<tr>
<td>Time for fixing a switch-board</td>
<td>tria(25,30,35)</td>
</tr>
<tr>
<td>Time before engineer brings in notes for billing</td>
<td>960 (=2 days)</td>
</tr>
</tbody>
</table>

*Table 6.3: Input data for the simulation model*

Simulation model and animation

The simulation model was implemented in the Siman IV simulation language and the Cinema IV animation package. Upon completion of the template in ARENA, the model was built again with the template. An example of this is shown in figure 5.10, for a situation of 3 service engineers. An animation screen of the model made with the Siman environment is shown below, in figure 6.3. A specification of the model code can be found in van Eijck et al. (1992).

![Simulation model and animation](image)

*Figure 6.3: Screendump of the animation*
Applying the design system: two cases revisited

When the model is running, it skips the periods of time when no resource is active (the nights). After each working day of 8 hours, it skips to the morning of the next day, keeping run time as short as possible.

Initial treatment

The requirements for an initial treatment have partly been given above. We only need to establish the initialization conditions and the run control conditions. Initialization conditions determine the warm-up time for the simulation model. Run control conditions determine the run length and the number of replications. As the system is a non-terminating system, we have to determine the warm-up time and the run length. When this is established, we can derive the number of replications from a set of initial replications.

The warm-up time can be estimated from a plot of the total number of items in the system or the throughput time of items during the course of a simulation run. Figure 6.5 shows these variables for the simulation model of the case study. From this plot we conclude that a warm-up time of 3 days and a run length of 10 days will be sufficient for our system. The number of replications can be determined by comparing the output values of a test set of replications to the empirical mean for that variable. The number of replications can be determined by the following formula (see Soest, 1988):

\[ n \geq \frac{4t^2\sqrt{\alpha}(n_0 - 1)s^2}{\delta^2} \]

In which \( n \) is the number of replications, \( t_{wa}(n_0 - 1) \) equals the upper \( \frac{1}{2}\alpha \) value of the student-t distribution with \( n_0 - 1 \) degrees of freedom, \( \alpha \) is the preferred confidence level, \( \delta \) equals the width of the confidence interval and \( s^2 \) is the variance of the output values after \( n_0 \) replications.

---

3 According to Sol (1982), the run length can be roughly determined by taking three times the longest cycle time within the system.

4 This formula applies under the condition that the replication results are mutually independent and normally distributed.
Figure 6.4: Plots to determine the warm-up time

With a confidence level of 95 % (α = 0.05) and an initial number of replications of \( n_0 = 10 \), formula (1) results in \( n_0 - 1 = 9 \) and \( t_{0.05} (9) = 2.262 \), see Soest, 1988):

\[
n \geq \frac{20.47 s^2}{\delta^2}
\]  

(2)

To illustrate the procedure, we give the calculation for the output variable of the response time of assignments which demand a maximum response time of 8 hours. We chose a value of 10 minutes for the width of our confidence interval (\( \delta = 10 \)). The variance of the 10 means resulting from 10 replications (\( s^2 \)) was 38.04. This means that the number of replications should be greater than \( 20.47 \times 38.4 / 100 = 7.86 \), meaning 8 or more. After a similar investigation of the other output variables we conclude that the initial 10 replications is sufficient for statistical analyses at the chosen level of significance.
6.2.3 Verification and validation

Verification is carried out in order to prove that the model is sufficiently accurate with respect to its internal functioning. Validation analyzes the correspondence of the model with the real problem situation. Below we discuss verification and validation for the service engineers case study.

Verification

Verification was carried out by testing individual parts of the model for their behavior. For example the generation pattern of jobs and the work sequence of engineers were tested. The following methods were used for verification:

- Animation: the animation proved to be very helpful in finding pieces of code that were erroneous.
- Walk-throughs: simulation code was tested by walking through it, both manually and by generating one item that follows the code.
- Dynamic behavior of variables: this test concerned whether output variables could reach invalid values (e.g. the capacity utilization of active engineers should always be greater than zero).

Validation

The model was checked against reality by expert validation and by replicative validation. By showing the model and the model outcomes to two members of the project team that work with Dutch Telecom, the basic functioning of the model could be validated and a rough estimate of the numerical correspondence could be made.

A replicative validation was carried out by performing a t-test on the difference between the means of the output variables and the empirical means (see Soest, 1988). We illustrate how this was done by giving examples of two output variables that concern the mean response time for assignments demanding response within 8 hours (R8 assignments)
Designing Organizational Coordination

and those demanding response within 18 hours (R18 assignments). We have the following data to perform the t-test (the empirical mean and variance result from a sample of 11 mean response times, collected from 11 consecutive weeks):

<table>
<thead>
<tr>
<th>Variable</th>
<th>Experimental mean (10 reps.)</th>
<th>Experimental variance (10 reps.)</th>
<th>Empirical mean (11 obs.)</th>
<th>Empirical variance (11 obs.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R8 Response</td>
<td>285</td>
<td>38.04</td>
<td>285</td>
<td>39.80</td>
</tr>
<tr>
<td>R18 Response</td>
<td>524</td>
<td>18.00</td>
<td>529</td>
<td>44.60</td>
</tr>
</tbody>
</table>

*Table 6.4: t-test data*

We test the difference between the experimental means and the empirical means. Test variable is the difference d between the sample means and the means from the 10 replications of the simulation model \( d = \bar{x}_1 - \bar{x}_2 \). The model will pass the test if \( |d / s_d| \) lies between the following critical values:

\[
-t_{\alpha}(n_1 + n_2 - 2) \leq \frac{d}{s_d} \leq t_{\alpha}(n_1 + n_2 - 2) \quad \text{or} \quad \left| \frac{d}{s_d} \right| \leq t_{\alpha}(n_1 + n_2 - 2)
\]  

where

\[
s_d = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} \left( \frac{1}{n_1} + \frac{1}{n_2} \right)
\]  

We refer to \(|d / s_d|\) as \( |t|\). Applying (3) and (4) to the two variables leads to table 6.5, where we chose a confidence level (\( \alpha \)) of 0.05. The t value for this level of confidence and 19 degrees of freedom (\( n_1 + n_2 - 2 = 19 \)) is \( t_{0.025}(19) = 2.093 \).

From table 6.5, we conclude that there is no significant difference between empirical and experimental values. We consider the simulation model to be valid for the means of the R8 and R18 response times as in both occasions d lies between the critical values.

154
Applying the design system: two cases revisited

<table>
<thead>
<tr>
<th></th>
<th>R8 response time</th>
<th>R18 response time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_1 / n_2$</td>
<td>11 / 10</td>
<td>11 / 10</td>
</tr>
<tr>
<td>$x_1 / x_2$</td>
<td>285 / 285</td>
<td>525 / 529</td>
</tr>
<tr>
<td>$d$</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>$s_1^2 / s_2^2$</td>
<td>39.80 / 38.04</td>
<td>44.60 / 18.00</td>
</tr>
<tr>
<td>$s_n$</td>
<td>2.72</td>
<td>2.47</td>
</tr>
<tr>
<td>$t_1 / t_{2.093}$ (19)</td>
<td>0 / 2.093</td>
<td>1.62 / 2.093</td>
</tr>
</tbody>
</table>

Table 6.5: Results of the t-test

6.2.4 Solution finding and consistency check

An analysis of the simulation model taught us that the dispatchers as well as the engineers showed a very high capacity utilization (over 80 percent) because of the length of the phone calls of engineers. To improve this situation, the capacity of the dispatchers should be increased or the time for coordination between dispatcher and engineer should be decreased.

An analysis of possibilities for the latter alternative led to the idea of equipping the engineers with mobile phones, laptop computers and modems. At the home base, one could install an extra database in which assignments could be stored for electronic collection by the engineer. Communication time would be minimal and the only task of the dispatcher would be to assign an engineer to a job and put it in the local database.

The engineers, having a lap-top computer, could now fill out the billing notes by typing the information about used material and time spent into a program on their lap-top. Upon communication with the home base, this information could also be transferred, solving the problem of poor handwriting. The lap-top and modem alternative was implemented in the simulation model for evaluation. The results of this alternative and a comparison with the existing situation is given in the next section. For this alternative, the following changes were made in the simulation model’s input variables:
Designing Organizational Coordination

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to assign and store a job in the database</td>
<td>tria(0.5, 0.75, 1)</td>
</tr>
<tr>
<td>Time for dispatching an assignment</td>
<td>1 (minute)</td>
</tr>
<tr>
<td>Time for trying a call by engineer</td>
<td>tria(1, 1.5, 2)</td>
</tr>
<tr>
<td>Time before engineer brings in notes for billing</td>
<td>same day (electronically)</td>
</tr>
</tbody>
</table>

*Table 6.6: Changed input data for the simulation model*

According to the results of a test run, the start-up time did not have to be altered. The run-length was again determined at 10 days and, using formula (1), the number of replications was also kept at 10.

The adaption of the model in terms of lap-tops and communication devices will not render the simulation model inconsistent with the conceptual model. The adaptations of the simulation model can be made in terms of extra links (modems / mobile phones) and extra repositories (the lap-tops and a central computing unit).

### 6.2.5 Results

In this section, we compare the results of the simulation model of the existing situation to the results of the model of the lap-top alternative. First of all, an overview of the results is given in table 6.7. Each row of the table shows the experimental results for one output variable, relative to the number of engineers that are active. For all variables, the value for the current situation, the value of the lap-top alternative and the \(| t |\) value of a t-test on the difference between current situation and alternative are given (top row = current situation, middle row = lap-top alternative, bottom row = \(| t |\) value). The variation of each result is given by showing the bandwidth of the 95% confidence interval for that variable. The 95% confidence interval for a variable can be determined as follows:

\[
x - t_{\alpha/2}(n) \cdot \frac{s}{\sqrt{n}} \leq \text{variable} \leq x + t_{\alpha/2}(n) \cdot \frac{s}{\sqrt{n}} \quad (\text{where } \alpha = 0.05)
\]  

(5)
Applying the design system: two cases revisited

Improvements are significant (bold in table 6.7) when the $|t| \leq t_{\alpha, \nu_1 + \nu_2 - 2} = t_{0.05}(18) = 2.101$. A graphical representation of the results, is given in figures 6.5 to 6.10.

<table>
<thead>
<tr>
<th>Number of engineers active</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
</tr>
<tr>
<td>Mean Response time R8</td>
</tr>
<tr>
<td>Current</td>
</tr>
<tr>
<td>Laptop</td>
</tr>
<tr>
<td>ltl</td>
</tr>
<tr>
<td>Mean Response time R18</td>
</tr>
<tr>
<td>Current</td>
</tr>
<tr>
<td>Laptop</td>
</tr>
<tr>
<td>ltl</td>
</tr>
<tr>
<td>Capacity utilization</td>
</tr>
<tr>
<td>engineers (%)</td>
</tr>
<tr>
<td>Current</td>
</tr>
<tr>
<td>Laptop</td>
</tr>
<tr>
<td>ltl</td>
</tr>
<tr>
<td>Capacity utilization</td>
</tr>
<tr>
<td>dispatchers (%)</td>
</tr>
<tr>
<td>Current</td>
</tr>
<tr>
<td>Laptop</td>
</tr>
<tr>
<td>ltl</td>
</tr>
<tr>
<td>Occasions of waiting jobs</td>
</tr>
<tr>
<td>Current</td>
</tr>
<tr>
<td>Laptop</td>
</tr>
<tr>
<td>ltl</td>
</tr>
<tr>
<td>Total throughput time</td>
</tr>
<tr>
<td>Current</td>
</tr>
<tr>
<td>Laptop</td>
</tr>
<tr>
<td>ltl</td>
</tr>
</tbody>
</table>

$\alpha = 0.05, n_1 = 10, n_2 = 10, t_{\alpha, \nu_1 + \nu_2 - 2} = t_{0.05}(18) = 2.101$

**Table 6.7:** Experimental results of the simulation model

The mean response times for R8 and R18 assignments are given in figures 6.5 and 6.6. The average relative improvement is 8.8 percent and 5.1 percent respectively. The relative improvement drops when more engineers are active.

The response time that can be achieved with 14 engineers in a normal situation can already be achieved by 10 engineers when they use the communication equipment.
Figure 6.5: Mean response time for R8 assignments

Figure 6.6: Response time of R 18 assignments
Figures 6.7 and 6.8 display the capacity utilization of the engineers and the dispatchers. For the engineers we see a relative improvement in capacity utilization by 20.7 percent on average for the engineers and by 86.5 percent for the dispatchers.

**Figure 6.7: Capacity utilization of engineers**

**Figure 6.8: Capacity utilization of dispatchers**
Designing Organizational Coordination

This means that with the lap-top alternative, the normal work for each group of five engineers can now be done by four. The number of dispatchers can be adjusted to one instead of two. Figure 6.9 shows the average number of occasions that a job cannot be dispatched because all active engineers are busy. This value drops when more engineers are active. However, when the lap-tops are installed, this number immediately drops to a minimum value at 10 or 11 active engineers.

![Graph showing the number of waiting jobs](image)

**Figure 6.7: Number of waiting jobs**

Finally, figure 6.10 displays the mean total throughput time of an assignment from entering the system until sending the bill. Because of billing time and other handling times, this value is not influenced by the number of engineers. However, with the lap-top alternative, the engineers deliver their billing notes electronically and on the same day instead of manually and after two days. This explains the drop in total throughput time between existing and new situation.

As a result of electronic communication instead of handwritten billing notes, billing can also be done more accurately.
Applying the design system: two cases revisited

**Figure 6.10: Total throughput time**

6.2.6 Conclusions

From this case study we can draw several conclusions. First of all, for the service engineers and the dispatchers to coordinate successfully and fast, the implementation of information and communication technology is feasible and promising. From table 6.7 we conclude that the lap-top alternative results in significant improvements of all variables except for one in the situation of 13 active service engineers.

The results of the case study clearly indicate that a relatively small investment in terms of coordination between work processes, results in substantial gains in terms of efficiency and effectiveness. The number of engineers needed to accurately finish the workload can be reduced from 12 to 10. This figure can be extrapolated to each team of engineers operated by Dutch Telecom. For further information on this case study, we refer to the case report (van Eijck et al., 1992).

Furthermore, the method as applied in this case study proved successful for the development of a thorough understanding of the existing
Designing Organizational Coordination

situation and the assessment of an alternative. With the model, the results of other alternatives like increasing the number of dispatchers, can also be calculated.

This shows that the methodology we developed fits one of the problem situations from which we departed. In the following section we do the same in another environment.

However, this is a relatively simple case. Neither the network nor the process model show much complexity and the simulation model is relatively straightforward too. The methodology should be tested further in more complicated environments. Cases two to five are about such environments.
Applying the design system: two cases revisited

6.3 Case 2: Insuraco (revisited)

The second case study, too, is one we have encountered before in chapter 2. It concerns the business unit of individual insurances of a large general insurance company. We focus on the investigation of the primary processes. For a detailed description of the case environment, we refer to chapter 2. Below, we give a brief sketch of the problem situation.

Due to the increasing number of loss claims and due to strong competition from other companies, especially the direct writers, the organization is forced to revise its primary processes, in order to make the internal cost rate more acceptable. Internal cost rates are determined mainly by four types of primary processes: accepting new insurance applications, handling mutations on registered insurance policies, handling loss claims and supplying the customer (usually by telephone) with information. These processes are not always implemented as efficiently and effectively as possible, resulting in an unacceptable performance level in terms of backlog and throughput time. Not only is there the problem of the costs related to high backlog and throughput time, but there is also a commercial risk: insurance intermediaries demand speedy processing for their clients. To design processes that are more efficient and effective, the processes and their coordination have been analyzed using the design system.

6.3.1 Conceptualization

In this section we present the conceptualization, according to the steps as outlined in chapter 4. We first identify the object classes, then we construct the conceptual model and formulate a problem statement.

Identify object classes

The identification of object classes can be done by describing the four processes from beginning to end. For this, we summarize the process description as given in section 2.6.2.
Designing Organizational Coordination

Early in the morning the mail bags arrive at the Insuraco mail room. The mail is sorted according to department with the assistance of some employees of Dutch Mail. After sorting, the mail can be sent to the departments or to individual employees. Mail is sent to the acceptance and administration department and the loss claims department.

At the acceptance department, each document is sorted by one of the employees according to whether it concerns a fire insurance or a motor insurance. The sorter also differentiates between mutations and new insurance applications. After this, the mail is sent to the acceptors at the respective sub-departments. The batches are processed here by the acceptors. If an acceptor is unable to accept or reject an application or mutation, then he sends it to one of the liaison managers who contacts the agent. If the liaison manager reaches agreement with the agent, the order is sent back to the acceptor for system input. However, this does not mean to say that an acceptor never communicates with the agents. He may write or phone an agent to correct errors or omissions on the application form. Incoming phone calls for the acceptance department are received by a special phone department called the teledesk. Many applications and questions can be handled by phone. The phone employees have the same expertise and authorization as the acceptors.

At the loss claims department, the claims are sorted into complex claims and non-complex claims. The claims are sent to the respective employees (employees that can handle complex claims may also handle non-complex claims). Complex claims require more frequent contact with the insurance agent. This is usually in writing but there is also frequent phone contact.

From this description, we can identify the object classes for our conceptual model. Following a loss claim or an insurance application from the intermediary to the department where it is processed, we encounter actors like mail sorters, loss claims employees and acceptance employees and we encounter repositories like mail bags and personal workloads. A discussion of this approach with Insuraco management led to the object classes of tables 6.8 to 6.11 (for reasons of space usage, we present the object classes in a tabular format).

164
Applying the design system: two cases revisited

Table 6.8 represents the objects of the actor class, including the employees of Insuraco, the Insuraco mailroom and the insurance intermediary.

<table>
<thead>
<tr>
<th>Object</th>
<th>Attributes</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insurance intermediary</td>
<td>Identifier</td>
<td>Send applications / mutations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Send loss claim</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Make phone call</td>
</tr>
<tr>
<td>Insuraco mailroom</td>
<td></td>
<td>Sort mail</td>
</tr>
<tr>
<td>Acceptance sorter</td>
<td>Identifier</td>
<td>Sort mail into fire / motor</td>
</tr>
<tr>
<td>Motor acceptor</td>
<td>Identifier</td>
<td>Handle motor application / mutation</td>
</tr>
<tr>
<td></td>
<td>Expertise</td>
<td>Send document to liaison manager</td>
</tr>
<tr>
<td>Fire acceptor</td>
<td>Identifier</td>
<td>Handle fire application / mutation</td>
</tr>
<tr>
<td></td>
<td>Expertise</td>
<td>Send document to liaison manager</td>
</tr>
<tr>
<td>Liaison manager</td>
<td>Identifier</td>
<td>Accept application / mutation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reject application / mutation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contact intermediary</td>
</tr>
<tr>
<td>Loss claims sorter</td>
<td>Identifier</td>
<td>Sort mail into complex / non complex</td>
</tr>
<tr>
<td>Complex claims handler</td>
<td>Identifier</td>
<td>Handle complex loss claim</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Handle non-complex loss claim</td>
</tr>
<tr>
<td>Non complex claims handler</td>
<td>Identifier</td>
<td>Handle non-complex loss claim</td>
</tr>
<tr>
<td>Telephone desk employee</td>
<td>Identifier</td>
<td>Take phone call</td>
</tr>
</tbody>
</table>

Table 6.8: Actor classes

In table 6.9 we identify the repository classes such as the mail bags at the Insuraco mailroom and the workloads of documents to be sorted. Next to the sorting repositories, each employee has his own workload in a personal repository.

<table>
<thead>
<tr>
<th>Object</th>
<th>Attributes</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mail room documents</td>
<td>Identifier</td>
<td>Store documents</td>
</tr>
<tr>
<td>Acceptance documents to be sorted</td>
<td>Identifier</td>
<td>Store acceptance documents</td>
</tr>
<tr>
<td>Loss claims documents to be sorted</td>
<td>Identifier</td>
<td>Store loss claims documents</td>
</tr>
<tr>
<td>Telephone queue</td>
<td>Identifier</td>
<td>Hold calls</td>
</tr>
<tr>
<td>Personal workload (for each actor)</td>
<td>Actor ident.</td>
<td>Store documents</td>
</tr>
</tbody>
</table>

Table 6.9: Repository classes

In table 6.10, we identify the link classes. Three links are of importance here; the Dutch mail, Insuraco’s internal mail service and the phone system.
Table 6.10: Link classes

Finally, the item classes are displayed in table 6.11. Insuraco differentiates between acceptance documents, loss claims and phone calls. Attributes determine the complexity of a claim, the type of insurance and the type of document.

Table 6.11: Item classes

Construct conceptual model

From the case situation and the object classes as identified, we can construct conceptual models according to the steps of chapter 4. The network model of the problem situation is depicted in figure 6.11. Items flow through the system from right to left. The Insuraco part of the system is divided into a loss claims section and an acceptance section. The former is divided into complex and non-complex claims and the latter into a fire department and a motor department.

The liaison manager is essential in the coordination between the acceptance department and the insurance intermediary. For the loss claims departments, there is no such person, resulting in direct (phone or mail) communication between loss claims employees and the intermediary.

166
In figure 6.12, we display part of the process model. The process model can be divided into a part for mail handling and a part for the phone desk. Figure 6.12 only displays the mail handling part. In the process model we recognize the tree-like structure of the network model, in terms of the different sorters and the acceptance and loss claims employees. Mail is sorted twice before it can be sent to the right specialist. For each of the employees an actor model can be made. One of these actor models, the one for the motor acceptors, is displayed in figure 6.13.

In figure 6.13 we see that the task of the motor acceptor can be divided into two parts: the actual acceptance work and the investigation of the documents for irregularities caused by the intermediary. Although the acceptance work is the most important part, the correction of errors
Designing Organizational Coordination

made by the intermediary also takes much time, especially when intermediaries use their own application forms. These forms are not adjusted to insuraco files and databases so they regularly do not contain the required information.

\[\text{Figure 6.12: Process model of mail handling}\]
Figure 6.13: Actor model of the motor acceptor
Designing Organizational Coordination

Other process and actor models were also constructed and discussed with the acceptors, liaison managers, phone desk and mail room employees at Insuraco. We do not incorporate these in this thesis, but refer to the case report for a full explanation (Eijck and Attema, 1993).

Formulate problem statement

The problem as formulated by Insuraco management is a logistic one. Average throughput time for loss claims, mutations and new applications was too high, resulting in unacceptable backlogs. The problem statement, in terms of our conceptual model, can therefore be formulated as follows:

*How can we reduce the backlog of loss and acceptance documents accumulating at the employees of the loss and acceptance departments. What role can electronic data interchange play to this effect?*

6.3.2 Specification

In this section we present the simulation model of the present situation, based on the conceptual model of the previous section. We do this by describing each of the steps as discussed in section 4.3.2.

Output variables

Based on the conceptual models and the introduction, we identify a logistic problem: backlog and throughput time are too high. With Insuraco management we established the following performance indicators for an analysis of the system:

- Backlog level for motor documents.
- Backlog level for fire documents.
- Backlog level for complex loss claims documents.
- Backlog level for non-complex loss claims documents.
- Throughput time for acceptance documents.
- Throughput time for loss claims documents.
Applying the design system: two cases revisited

Backlog levels are indicators which Insuraco keeps records of, so these can be used for replicative validation. Furthermore, as they use these indicators for adjusting the size of their workforce, backlog levels are very recognizable for Insuraco management. Insuraco was less used to throughput times as performance indicators. However since Insuraco wanted insight into their performance as perceived by intermediaries, these variables were incorporated in the model.

Model reductions

The conceptual models show that the empirical situation consists of the following main elements:

- Sub-systems that generate documents and phone calls.
- A mail sorting and routing subsystem.
- Sub-systems for the processing of documents and phone calls.
- Sub-systems that deal with exceptions such as the liaison managers.

These sub-systems are all specified in the simulation model, but in agreement with Insuraco management, and based on an investigation of the available empirical data, the following model reductions were made in order to reduce the simulation model’s complexity.

- Work is generated by three exponential distributions: one for new applications, one for mutations, and one for loss claims.
- Handling times are reduced to triangular distributions, estimated by Insuraco personnel. Apart from the difference between fire and motor, no distinction is made between specific types of insurance.
- The loss claims documents are split into complex and non-complex by chance: 81% of the documents is non-complex, 19% is complex. These figures are derived from Insuraco files.
- In the model, there is no interaction between the teledesk and the acceptance departments. In practice this happens only sporadically.
- The employees of the acceptance department do not receive phone calls. The time for calling is incorporated in handling time.
Designing Organizational Coordination

Input data

Input data was extracted from Insuraco files and from estimates of experts (acceptors, liaison managers, loss claims employees and mail room employees) working at Insuraco. The input values concerning handling times and item generation are displayed in table 6.12. The number of generated items was derived from Insuraco files. The handling times were estimated by several employees who carry out the related tasks. We did not take samples of the handling times as this is a very time-consuming activity, and because the act of measuring handling times influences the result. The balance between complex and non-complex loss claims and between fire and motor insurances was also found in Insuraco records.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applications generated per day</td>
<td>tria(175,210,284)</td>
</tr>
<tr>
<td>Mutations generated per day</td>
<td>tria(888,1047,1443)</td>
</tr>
<tr>
<td>Loss claims generated per day</td>
<td>tria(507,625,722)</td>
</tr>
<tr>
<td>Loss claims phone calls generated per hour</td>
<td>expo(60)</td>
</tr>
<tr>
<td>Balance complex / non-complex</td>
<td>1:4:26</td>
</tr>
<tr>
<td>Balance fire / motor</td>
<td>1:0:6</td>
</tr>
<tr>
<td>Handling time fire application</td>
<td>tria(9,5,12,14,5)</td>
</tr>
<tr>
<td>Handling time fire mutation</td>
<td>tria(4,2,5,2,6,2)</td>
</tr>
<tr>
<td>Handling time motor application</td>
<td>tria(10,12,14)</td>
</tr>
<tr>
<td>Handling time motor mutation</td>
<td>tria(4,5,6)</td>
</tr>
<tr>
<td>Handling time complex loss claims phone call</td>
<td>tria(3,6,15)</td>
</tr>
<tr>
<td>Handling time non-complex loss claims phone call</td>
<td>tria(1,5,10)</td>
</tr>
<tr>
<td>Handling time phone call (phone desk)</td>
<td>tria(0,5,2,10)</td>
</tr>
</tbody>
</table>

Table 6.12: Input data for the Insuraco model

Simulation model and animation

From the conceptual models, the input data and the output variables, a simulation model is constructed for the analysis of the current situation, together with several alternatives. The network model served as a base model for the simulation model, which was constructed along the lines of the generic translation into Siman IV code of chapter 4.
Applying the design system: two cases revisited

The simulation model consists of code blocks for the following parts of the Insuraco system (figure 6.14 shows a screen dump of the animation):

- Work generators: each type of work has its own work generator. The paperwork is sent to the Dutch Mail, who delivers it to Insuraco. The delivery is programmed using a wait/signal construction. Phone calls are connected to the teledesk if they concern acceptance and to the loss claims department if they concern loss claims.
- The mail room: the mail room sorts the documents into loss claims documents and acceptance documents and sends it to the respective departments via the internal mail routing service.
- The loss claims sorter: sorts out the complex and non-complex documents and delivers these to the right department.
- The phone desk for acceptance phone calls.
- The acceptance sorter: sorts out the fire and motor documents and delivers these to the right department.
- The loss claims employees: they handle loss claims. Both complex and non-complex loss claims employees are implemented. The loss claims employees also receive phone calls.
- The acceptors and the liaison managers.

Figure 6.14: Screenshot of the Insuraco model
Designing Organizational Coordination

Initial treatment

For the initial treatment, we have to establish the start-up time, the run-length and the number of replications needed to achieve results that reach a certain level of confidence. Like in the previous case, the system is non-terminating. We estimate the start-up time from a plot of the different stock levels of entities in the system. Measured levels are: number of fire documents, number of motor documents, number of complex loss documents, number of non-complex documents and the total number of documents in the system. To reduce the start-up time, we filled the related queues in the model with the average number of documents found in Insuraco records. Figure 6.15 shows the plots and we conclude that a start-up time of 5 days is sufficient. As cycle times are short (less than one day), we determined the run-length at 5 days.

![Graph showing plots to determine the start-up time](image)

**Figure 6.15: Plots to determine the start-up time**

After an initial number of 5 replications, we determined the number of replications needed for a treatment by means of formula (1) and the
Applying the design system: two cases revisited

procedure as sketched in section 6.2.2. Table 6.13 shows the results of this procedure for three output variables and a confidence level of 95%. The delta values are based on the variation in values from Insuraco files. We conclude that a total number of 10 replications is sufficient.

<table>
<thead>
<tr>
<th>Variable (results after 5 replications)</th>
<th>Mean</th>
<th>Standard deviation</th>
<th>Delta</th>
<th>Minimum # replications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backlog Fire documents</td>
<td>887</td>
<td>52.1</td>
<td>100</td>
<td>9</td>
</tr>
<tr>
<td>Backlog complex documents</td>
<td>235</td>
<td>25.1</td>
<td>50</td>
<td>8</td>
</tr>
<tr>
<td>Throughput time acceptance</td>
<td>53.4 (hours)</td>
<td>0.51</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6.13: Determining the number of replications

6.3.3 Verification and validation

Verification and validation was carried out in the same way as in the previous case study. Below we give a brief outline of both procedures.

Verification

The individual parts of the model as given in the model description above were tested for their behavior by means of an numerical analysis of their functioning and a check of the program code. The animation was helpful in finding errors in the simulation code.

The code was tested by manual walk-throughs and by letting individual items follow parts of the code and testing variable values afterwards. Since the complexity of the model is limited, this could be done rather quickly. For further information we refer to Eijck and Attema (1993).

Validation

The model was validated by means of replicative validation and by expert validation. Simulation results and the animation were shown to acceptors of the fire and motor departments, complex and non-complex loss claims employees, liaison managers and phone desk employees. The discussions led to the fine tuning of several input values.
Designing Organizational Coordination

A replicative validation was carried out by performing a t-test on the difference between the means of the output variables for 10 replications and the means of empirical values found in a sample of 5 observations from Insuraco records. Unfortunately, it was not possible to access the records for throughput times, so samples could only be generated from backlog files. Table 6.14 shows the results of the replicative validation where index 1 refers to the sample from Insuraco records and index 2 refers to the 10 replications of the model. The level of confidence is 95 %, resulting in a t-value of \( t_{\alpha}(n_1 + n_2 - 2) = t_{0.025}(13) = 2.160 \). The \( |t| \) values give the half-width of the 95 % confidence intervals of \( d / s_d \). Based on these values we conclude that there is no significant difference between model and reality for these variables.

<table>
<thead>
<tr>
<th>Document type</th>
<th>( x_1 )</th>
<th>( s^2_1 )</th>
<th>( x_2 )</th>
<th>( s^2_2 )</th>
<th>( d )</th>
<th>( s_d )</th>
<th>( t )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceptance</td>
<td>1933.4</td>
<td>450826</td>
<td>1886.0</td>
<td>8990</td>
<td>47.4</td>
<td>208.5</td>
<td>0.22</td>
</tr>
<tr>
<td>Non-complex claims</td>
<td>482.0</td>
<td>3046</td>
<td>441.8</td>
<td>6188</td>
<td>40.2</td>
<td>39.6</td>
<td>1.01</td>
</tr>
<tr>
<td>Complex claims</td>
<td>257.0</td>
<td>1089.5</td>
<td>271.7</td>
<td>1665</td>
<td>14.7</td>
<td>21.1</td>
<td>0.70</td>
</tr>
</tbody>
</table>

\( \alpha = 0.05, n_1 = 5, n_2 = 10, t_{\alpha}(n_1 + n_2 - 2) = 2.160 \)

**Table 6.14: Results of the replicative validation**

6.3.4 Solution finding and consistency check

In this section we show that the system is very sensitive to a reduction of the number of phone calls for the loss claims department and to changes in handling time for the loss claims and acceptance departments. Furthermore, we assess the influence of electronic data interchange on the internal processes.

With respect to the phone calls, the teledesk of the acceptance department was a logical example of an alternative for the loss claims department. An assessment of this alternative in terms of throughput times and backlog levels could be acquired by reducing the number of phone calls for the loss claims department in the simulation model. To get a clear picture of what the actual influence of phone calls is, we reduced the number of phone calls to the loss claims department to zero in one of the alternatives.

176
Applying the design system: two cases revisited

The EDI alternatives can be divided into two types: alternatives that incorporate electronic delivery, but still use manual data entry in the Insuraco systems and alternatives that influence handling times of acceptors and loss claims employees. The first type of alternative of merely delivering documents electronically, printing them and then entering them into the systems manually sounds a little strange, but it is the current practice of EDI in the insurance business in the Netherlands.

The above discussion of possible alternatives leads to the following treatment plan:

<table>
<thead>
<tr>
<th>Treatment</th>
<th>Alternative</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Current situation</td>
</tr>
<tr>
<td>1</td>
<td>No more phone calls for the loss claims department</td>
</tr>
<tr>
<td>2</td>
<td>50% EDI delivery, no influence on handling times</td>
</tr>
<tr>
<td>3</td>
<td>100% EDI delivery, no influence on handling times</td>
</tr>
<tr>
<td>4</td>
<td>100% EDI + 10% reduction of handling times due to EDI</td>
</tr>
<tr>
<td>5</td>
<td>As treatment 4 but no more phone calls for the loss claims department</td>
</tr>
</tbody>
</table>

Table 6.15: Treatment plan for Insuraco

In an EDI alternative, the documents are sent directly from the insurance intermediary to the employee that is going to handle the document. This means that mail routing and sorting activities are bypassed. In treatment 4 and 5, we multiplied all handling times by 0.90.

For treatments 1 to 3, the run-length was again set at 5 days and, using formula (1) the number of replications was also kept at 10. For treatments 4 and 5, we had to increase the start-up time to 20 days because the model does not become stable before day 15. The results of the treatments are discussed in the following section.

Consistency

The alternatives will not render the respective simulation models inconsistent with the conceptual models as shown in section 6.3.1. As in the previous case study, the changes can be made by means of adding link objects and making changes in input values.
Designing Organizational Coordination

6.3.5 Results

In this section we give the results of the treatments as described in the previous section. The intervals as mentioned in tables 6.16 to 6.21 are 95\% confidence intervals of the related variable.

Significance of output values of the alternatives was tested by a t-test on the difference between the means of the alternative and the means of treatment 0. In all cases, the level of confidence is 95\%, resulting in the following parameters for the t-test on the difference of the means:
\[ \alpha = 0.05, n_1 = 10, n_2 = 10, \ t_{\text{wa}}(n_1 + n_2 - 2) = t_{\text{wa}}(18) = 2.101. \]

Treatment 0

The first treatment concerns the current situation and serves as a yardstick for the alternatives. The values of the output variables are displayed in table 6.16.

<table>
<thead>
<tr>
<th>Output variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backlog level for motor documents</td>
<td>1090 ± 75</td>
</tr>
<tr>
<td>Backlog level for fire documents</td>
<td>796 ± 45</td>
</tr>
<tr>
<td>Backlog level for non-complex loss claims</td>
<td>442 ± 55</td>
</tr>
<tr>
<td>Backlog level for complex loss claims</td>
<td>272 ± 29</td>
</tr>
<tr>
<td>Average throughput time for acceptance documents</td>
<td>54.9 ± 1.2 hours</td>
</tr>
<tr>
<td>Average throughput time for loss claims</td>
<td>49.9 ± 1.6 hours</td>
</tr>
</tbody>
</table>

\[ \alpha = 0.05, n = 10, \text{values determine 95\% confidence intervals} \]

Table 6.16: Results of treatment 0, current situation

Treatment 1

Treatment 1 concerns the reduction of telephone calls for the employees of the loss claims department. The effect on the output variables that concern loss claims of a reduction of phone calls to zero is displayed in table 6.17 below. In this alternative, we assume that phone calls for the loss claims department are handled by a special phone desk, comparable to the one operating for the acceptance departments.
Applying the design system: two cases revisited

<table>
<thead>
<tr>
<th>Output variable</th>
<th>Value</th>
<th>t</th>
<th>Significant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backlog level for non-complex claims</td>
<td>24 ± 15 (-94%)</td>
<td>16.2</td>
<td>Yes</td>
</tr>
<tr>
<td>Backlog level for complex claims</td>
<td>23 ± 10 (-91.5%)</td>
<td>18.3</td>
<td>Yes</td>
</tr>
<tr>
<td>Average throughput time for claims</td>
<td>25.2 ± 0.2 hours (-51%)</td>
<td>33.4</td>
<td>Yes</td>
</tr>
</tbody>
</table>

\[ \alpha = 0.05, n_1 = 10, n_2 = 10, t_{calc}(n_1 + n_2 - 2) = t_{crit}(18) = 2.101 \]

**Table 6.17:** Results of treatment 1, reduction of claims phone calls

Between brackets we display the improvements of the different output variables relative to treatment 0 where they are significant according to a t-test on the difference of the means. | t | is defined as \(d/s_a\) where \(d\) is the difference between the means and \(s_a\) results from formula (4).

### Treatment 2

Treatment 2 concerns an EDI alternative. The chance of electronic transmission between intermediary and loss claims or acceptance departments is set to 50 percent. The influence of EDI transmission on handling times is assumed zero, so this alternative only bypasses mail room and sorting procedures. The results are displayed in table 6.18.

<table>
<thead>
<tr>
<th>Output variable</th>
<th>Value</th>
<th>t</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backlog level for motor documents</td>
<td>1454 ± 28 (+33%)</td>
<td>10.1</td>
<td>Yes</td>
</tr>
<tr>
<td>Backlog level for fire documents</td>
<td>1107 ± 82 (+39%)</td>
<td>7.4</td>
<td>Yes</td>
</tr>
<tr>
<td>Backlog level for non-complex loss claims</td>
<td>483 ± 40</td>
<td>0.1</td>
<td>No</td>
</tr>
<tr>
<td>Backlog level for complex loss claims</td>
<td>295 ± 25</td>
<td>1.3</td>
<td>No</td>
</tr>
<tr>
<td>Avg. throughput time for acceptance</td>
<td>56.8 ± 2.0 hrs.</td>
<td>1.7</td>
<td>No</td>
</tr>
<tr>
<td>Average throughput time for loss claims</td>
<td>42.1 ± 1.5 hrs. (-15%)</td>
<td>7.8</td>
<td>Yes</td>
</tr>
</tbody>
</table>

\[ \alpha = 0.05, n_1 = 10, n_2 = 10, t_{calc}(n_1 + n_2 - 2) = t_{crit}(18) = 2.101 \]

**Table 6.18:** Results of treatment 2, 50 % EDI

In this alternative we see that 50 % EDI without influencing the handling times has the opposite effect or no effect at all. Only the throughput time for loss claims shows a small but significant improvement. The increase in backlog levels is due to the fact that two internal buffers (the mail room and the sorters) are now bypassed. The documents that are normally in these buffers are now in the buffers of acceptors and loss claims employees. It is remarkable that the decrease in handling times (no more sorting) only has effect on loss claims documents.
Designing Organizational Coordination

Treatment 3

Treatment 3 concerns the same alternative as treatment 2, but now we assume that all documents are delivered electronically. The results of this alternative are given in table 6.19 below.

With this alternative, we see an amplification of the effect of treatment 2. For the loss claims department, it seems that bypassing the mail room and sorting procedures improves the average throughput time, meaning that sorting and routing takes longer than processing (if you include the time a document waits in the different queues).

<table>
<thead>
<tr>
<th>Output variable</th>
<th>Value</th>
<th>t</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backlog level for motor documents</td>
<td>$2021 \pm 76 (\pm 85%)$</td>
<td>19.3</td>
<td>Yes</td>
</tr>
<tr>
<td>Backlog level for fire documents</td>
<td>$1206 \pm 58 (\pm 52%)$</td>
<td>12.5</td>
<td>Yes</td>
</tr>
<tr>
<td>Backlog level for non-complex loss claims</td>
<td>$706 \pm 53 (\pm 60%)$</td>
<td>7.6</td>
<td>Yes</td>
</tr>
<tr>
<td>Backlog level for complex loss claims</td>
<td>$311 \pm 17 (\pm 14%)$</td>
<td>2.6</td>
<td>Yes</td>
</tr>
<tr>
<td>Avg. throughput time for acceptance</td>
<td>$58.2 \pm 1.0$ hours (+ 6%)</td>
<td>4.6</td>
<td>Yes</td>
</tr>
<tr>
<td>Average throughput time for loss claims</td>
<td>$40.3 \pm 2.0$ hours (- 20%)</td>
<td>8.2</td>
<td>Yes</td>
</tr>
</tbody>
</table>

$\alpha = 0.05, n_1 = 10, n_2 = 10, t_{18}(n_1 + n_2 - 2) = t_{\alpha}(18) = 2.101$

Table 6.19: Results of treatment 3, 100 % EDI

Treatment 4

In treatment 4, EDI is implemented in such a way that it influences handling time. We multiplied all handling times of the loss claims and acceptance departments by 0.9. The effect of this alternative was such that the simulation model could not become stable with a start-up time of 5 days.

The backlog levels dropped to a very low level, but they did so only after about 13 days. To arrive at a treatment that gives reliable and stable output, we increased the start-up time to 20 days. The results of this treatment are given in table 6.20 below.
Applying the design system: two cases revisited

<table>
<thead>
<tr>
<th>Output variable</th>
<th>Value</th>
<th>t</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backlog level for motor documents</td>
<td>231 ± 13 (79 %)</td>
<td>24.9</td>
<td>Yes</td>
</tr>
<tr>
<td>Backlog level for fire documents</td>
<td>132 ± 9 (83 %)</td>
<td>32.5</td>
<td>Yes</td>
</tr>
<tr>
<td>Backlog level for non-complex loss claims</td>
<td>127 ± 8 (71 %)</td>
<td>12.6</td>
<td>Yes</td>
</tr>
<tr>
<td>Backlog level for complex loss claims</td>
<td>32 ± 7 (88 %)</td>
<td>18.1</td>
<td>Yes</td>
</tr>
<tr>
<td>Avg. throughput time for acceptance</td>
<td>7.7 ± 0.6 hours (86 %)</td>
<td>76.6</td>
<td>Yes</td>
</tr>
<tr>
<td>Average throughput time for loss claims</td>
<td>7.7 ± 0.1 hours (85 %)</td>
<td>57.5</td>
<td>Yes</td>
</tr>
<tr>
<td>α = 0.05, n₁ = 10, n₂ = 10, tₚ₃(ₚ₁ + n₂ - 2) = tₚ₃₉₈(18) = 2.101</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 6.20: Results of treatment 4, EDI and 90 % handling time**

From these results we see that an alternative where EDI has a small influence on handling times may still have a big influence on throughput times and backlog levels.

**Treatment 5**

Treatment 5 concerns a combination of treatment 1 and treatment 4. The results of this treatment are given in table 6.21 below.

<table>
<thead>
<tr>
<th>Output variable</th>
<th>Value</th>
<th>t</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backlog level for motor documents</td>
<td>229 ± 10 (79 %)</td>
<td>25.2</td>
<td>Yes</td>
</tr>
<tr>
<td>Backlog level for fire documents</td>
<td>138 ± 11 (83 %)</td>
<td>31.9</td>
<td>Yes</td>
</tr>
<tr>
<td>Backlog level for non-complex loss claims</td>
<td>136 ± 6 (69 %)</td>
<td>12.2</td>
<td>Yes</td>
</tr>
<tr>
<td>Backlog level for complex loss claims</td>
<td>30 ± 4 (87 %)</td>
<td>18.6</td>
<td>Yes</td>
</tr>
<tr>
<td>Avg. throughput time for acceptance</td>
<td>7.7 ± 0.6 hours (86 %)</td>
<td>77.1</td>
<td>Yes</td>
</tr>
<tr>
<td>Average throughput time for loss claims</td>
<td>6.5 ± 0.1 hours (87 %)</td>
<td>59.2</td>
<td>Yes</td>
</tr>
<tr>
<td>α = 0.05, n₁ = 10, n₂ = 10, tₚ₃(ₚ₁ + n₂ - 2) = tₚ₃₉₈(18) = 2.101</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 6.21: Results of treatment 5, combining treatments 1 and 4**

All documents are delivered electronically, having an influence of 10 percent on handling times. Furthermore, the phone calls for the loss claims department are switched off, assuming that these are handled by a special phone department.

Logically, the results for acceptance documents do not significantly differ from the results of treatment 4. However, the results for the loss claims department improve significantly compared to treatment 4.
6.3.6 Conclusions

From this case study, we can draw several conclusions. First of all, the case study is more complex than the Telecom case of section 6.2. The empirical situation in itself was more complex, but we also investigated more alternatives. Therefore, the conceptual models are more complex. Although the models show more complexity, they are still easy to read and understand. The models were used in discussions with Insuraco management and personnel to get a clear and correct image of the business processes. The structure of the simulation models was based on the network model: before we implemented process logic, we first programmed the elements of the network model (actors, repositories and links) according to the generic translation as described in section 4.4.3.

The results of the experiments were clearly surprising for Insuraco management. The plans for EDI were not much further than the electronic transmission of data, printing the transmitted results and manually processing the data in the in-house systems. Insuraco expected to speed up the process by just bypassing the mail handling and sorting routines. The results of our analysis clearly show that this is not the case and that EDI will only work if it reduces the handling time of jobs done by acceptors and loss claims employees. In the case of Insuraco, this influence does not have to be very great to have a substantial effect on throughput times and backlog levels. In other words: the current system of Insuraco acceptance and claims handling is balanced, but unstable. The alternative of reducing the phone calls is another example.

Instability can be seen as a weakness of the current system, but it is also an opportunity. The fact that the system is sensitive to small changes in handling time stimulates a more detailed investigation of the tasks of acceptors and loss claims employees.

Another advice that can be given to Insuraco is to revise their strategy concerning electronic data interchange and the related role of the intermediaries.
Applying the design system: two cases revisited

If it is possible to stimulate the intermediaries to coordinate their systems to the Insuraco back-office, then tasks that are carried out both by the intermediaries and the back-office, such as data entry, can be eliminated. This could just be the small influence on handling times that would decrease average throughput time from more than two days to less than one.

This concludes the discussion of case studies that were already presented in chapter 2. Chapter 7 reveals new material, presenting three case studies on which the design system is tested.
Designing Organizational Coordination
7. Applying the design system: three new cases

'It's a whole complicated insurance thing. They just bury the whole thing. Pretend it never happened. The insurance business is completely screwy now. You know they've reintroduced the death penalty for insurance company directors?'
- Douglas Adams

7.1 Introduction

This chapter provides three case studies that we have not encountered before and to which the design system is being applied. The case studies took place in several different insurance companies, all operating under the same holding organization. This holding organization, called Dutch International, is one of the largest Dutch organizations operating on the international financial market. For reasons of confidentiality, we anonymized the names, so Dutch International and the names of the insurance companies are in reality different.

Case studies 3 and 4 concern the acceptance and surrender processes of life insurance policies at Dutch Life, the life insurance company of Dutch International. We investigated the logistics of policy acceptance and the coordination between several departments of Dutch Life and between Dutch Life and the intermediaries. Several change alternatives were analyzed from which recommendations were made for Dutch Life.

Case 5 was carried out at Dutch General, the general insurance company of Dutch International. We investigated the logistic processes of internal mail sorting and mail delivery. After a thorough investigation of the current situation, an alternative integrated system was designed to achieve better coordination between the respective departments of Dutch General.
Designing Organizational Coordination

7.2 Case 3: Dutch Life insurance acceptance

In this section we present a case study, carried out in another large Dutch insurance company, called Dutch Life. Dutch Life is the life insurance company of a large Dutch holding called Dutch International that operates on the international financial market.

Dutch Life sells Life Insurances, investment plans, pension plans, mortgages and several combinations of financial products to large organizations, to smaller businesses and to individuals. The philosophy behind the products of Dutch Life is that financial services should always be tailor-made. This is the main reason why, except for collective financial services, Dutch Life has chosen for the independent insurance intermediary as the main sales channel. The choice for the intermediary as the main sales channel implies that insurance intermediaries have to be stimulated to sell insurance policies of Dutch Life and therefore need to be supported in many different ways. The organizational structure of Dutch Life is displayed (partly) in figure 7.1.

![Organization chart of Dutch Life](image)

**Figure 7.1: Organization chart of Dutch Life**

186
Applying the design system: three new cases

The chart already shows that, for our case study, we zoom in on the business unit that serves the market for individual pension plans in the province of South-Holland. Although this is just a small business unit in the line organization, many of the results of the case study can be extrapolated to other business units as well. The subject of this case study was the acceptance department of this business unit. The reason why we chose this department is given below.

One of the ways of stimulating an intermediary to sell Dutch Life policies is a fast handling and processing of the applications and mutations that are sent to the back-office by the intermediary. According to Dutch Life’s management there is room for essential improvements with respect to this processing. In other words: it takes too long for an application to be processed and it is unclear why this is the case.

With this problem situation in mind, we searched for a part of the organization where the problem could be analyzed without having to bother with the complexity of all kinds of different insurance policies. We were looking for a business unit that processes a high volume of relatively simple products. It could not be one of the quotation departments because these would be incorporated in the acceptance departments in the near future and because more and more the intermediaries tend to produce quotations themselves. The choice fell on one of the acceptance departments of the business unit mentioned before.

The process of handling new applications depends on coordination between many parties inside and outside Dutch Life. According to Dutch Life’s management, the analysis of the processes should include these parties and analysis of alternatives for change should include possibilities of electronic data interchange, workflow management and IT support for the employees of Dutch Life and the intermediaries.

Before the decision was made to investigate the processes using the design system, Dutch Life did an external and an internal analysis. The analysis concerned 8 interviews with intermediaries and resulted in an assessment of the external processes. The internal analysis concerned the internal processes of offering and accepting insurance poli-
Designing Organizational Coordination

cies. The analyses provided the input for the application of the design system. One of the conclusions from the analyses as performed by Dutch Life was that in the overall processes many tasks were performed both by the intermediary and by employees of Dutch Life. After the analyses were completed, a project was formulated by Dutch Life to answer the following question:

How can coordination of processes within the acceptance departments and between the acceptance departments and the intermediaries be improved?

The answer to this question should reveal options for changing the existing processes, resulting in lower costs, shorter throughput times, shorter queues, and less administrative activities per policy. To prevent the investigation from becoming too complex, we reduced the scope of our analysis to application handling for one type of insurance. Results could be extrapolated to other departments and other types of insurance. The project was carried out by a design team, consisting of a general IT manager, a marketing manager, an organization specialist, three team leaders of acceptance teams and the author.

This problem situation provided an excellent environment to apply and test the design system. The following sections describe the results of this project along the familiar lines of conceptualization, specification, verification and validation, and solution finding.

7.2.1 Conceptualization

Before we identify the object classes, we first give a textual description of the structure and processes under investigation. Within Dutch Life, so-called acceptance teams perform the acceptance of insurance applications and mutations. The case study concerns one of those teams, consisting of 10 acceptors (of which one is the team leader) and a pre-processor. An acceptance team receives applications from the mail room of Dutch Life which we regard as a black box. The applications undergo several procedures:
Applying the design system: three new cases

Pre-processing. This task is done by the pre-processor who retrieves documents from the archive where necessary, enters data in a name registration system, and in a document guidance system for tracking and tracing. The pre-processor hands the processed applications to the team leader who distributes them among the acceptors.

Intermediary check: the acceptor checks the name and number of the intermediary in an inquiry system. If the intermediary is known as reliable, the application will continue to be processed. Otherwise, the intermediary is contacted.

Technical acceptance: the acceptor establishes whether the application can be accepted and determines the conditions against which this can be done. This may result in the following decisions:

- Accept.
- Reject and inform intermediary.
- Contact intermediary for further information (sometimes application forms contain errors that should be corrected upon acceptance).

Medical acceptance: depending on the insured value, the medical status of the client should be checked. According to very strict rules, the following decisions can be made:

- Accept.
- Send information to medical affairs and wait for advice.
- Reject and contact intermediary.

Premium calculation: Depending on the complexity of the insurance policy, the acceptor can either calculate the premium himself or he can send the application to the actuaries for further calculation.

Data entry: Once the policy has been accepted both technically and medically and the premium has been calculated, the acceptor can enter all data in one of the policy registration systems. Usually, the data is entered in the mainframe system, in which the data set will be proc-
Designing Organizational Coordination

essed. After data entry, a colleague of the acceptor checks the results for errors and confirms its correctness in the system (otherwise, he notifies his colleague). The night after the confirmation is made in the system, the data is processed and a printer produces the policy form.

**Post processing:** The policy form is returned to the acceptor. He completes it with the conditions and some brochures. The team-leader signs the document and finally it can be sent to the intermediary.

**Identify object classes**

From the above description, we derive the object classes as described in tables 7.1 to 7.4. As in the previous cases, we distinguish between actor classes, repository classes, link classes and item classes. Again, we present the classes in a tabular format. The actor classes constitute the chain through which the applications and mutations are processed, starting and ending at the intermediary. The mailroom is regarded a black box and causes a time delay for sorting and routing procedures. An acceptance team has three actor types, a team leader, several acceptors and a preprocessor. Furthermore, the team uses the computing center, the medical department and the actuaries department.

<table>
<thead>
<tr>
<th><strong>Object</strong></th>
<th><strong>Attributes</strong></th>
<th><strong>Actions</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>Identifier</td>
<td>Apply for insurance</td>
</tr>
<tr>
<td>Intermediary</td>
<td>Identifier</td>
<td>Advise client</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Send application</td>
</tr>
<tr>
<td>Mailroom</td>
<td>Identifier</td>
<td>Send application to preprocessor</td>
</tr>
<tr>
<td>Preprocessor</td>
<td>Identifier</td>
<td>Get documents from archive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enter data in systems</td>
</tr>
<tr>
<td>Team leader</td>
<td>Identifier</td>
<td>Distribute applications</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sign application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Acceptor tasks</td>
</tr>
<tr>
<td>Acceptor</td>
<td>Identifier</td>
<td>Acceptor tasks (see description)</td>
</tr>
<tr>
<td>Computing center</td>
<td>Identifier</td>
<td>Produce policy form</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Send policy form to acceptor</td>
</tr>
<tr>
<td>Medical department</td>
<td>Identifier</td>
<td>Give medical advice</td>
</tr>
<tr>
<td>Actuaries department</td>
<td>Identifier</td>
<td>Calculate premium</td>
</tr>
</tbody>
</table>

**Table 7.1: Actor classes**

190
Applying the design system: three new cases

Apart from personal workloads and stacks, there are four central repositories that are used by the teams: the policy system, the address system, the guidance system and the paper archive.

<table>
<thead>
<tr>
<th>Object</th>
<th>Attributes</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy system</td>
<td>Identifier</td>
<td>Store policy</td>
</tr>
<tr>
<td>Address system</td>
<td>Identifier</td>
<td>Store client data</td>
</tr>
<tr>
<td>Guidance system</td>
<td>Identifier</td>
<td>Store document data</td>
</tr>
<tr>
<td>Archive</td>
<td>Identifier</td>
<td>Store paper documents</td>
</tr>
</tbody>
</table>

*Table 7.2: Repository classes*

An acceptor has several systems that may produce a policy, but all information regarding insurance policies is centrally stored in the mainframe system at the computing center.

There are three types of link that are relevant to our problem situation: the Dutch mail, being the link between the intermediary and the acceptance team, the computer network, establishing the link between the teams and the computing center and the internal mail system, being the link between the acceptance team and the other departments such as medical affairs or the actuaries.

<table>
<thead>
<tr>
<th>Object</th>
<th>Attributes</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dutch mail</td>
<td>Identifier</td>
<td>Transport documents</td>
</tr>
<tr>
<td>Internal mail</td>
<td>Identifier</td>
<td>Transport documents</td>
</tr>
<tr>
<td>Computer network</td>
<td>Identifier</td>
<td>Transmit policy data</td>
</tr>
</tbody>
</table>

*Table 7.3: Link classes*

Throughout the process of accepting an insurance policy, we identify ten item types. First of all, we identify the application itself, coming from the intermediary. Once the application is received by the team, we identify data for the address system, data for the guidance system and data that is retrieved from the archive. Once an acceptor handles the application, he may send a request to medical affairs or to the actuaries and receive their answers. Furthermore, he has to enter policy data into the policy system and eventually he produces a policy form. This concludes the description of the object classes, from which we develop the conceptual models in the next subsection.
Designing Organizational Coordination

<table>
<thead>
<tr>
<th>Object</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insurance application</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>Status</td>
</tr>
<tr>
<td>Archival documents</td>
<td>Identifier</td>
</tr>
<tr>
<td>Data for address system</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td>Address</td>
</tr>
<tr>
<td>Guidance data</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>Client</td>
</tr>
<tr>
<td>Request for medical advice</td>
<td>Identifier</td>
</tr>
<tr>
<td>Medical advice</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>Advice</td>
</tr>
<tr>
<td>Request for calculation</td>
<td>Identifier</td>
</tr>
<tr>
<td>Calculation results</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>Premium</td>
</tr>
<tr>
<td>Policy data</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>Client identifier</td>
</tr>
<tr>
<td></td>
<td>type of insurance</td>
</tr>
<tr>
<td>Policy form / copy of policy form</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>Policy data</td>
</tr>
</tbody>
</table>

Table 7.4: Item classes

Construct conceptual model

The objects as identified above form the basis of the network model which represents the chain through which the applications are communicated and processed.

Figure 7.2: Network model from the macro-perspective
Applying the design system: three new cases

The network models we have seen in the previous case studies were not very complicated and could be visualized using one diagram. To get a clear picture of the organizational networks in this situation, we first sketch the network model from a macro-perspective and then zoom in to a network model of the relevant parts of Dutch Life, looking from the meso-perspective. The former is displayed in figure 7.2, the latter in figure 7.3.

![Network Model Diagram]

**Figure 7.3: Network model from the meso-perspective**

The process model is displayed in figure 7.4 and depicts the processes carried out at Dutch Life, from the point when an application enters the organization at the mail room, until the policy leaves the organization via the mail room. Some of the processes can be further specified in actor models. Of the actor models made, we give the one for the team leader as an example in figure 7.5.
Designing Organizational Coordination

Figure 7.4: Process model of the Dutch Life case

194
Applying the design system: three new cases

For the team leader, the other activities mentioned in the actor model include acceptors tasks, staff meetings and special projects.

![Diagram showing a flowchart of a team leader's tasks]

**Figure 7.5: Actor model of a team leader**

The conceptual models were used as a basis for discussions with the members of the team that was being modeled, and with Dutch Life’s management. After two discussion sessions with Dutch Life management and three team leaders of comparable acceptance teams, we concluded that the level of detail was sufficient and that the models represented the situation of the insurance chain and the teams within Dutch Life correctly. Based on the conceptual models and the experience of the modeling process we reformulated the problem and constructed a simulation model, assisted by Dutch Life’s management and the three team leaders.
Designing Organizational Coordination

Formulate problem statement

After conceptualization, and in agreement with Dutch Life's management, we formulated the problem as follows:

What measures should be taken to improve the internal coordination of the actors involved in the acceptance process and the external coordination between Dutch Life and the intermediaries? What effect do these improvements have on logistic indicators such as throughput times, workloads, direct production costs and handling times related to producing a policy?

7.2.2 Specification

Based on the conceptual models and the problem formulation, a simulation model was developed. Parts of the simulation template were used for the construction of the model. The specification was carried out by the team that was also involved in conceptualization and included a representation of Dutch Life's management and the three team-leaders mentioned before. We discuss the specification in the same way as we did with the previous case studies.

Output variables

Dutch Life keeps track of several performance indicators, so a numerical validation of the model would be facilitated by incorporating the same indicators as output variables in the model. The indicators that were accessible from Dutch Life records concerned throughput time and direct policy costs. Dutch Life identifies three types of throughput time:

1. Total throughput time, incorporating the whole chain from the moment that the intermediary produces the application to the moment that the consumer receives his policy.
2. Internal throughput time, incorporating everything that happens from the entrance of an application at the mailroom until the signed policy leaves the building.
Applying the design system: three new cases

3. Team throughput time, including the time an application remains with the acceptance team (this excludes the time that an application is handled by the medical department, the actuaries or the time that is needed for further written communication with the intermediary).

The latter two indicators and direct policy costs are kept track of by Dutch Life. Next to these indicators, Dutch Life's management was highly interested in the workloads of the teams, the number of policies that require further written communication with the intermediary (called agenda documents) and the percentage of time an acceptor is engaged in handling applications for new insurance policies (the remaining time is needed for handling mutations, meetings, other activities and breaks).

A discussion of the indicators above with Dutch Life management and team members led to the following output variables for the simulation model:

- Direct policy costs, including all costs that are directly related to the production of one policy.
- Total throughput time.
- Internal throughput time.
- Team throughput time.
- Percentage of time required for new applications.
- Workload.
- Number of agenda documents.

The internal throughput time and the team throughput time will be used for replicative validation. These are the only variables for which empirical values can be collected through the analysis of Dutch Life records.

According to Dutch Life Management, a reduction of the throughput times is the main issue for which alternative solutions should be generated.
Designing Organizational Coordination

Model reductions

In order to arrive at a simulation model that shows good correspondence to reality and in order to avoid unnecessary complexity, the design team was engaged in a discussion of model reductions. In this discussion, we concluded that it was important for internal support for the model that it should incorporate the whole chain, including the intermediary. The design team identified two types of reductions:

Structural reductions, where certain entities from reality are regarded as black boxes:

- The mail room is represented as an accumulating queue that is emptied each morning at 9 o'clock.
- The group of intermediaries that send applications to the team are reduced to one intermediary who is responsible for all applications.
- The medical department and the calculations department are represented as storages that hold documents for a certain period.
- Documents that end up in the archive or at the client are removed from the system (after gathering statistics).
- The retrieval of documents from the archive and the data entry in the address and guidance systems are represented by delays at the pre-processor. These systems do not figure in the simulation model.

Numerical reductions, where handling times, routing times and decisions are reduced to stochastic values:

- Document generation is an exponential inter-arrival process with a frequency of 11 applications per day.
- Handling times are reduced to triangular or continuous distributions, based on an assessment of team members and team leaders.
- The routing time of documents is represented as a triangular distribution, the documents are batched in queues and released according to the times when the internal mail employees make their rounds through the building.
- All decisions are stochastic values, assessed by the team members.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of applications generated (the interarrival time is exponentially distributed over the day)</td>
<td>tria(8,11,15)</td>
</tr>
<tr>
<td>Handling time at intermediary for all documents</td>
<td>15 minutes</td>
</tr>
<tr>
<td>Route time from intermediary to Dutch life</td>
<td>expo(12) hours (during the night)</td>
</tr>
<tr>
<td>Direct costs per hour per employee</td>
<td>DFL 75,= except medical affairs.</td>
</tr>
<tr>
<td><strong>Preprocessor</strong></td>
<td></td>
</tr>
<tr>
<td>Time for input in name system and archive retrieval</td>
<td>cont(0.1,3,0.9,4,1,6) minutes</td>
</tr>
<tr>
<td>Time for input in guidance system</td>
<td>cont(0.1,4,0.9,5,1,7) minutes</td>
</tr>
<tr>
<td>Time for routing documents to team leader</td>
<td>1 minute per daily batch</td>
</tr>
<tr>
<td><strong>Team leader</strong></td>
<td></td>
</tr>
<tr>
<td>Time for assigning a document to a team member</td>
<td>1 minute</td>
</tr>
<tr>
<td>Time for checking and signing a policy</td>
<td>2 minutes</td>
</tr>
<tr>
<td>Probability of errors in policy</td>
<td>4 percent</td>
</tr>
<tr>
<td>Route time to team member (per daily batch)</td>
<td>1 minute</td>
</tr>
<tr>
<td><strong>Team member</strong></td>
<td></td>
</tr>
<tr>
<td>Time for intermediary check</td>
<td>cont(0.95,1,1,5) minutes</td>
</tr>
<tr>
<td>Probability of putting document aside (agenda)</td>
<td>5 percent</td>
</tr>
<tr>
<td>Time until intermediary reacts on document</td>
<td>cont(0.8,5,1,20) days</td>
</tr>
<tr>
<td>Time for technical acceptance</td>
<td>1 minute</td>
</tr>
<tr>
<td>Probability of putting document aside (agenda)</td>
<td>20 percent</td>
</tr>
<tr>
<td>Probability of making a phone call to the intermediary</td>
<td>10 percent</td>
</tr>
<tr>
<td>Time until intermediary reacts on document</td>
<td>tria(7,10,13) days</td>
</tr>
<tr>
<td>Extra time for phone call</td>
<td>tria(5,7,10) minutes</td>
</tr>
<tr>
<td>Time for medical acceptance</td>
<td>cont(0.6,1,15) minutes</td>
</tr>
<tr>
<td>Probability of involvement medical affairs</td>
<td>35 percent</td>
</tr>
<tr>
<td>Time for checking the input of a colleague</td>
<td>tria(4,10,20) minutes</td>
</tr>
<tr>
<td>Probability of errors</td>
<td>5 percent</td>
</tr>
<tr>
<td>Time for calculating premium</td>
<td>1 minute</td>
</tr>
<tr>
<td>Probability of involvement actuaries</td>
<td>10 percent</td>
</tr>
<tr>
<td>Time for data entry</td>
<td>tria(7,15,30) minutes</td>
</tr>
<tr>
<td>Time for completion of policy with supplements</td>
<td>10 minutes</td>
</tr>
<tr>
<td>Probability of errors in output</td>
<td>1 percent</td>
</tr>
<tr>
<td>Time for packing and sending</td>
<td>4 minutes</td>
</tr>
<tr>
<td><strong>Medical affairs</strong></td>
<td></td>
</tr>
<tr>
<td>Costs</td>
<td>disc(0.85,29.20,1,219.60) DFL.</td>
</tr>
<tr>
<td>Time</td>
<td>tria(3,16,40) days</td>
</tr>
<tr>
<td><strong>Actuaries</strong></td>
<td></td>
</tr>
<tr>
<td>Time for calculation</td>
<td>tria(10,15,20) minutes</td>
</tr>
<tr>
<td>Waiting and routing time</td>
<td>2 days (1 back, 1 forth)</td>
</tr>
</tbody>
</table>

*Table 7.5: Input data for the Dutch Life case*
Designing Organizational Coordination

Input data

The input data for costs were obtained from Dutch Life records. The other input variables were based on assessments by team members and team leaders. In table 7.5 we present the input values. Input data is given for each of the actors present in the model.

Simulation model and animation

The network model served as a basis for the construction of the simulation model. The model consists of code blocks for each actor in the network models. The acceptors are implemented as one generic code block of the acceptor. An attribute in each item (document) holds the identification of the acceptor to which it is related. A status attribute determines the next activity that has to be carried out on the item. Based on this attribute(s), the acceptor decides what activity to perform on an incoming item. The dialog screen is shown in figure 7.6.

![Figure 7.6: Dialog screen of the acceptor](image-url)
Applying the design system: three new cases

The simulation model has three animation screens: a screen for the intermediary, a screen for Dutch Life's processing and a screen for the display of statistics like throughput time, average cost per policy and workload levels. The screen for Dutch Life's processing is given below.

![Figure 7.7: Screen dump of the processing at Dutch Life](image)

In figure 7.7 we see the preprocessor and the team leader in the upper section and the team members in the lower section. In the middle, the actuary and the department of medical affairs is situated as well as the policy system and the archive.

The screen for the intermediary consists of two parts, one for the interaction between the intermediary and his clients and one for the administrative processing of application forms and policies on the part of the intermediary.
Designing Organizational Coordination

Initial treatment

For the initial treatment, we have to establish the start-up time, the run-length and the number of replications needed to achieve results that achieve a certain level of confidence. Like in the previous case, the system is non-terminating. We estimate the start-up time from a moving average plot of the measured internal and total throughput times, Figure 7.8 shows the two moving average plots. We conclude from this that a start-up time of 50 days is sufficient. As cycle times may exceed 25 days, a run length of 100 days seemed appropriate.

![Moving average throughput times](image)

**Figure 7.8: Moving average throughput times**

After an initial number of 10 replications, we determined the number of replications needed for a treatment by means of formula (1) and the procedure as sketched in section 6.2.2. Table 7.6 shows the results of this procedure for three output variables and a confidence level of 95%. The delta values are based on the variation in values from Dutch Life files. From this table, we conclude that a total number of 20 replications is sufficient. Ten extra replications were performed.
Applying the design system: three new cases

<table>
<thead>
<tr>
<th>Variable (results after 10 replications)</th>
<th>Mean</th>
<th>Standard deviation</th>
<th>Delta</th>
<th>Minimum # replications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct policy costs</td>
<td>DFL. 104.75</td>
<td>1.57</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>Total throughput time</td>
<td>16.94 (days)</td>
<td>0.42</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Number of agenda documents</td>
<td>35.93</td>
<td>4.10</td>
<td>5</td>
<td>14</td>
</tr>
</tbody>
</table>

Table 7.6: Determining the number of replications

7.2.3 Verification and validation

Verification and validation was carried out in the same way as in the previous case study. Below we briefly discuss both.

Verification

The individual parts of the model as given in the model description above were tested for their behavior by means of a numerical analysis of their functioning and a check of the program code. The animation was helpful in finding errors in the simulation code.

The code was tested by manual walkthroughs and by letting individual items follow parts of the code and testing variable values afterwards. The design team reserved a full afternoon for the simulation model walkthrough. Each input value was evaluated and agreed upon. Small changes were made to parts of the model code and some input values (mainly handling times). For further information we refer to Eijck and Sol (1994).

Validation

The model was validated by means of replicative validation and by expert validation. Simulation results and the animation were discussed with the design team and with members of the acceptance teams. The discussions led to the fine tuning of several input values.

A replicative validation was carried out by performing a t-test on the difference between the means of the output variables for 20 replica-
Designing Organizational Coordination

tions and the means of empirical values found in Dutch Life records. Dutch life provided a file with internal and team throughput times in which observations are recorded for each policy over a period of one year. As the model generates approximately 500 throughput times per replication, we produced 10 samples of 500 throughput times, randomly taken from the Dutch Life file, and calculated the mean values for each sample, resulting in 10 means and a standard deviation over these means.

Table 7.7 shows the results of the replicative validation where index 1 refers to the samples from Dutch Life records and index 2 to the 20 replications of the model. The level of confidence is 95 %, resulting in a t-value of $t_{wa}(n_1 + n_2 - 2) = t_{0.025}(28) = 2.048$. The $|t|$ values give the half-with of the 95 % confidence intervals of $d / s_d$. Based on these values we conclude that there is no significant difference between model and reality for these variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>$x_1$</th>
<th>$s_1^2$</th>
<th>$x_2$</th>
<th>$s_2^2$</th>
<th>$d$</th>
<th>$s_d$</th>
<th>$t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal throughput t.</td>
<td>15</td>
<td>50.63</td>
<td>12.53</td>
<td>0.12</td>
<td>2.47</td>
<td>1.57</td>
<td>1.57</td>
</tr>
<tr>
<td>Team throughput t.</td>
<td>5</td>
<td>4.97</td>
<td>4.43</td>
<td>0.007</td>
<td>0.57</td>
<td>0.49</td>
<td>1.16</td>
</tr>
<tr>
<td>$\alpha = 0.05, n_1 = 10, n_2 = 20, t_{wa}(n_1 + n_2 - 2) = 2.048$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table 7.7: Replicative validation results*

7.2.4 Solution finding and consistency check

Dutch Life management was presented with the animation and the first model results. Based on these results, the design team and Dutch Life management discussed possible alternative strategies. The model showed that documents were staying at medical affairs and the calculations department for a long time (up to 40 days for medical affairs). Furthermore, an improvement in throughput time seemed possible by reducing the number of documents in need of extra communication with the intermediaries (the agenda documents). The design team also concluded that the day’s waiting between data entry and input checking was unnecessary.
Applying the design system: three new cases

With these findings in mind, the design team proposed to run four treatments to analyze the effects of eliminating the problems mentioned above. We refer to these treatments as internal alternatives, as they concern changes in the internal organization of Dutch Life. The treatments are described in table 7.8.

<table>
<thead>
<tr>
<th>Treatment</th>
<th>Alternative</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Current situation</td>
</tr>
<tr>
<td>1</td>
<td>Bypass the department of medical affairs in all cases</td>
</tr>
<tr>
<td>2</td>
<td>Bypass the actuaries department in all cases</td>
</tr>
<tr>
<td>3</td>
<td>No agenda documents</td>
</tr>
<tr>
<td>4</td>
<td>Checking phase immediately after data entry</td>
</tr>
</tbody>
</table>

Table 7.8: Treatments for internal change

In treatment 1, the influence of medical affairs is analyzed, by changing the model in such a way that no applications need medical advice from this department. In treatment 2, we do the same for the actuaries department. Treatment 3 analyzes the influence of the fact that in many cases the acceptors need to communicate in writing with the intermediary. Finally, treatment 4 analyzes the effect of data entry checking by a colleague.

Next to the 'internal' alternatives, the design team proposed three sets of so-called 'chain' alternatives. The first set concerns the application of electronic data transportation between the intermediary and the acceptance teams. This means that the intermediary enters the data for an insurance application in a computer file, which is transmitted to the acceptance departments for further processing. We assume that handling times do not change under this alternative. We devised three treatments, one where the alternative applies to 20 % of the applications, one for 50 %, and one for 100 % of the applications.

The second set of treatments concerns an extension to the first set, namely the electronic acceptance of policies after electronic transmission. According to the results of prototype acceptance systems, we estimated that about 35 % of the incoming applications could be accepted electronically. We also differentiated between 20 %, 50 % and 100 % of the applications being transmitted electronically.
Designing Organizational Coordination

The third set of treatments relates to a further extension: we assume that the results of the acceptance process go back to the intermediary in an electronic format. The intermediary then prints and signs the policy form at his office. In the best case, the client may leave the intermediary with the policy in his possession. Table 7.9 shows a summary of the three treatment sets. Each set consists of a treatment where the alternative is applied to 20 % (a) of the policy applications, a treatment for 50 % (b) of the applications and a treatment for 100 % (c) of the applications.

<table>
<thead>
<tr>
<th>Treatment</th>
<th>Alternative</th>
</tr>
</thead>
<tbody>
<tr>
<td>5a,5b,5c</td>
<td>Electronic data entry by the intermediary.</td>
</tr>
<tr>
<td></td>
<td>(20, 50 and 100 % of the applications)</td>
</tr>
<tr>
<td>6a,6b,6c</td>
<td>Electronic data entry by the intermediary and electronic acceptance.</td>
</tr>
<tr>
<td></td>
<td>(20, 50 and 100 % of the applications)</td>
</tr>
<tr>
<td>7a,7b,7c</td>
<td>Electronic data entry by the intermediary, electronic acceptance and policy</td>
</tr>
<tr>
<td></td>
<td>printing at the intermediary.</td>
</tr>
<tr>
<td></td>
<td>(20, 50 and 100 % of the applications)</td>
</tr>
</tbody>
</table>

*Table 7.9: Treatments for changing the chain*

Consistency

The alternatives will not render the respective simulation models inconsistent with the conceptual models as shown in section 7.2.1. As in the previous case study, the changes can be made by means of adding extra links, tasks, decisions and changing input data.

7.2.5 Results

Treatment 0

Treatment 0 concerns the current situation and serves as a yardstick for all other treatments. The other treatments are tested on significant change relative to treatment 0 by means of a t-test. In the tables, we note the output variables, their 95 % confidence intervals, the |t| values and the related significance. For treatment 0 we only note the variable and the confidence interval. The results of treatment 0 are displayed in table 7.10.
Applying the design system: three new cases

<table>
<thead>
<tr>
<th>Output variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct policy costs (DFL.)</td>
<td>104.75 ± 0.78</td>
</tr>
<tr>
<td>Total throughput time (days)</td>
<td>17.07 ± 0.16</td>
</tr>
<tr>
<td>Internal throughput time (days)</td>
<td>12.52 ± 0.16</td>
</tr>
<tr>
<td>Team throughput time (days)</td>
<td>4.43 ± 0.04</td>
</tr>
<tr>
<td>Percentage of time required for new applications (%)</td>
<td>12.38 ± 0.519</td>
</tr>
<tr>
<td>Workload</td>
<td>139.6 ± 2.2</td>
</tr>
<tr>
<td>Number of agenda documents</td>
<td>35.9 ± 1.5</td>
</tr>
</tbody>
</table>

\( \alpha = 0.05, n = 20 \), values determine 95% confidence intervals

**Table 7.10: Results of treatment 0, current situation**

**Treatment 1**

Treatment 1 concerns the bypass of the medical affairs department in all cases. Part of this alternative could be realized by increasing the insured value above which an advice from medical affairs is imperative. The results are displayed in table 7.11.

<table>
<thead>
<tr>
<th>Output variable</th>
<th>Value (rel. change)</th>
<th>t t l</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct policy costs</td>
<td>87.29 ± 0.36 (-18%)</td>
<td>42.4</td>
<td>YES</td>
</tr>
<tr>
<td>Total throughput time</td>
<td>10.72 ± 0.10 (-38%)</td>
<td>70.3</td>
<td>YES</td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>6.11 ± 0.10 (-52%)</td>
<td>70.9</td>
<td>YES</td>
</tr>
<tr>
<td>Team throughput time</td>
<td>3.05 ± 0.03 (-30%)</td>
<td>59.2</td>
<td>YES</td>
</tr>
<tr>
<td>Percentage of time for new applications</td>
<td>11.84 ± 0.48</td>
<td>1.6</td>
<td>NO</td>
</tr>
<tr>
<td>Workload</td>
<td>65.4 ± 1.3 (-54%)</td>
<td>60.9</td>
<td>YES</td>
</tr>
<tr>
<td>Number of agenda documents</td>
<td>33.5 ± 1.3 (-6%)</td>
<td>2.5</td>
<td>YES</td>
</tr>
</tbody>
</table>

\( \alpha = 0.05, n_1 = 20, n_2 = 20, t_{0.05}(n_1 + n_2 - 2) = t_{0.05}(38) = 2.021 \)

**Table 7.11: Results of treatment 1, bypass of medical affairs**

We see that in this alternative, although the relative time used for new applications does not change significantly, the throughput times drop as well as the workloads and the policy costs.

**Treatment 2**

Treatment 2 concerns the bypassing of the actuaries. This means that for all applications, the calculations are done by the acceptor. To implement this alternative, acceptors have to be trained and provided with the computer programs that support the calculation task.
Designing Organizational Coordination

<table>
<thead>
<tr>
<th>Output variable</th>
<th>Value (rel. change)</th>
<th>t t</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct policy costs</td>
<td>101.19 ± 0.64 (-4 %)</td>
<td>7.4</td>
<td>YES</td>
</tr>
<tr>
<td>Total throughput time</td>
<td>16.50 ± 0.17 (-3 %)</td>
<td>5.0</td>
<td>YES</td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>11.94 ± 0.19 (-5 %)</td>
<td>4.9</td>
<td>YES</td>
</tr>
<tr>
<td>Team throughput time</td>
<td>3.96 ± 0.03 (-12 %)</td>
<td>19.2</td>
<td>YES</td>
</tr>
<tr>
<td>Percentage of time for new applications</td>
<td>11.77 ± 0.48</td>
<td>1.6</td>
<td>NO</td>
</tr>
<tr>
<td>Workload</td>
<td>131.9 ± 1.9 (-6 %)</td>
<td>60.9</td>
<td>YES</td>
</tr>
<tr>
<td>Number of agenda documents</td>
<td>35.1 ± 1.4</td>
<td>1.4</td>
<td>NO</td>
</tr>
</tbody>
</table>

\[ \alpha = 0.05, n_1 = 20, n_2 = 20, t_{\alpha}(n_1 + n_2 - 2) = t_{0.05}(38) = 2.021 \]

**Table 7.12: Results of treatment 2, bypass of the actuaries**

Here we see that the improvements are not as spectacular as in the first treatment. The medical affairs department has a greater influence on the system than the actuaries. This is probably due to the fact that acceptors already carry out many calculations themselves.

**Treatment 3**

The third treatment concerns the influence of errors made by intermediaries. For this treatment we assume that it is possible to achieve a situation where all incoming application forms are error-free. This might partly be done by supporting the intermediary with software that helps him fill out the application form. The results of this treatment are displayed in table 7.13.

<table>
<thead>
<tr>
<th>Output variable</th>
<th>Value (rel. change)</th>
<th>t t</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct policy costs</td>
<td>103.96 ± 0.66</td>
<td>1.6</td>
<td>NO</td>
</tr>
<tr>
<td>Total throughput time</td>
<td>14.05 ± 0.21 (-18 %)</td>
<td>23.37</td>
<td>YES</td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>9.41 ± 0.23 (-25 %)</td>
<td>23.3</td>
<td>YES</td>
</tr>
<tr>
<td>Team throughput time</td>
<td>4.38 ± 0.05</td>
<td>1.6</td>
<td>NO</td>
</tr>
<tr>
<td>Percentage of time for new applications</td>
<td>11.73 ± 0.50</td>
<td>1.9</td>
<td>NO</td>
</tr>
<tr>
<td>Workload</td>
<td>101.1 ± 2.1 (-17 %)</td>
<td>26.2</td>
<td>YES</td>
</tr>
<tr>
<td>Number of agenda documents</td>
<td>0 ± 0 (-100 %)</td>
<td>50.7</td>
<td>YES</td>
</tr>
</tbody>
</table>

\[ \alpha = 0.05, n_1 = 20, n_2 = 20, t_{\alpha}(n_1 + n_2 - 2) = t_{0.05}(38) = 2.021 \]

**Table 7.13: Results of treatment 3, no agenda documents**

In this treatment we see a relative improvement in throughput times and workloads (logically the number of agenda documents is zero).
Applying the design system: three new cases

However, there is no significant improvement in policy costs and team throughput time. This means that the amount of extra work an acceptor has to do on erroneous applications has far less influence than the extra time the applications spend in the agenda. Time that is not considered as throughput time for the team.

Treatment 4

Treatment 4 concerns the acceleration of application processing by cutting out the night between data entry by the acceptor and data checking by a colleague. Logically, you would expect a small decrease in throughput time and a small decrease in workload, but no change in the other variables. Table 7.14 shows that this is exactly the case.

<table>
<thead>
<tr>
<th>Output variable</th>
<th>Value (rel. change)</th>
<th>t t l</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct policy costs</td>
<td>104.09 ± 0.79</td>
<td>1.2</td>
<td>NO</td>
</tr>
<tr>
<td>Total throughput time</td>
<td>15.76 ± 0.23 (-8 %)</td>
<td>9.9</td>
<td>YES</td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>11.19 ± 0.20 (-11 %)</td>
<td>10.8</td>
<td>YES</td>
</tr>
<tr>
<td>Team throughput time</td>
<td>3.17 ± 0.04 (-28 %)</td>
<td>45.0</td>
<td>YES</td>
</tr>
<tr>
<td>Percentage of time for new applications</td>
<td>12.33 ± 0.55</td>
<td>0.16</td>
<td>NO</td>
</tr>
<tr>
<td>Workload</td>
<td>123.3 ± 1.9 (-12 %)</td>
<td>11.8</td>
<td>YES</td>
</tr>
<tr>
<td>Number of agenda documents</td>
<td>34.6 ± 1.4</td>
<td>1.36</td>
<td>NO</td>
</tr>
</tbody>
</table>

Table 7.14: Results of treatment 4, checking policies instantly

This concludes the results of the so called 'internal' alternatives. It is evident that, as far as the internal procedures are concerned, medical affairs is the main bottleneck (treatment 1). Bypassing medical affairs results in improvements of throughput time of up to 52 percent.

Treatments 5a, 5b and 5c

Treatment set 5 concerns the electronic delivery of application data by the intermediary via the Dutch Insurance Data Network. We assume that when the data arrive at Dutch Life, they are printed and manually inserted into the in-house systems by the acceptors. The results can be read from tables 7.15a, 7.15b and 7.15c, for each of the percentages assumed (20 %, 50 % and 100 %).
Designing Organizational Coordination

<table>
<thead>
<tr>
<th>Output variable (20 %)</th>
<th>Value (rel. change)</th>
<th>t</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct policy costs</td>
<td>97.17 ± 0.69 (-8 %)</td>
<td>15.1</td>
<td>YES</td>
</tr>
<tr>
<td>Total throughput time</td>
<td>15.29 ± 0.24 (-10 %)</td>
<td>12.6</td>
<td>YES</td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>11.56 ± 0.24 (-7 %)</td>
<td>6.9</td>
<td>YES</td>
</tr>
<tr>
<td>Team throughput time</td>
<td>4.23 ± 0.05 (-5 %)</td>
<td>6.9</td>
<td>YES</td>
</tr>
<tr>
<td>Percentage of time for new applications</td>
<td>11.29 ± 0.50 (-9 %)</td>
<td>3.3</td>
<td>YES</td>
</tr>
<tr>
<td>Workload</td>
<td>123.8 ± 2.4 (-10 %)</td>
<td>9.0</td>
<td>YES</td>
</tr>
<tr>
<td>Number of agenda documents</td>
<td>27.7 ± 1.3 (-25 %)</td>
<td>8.7</td>
<td>YES</td>
</tr>
</tbody>
</table>

\[ \alpha = 0.05, n_1 = 20, n_2 = 20, t_{\text{cal}}(n_1 + n_2 - 2) = t_{\text{cal}}(38) = 2.021 \]

Table 7.15a: Results of treatment 5a, 20% electronic data entry

<table>
<thead>
<tr>
<th>Output variable (50 %)</th>
<th>Value (rel. change)</th>
<th>t</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct policy costs</td>
<td>86.67 ± 0.85 (-17 %)</td>
<td>15.1</td>
<td>YES</td>
</tr>
<tr>
<td>Total throughput time</td>
<td>13.29 ± 0.20 (-22 %)</td>
<td>30.8</td>
<td>YES</td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>10.47 ± 0.19 (-16 %)</td>
<td>17.0</td>
<td>YES</td>
</tr>
<tr>
<td>Team throughput time</td>
<td>4.01 ± 0.05 (-9 %)</td>
<td>13.9</td>
<td>YES</td>
</tr>
<tr>
<td>Percentage of time for new applications</td>
<td>10.21 ± 0.49 (-17 %)</td>
<td>6.3</td>
<td>YES</td>
</tr>
<tr>
<td>Workload</td>
<td>112.3 ± 1.8 (-19 %)</td>
<td>20.2</td>
<td>YES</td>
</tr>
<tr>
<td>Number of agenda documents</td>
<td>18.6 ± 0.7 (-50 %)</td>
<td>21.9</td>
<td>YES</td>
</tr>
</tbody>
</table>

\[ \alpha = 0.05, n_1 = 20, n_2 = 20, t_{\text{cal}}(n_1 + n_2 - 2) = t_{\text{cal}}(38) = 2.021 \]

Table 7.15b: Results of treatment 5b, 50% electronic data entry

<table>
<thead>
<tr>
<th>Output variable (100 %)</th>
<th>Value (rel. change)</th>
<th>t</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct policy costs</td>
<td>69.70 ± 0.68 (-33 %)</td>
<td>70.7</td>
<td>YES</td>
</tr>
<tr>
<td>Total throughput time</td>
<td>10.68 ± 0.15 (-37 %)</td>
<td>60.55</td>
<td>YES</td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>8.46 ± 0.15 (-32 %)</td>
<td>38.2</td>
<td>YES</td>
</tr>
<tr>
<td>Team throughput time</td>
<td>3.61 ± 0.03 (-18 %)</td>
<td>33.8</td>
<td>YES</td>
</tr>
<tr>
<td>Percentage of time for new applications</td>
<td>7.95 ± 0.41 (-36 %)</td>
<td>14.0</td>
<td>YES</td>
</tr>
<tr>
<td>Workload</td>
<td>85.8 ± 1.8 (-39 %)</td>
<td>40.1</td>
<td>YES</td>
</tr>
<tr>
<td>Number of agenda documents</td>
<td>1.1 ± 0.23 (-97 %)</td>
<td>48.6</td>
<td>YES</td>
</tr>
</tbody>
</table>

\[ \alpha = 0.05, n_1 = 20, n_2 = 20, t_{\text{cal}}(n_1 + n_2 - 2) = t_{\text{cal}}(38) = 2.021 \]

Table 7.15c: Results of treatment 5c, 100% electr. data entry

For all these alternatives, we see the improvement increasing with the percentage of electronically delivered applications. Especially the agenda documents tend to reduce in volume rather dramatically. Also, the effect on throughput time is already very substantial. However, we will see more improvement in the next set of treatments.
Applying the design system: three new cases

Treatments 6a, 6b and 6c

The results of the second ‘external’ alternative, concerning the electronic acceptance of applications are displayed in tables 7.16a - 7.16c.

<table>
<thead>
<tr>
<th>Output variable (20 %)</th>
<th>Value (rel. change)</th>
<th>t</th>
<th>t</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct policy costs</td>
<td>96.31 ± 0.65 (-9 %)</td>
<td>17.4</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Total throughput time</td>
<td>15.42 ± 0.18 (-9 %)</td>
<td>13.9</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>11.64 ± 0.15 (-7 %)</td>
<td>8.0</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Team throughput time</td>
<td>4.22 ± 0.03 (-5 %)</td>
<td>8.7</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Percentage of time for new apps</td>
<td>11.09 ± 0.04 (-10 %)</td>
<td>4.2</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Workload</td>
<td>125.8 ± 2.2 (-22 %)</td>
<td>9.4</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Number of agenda documents</td>
<td>28.8 ± 1.3 (-25 %)</td>
<td>7.5</td>
<td>YES</td>
<td></td>
</tr>
</tbody>
</table>

α = 0.05, n₁ = 20, n₂ = 20, \( t_{\text{val}}(n₁ + n₂) = t_{\text{val}}(38) = 2.021 \)

**Table 7.16a: Results of treatment 6a, 20% electronic acceptance**

<table>
<thead>
<tr>
<th>Output variable (50 %)</th>
<th>Value (rel. change)</th>
<th>t</th>
<th>t</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct policy costs</td>
<td>85.50 ± 0.62 (-18 %)</td>
<td>40.4</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Total throughput time</td>
<td>13.08 ± 0.19 (-20 %)</td>
<td>33.4</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>10.22 ± 0.18 (-21 %)</td>
<td>19.7</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Team throughput time</td>
<td>3.91 ± 0.04 (-53 %)</td>
<td>19.5</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Percentage of time for new apps</td>
<td>9.94 ± 0.49 (-24 %)</td>
<td>7.17</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Workload</td>
<td>110.7 ± 2.4 (-18 %)</td>
<td>18.8</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Number of agenda documents</td>
<td>17.8 ± 0.8 (-11 %)</td>
<td>22.6</td>
<td>YES</td>
<td></td>
</tr>
</tbody>
</table>

α = 0.05, n₁ = 20, n₂ = 20, \( t_{\text{val}}(n₁ + n₂) = t_{\text{val}}(38) = 2.021 \)

**Table 7.16b: Results of treatment 6b, 50% electronic acceptance**

<table>
<thead>
<tr>
<th>Output variable (100 %)</th>
<th>Value (rel. change)</th>
<th>t</th>
<th>t</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct policy costs</td>
<td>68.23 ± 0.73 (-35 %)</td>
<td>71.0</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Total throughput time</td>
<td>10.56 ± 0.18 (-38 %)</td>
<td>55.9</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>8.30 ± 0.18 (-34 %)</td>
<td>36.6</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Team throughput time</td>
<td>3.51 ± 0.03 (-18 %)</td>
<td>38.3</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Percentage of time for new apps</td>
<td>7.8 ± 0.53 (-37 %)</td>
<td>13.0</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Workload</td>
<td>85.7 ± 1.7 (-39 %)</td>
<td>41.0</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Number of agenda documents</td>
<td>1.2 ± 0.2 (-97 %)</td>
<td>48.4</td>
<td>YES</td>
<td></td>
</tr>
</tbody>
</table>

α = 0.05, n₁ = 20, n₂ = 20, \( t_{\text{val}}(n₁ + n₂) = t_{\text{val}}(38) = 2.021 \)

**Table 7.16c: Results of treatment 6c, 100% electr. acceptance**

211
Designing Organizational Coordination

Treatments 7a, 7b and 7c

The results of the third ‘external’ alternative, where the intermediary is responsible for printing the policy, are displayed in tables 7.17a-c.

<table>
<thead>
<tr>
<th>Output variable (20 %)</th>
<th>Value (rel. change)</th>
<th>t t l</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct policy costs</td>
<td>87.16 ± 0.75 (-17 %)</td>
<td>33.8</td>
<td>YES</td>
</tr>
<tr>
<td>Total throughput time</td>
<td>9.28 ± 0.15 (-45 %)</td>
<td>72.3</td>
<td>YES</td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>5.95 ± 0.15 (-53 %)</td>
<td>61.1</td>
<td>YES</td>
</tr>
<tr>
<td>Team throughput time</td>
<td>1.95 ± 0.04 (-57 %)</td>
<td>90.2</td>
<td>YES</td>
</tr>
<tr>
<td>Percentage of time for new applications</td>
<td>8.83 ± 0.56 (-29 %)</td>
<td>9.7</td>
<td>YES</td>
</tr>
<tr>
<td>Workload</td>
<td>62.9 ± 1.8 (-54 %)</td>
<td>55.5</td>
<td>YES</td>
</tr>
<tr>
<td>Number of agenda documents</td>
<td>6.2 ± 0.5 (-83 %)</td>
<td>39.6</td>
<td>YES</td>
</tr>
<tr>
<td>(\alpha = 0.05, n_1 = 20, n_2 = 20, t_{\text{val}}(n_1 + n_2 - 2) = t_{\text{val}}(38) = 2.021)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 7.17a:** Results of treatment 7a, 20% printing at agent

<table>
<thead>
<tr>
<th>Output variable (50 %)</th>
<th>Value (rel. change)</th>
<th>t t l</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct policy costs</td>
<td>72.86 ± 0.80 (-31 %)</td>
<td>59.5</td>
<td>YES</td>
</tr>
<tr>
<td>Total throughput time</td>
<td>7.52 ± 0.13 (-56 %)</td>
<td>95.4</td>
<td>YES</td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>5.52 ± 0.13 (-56 %)</td>
<td>69.6</td>
<td>YES</td>
</tr>
<tr>
<td>Team throughput time</td>
<td>1.83 ± 0.04 (-59 %)</td>
<td>97.8</td>
<td>YES</td>
</tr>
<tr>
<td>Percentage of time for new applications</td>
<td>6.94 ± 0.39 (-44 %)</td>
<td>17.6</td>
<td>YES</td>
</tr>
<tr>
<td>Workload</td>
<td>61.0 ± 2.0 (-57 %)</td>
<td>55.9</td>
<td>YES</td>
</tr>
<tr>
<td>Number of agenda documents</td>
<td>3.9 ± 0.4 (-92 %)</td>
<td>43.3</td>
<td>YES</td>
</tr>
<tr>
<td>(\alpha = 0.05, n_1 = 20, n_2 = 20, t_{\text{val}}(n_1 + n_2 - 2) = t_{\text{val}}(38) = 2.021)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 7.17b:** Results of treatment 7b, 50% printing at agent

<table>
<thead>
<tr>
<th>Output variable (100 %)</th>
<th>Value (rel. change)</th>
<th>t t l</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct policy costs</td>
<td>48.61 ± 0.64 (-54 %)</td>
<td>115.9</td>
<td>YES</td>
</tr>
<tr>
<td>Total throughput time</td>
<td>5.70 ± 0.17 (-66 %)</td>
<td>101.5</td>
<td>YES</td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>4.66 ± 0.15 (-62 %)</td>
<td>73.6</td>
<td>YES</td>
</tr>
<tr>
<td>Team throughput time</td>
<td>1.61 ± 0.04 (-64 %)</td>
<td>104.3</td>
<td>YES</td>
</tr>
<tr>
<td>Percentage of time for new applications</td>
<td>2.91 ± 0.17 (-77 %)</td>
<td>36.1</td>
<td>YES</td>
</tr>
<tr>
<td>Workload</td>
<td>51.34 ± 2.0 (-63 %)</td>
<td>62.5</td>
<td>YES</td>
</tr>
<tr>
<td>Number of agenda documents</td>
<td>0 ± 0 (-100 %)</td>
<td>50.74</td>
<td>YES</td>
</tr>
<tr>
<td>(\alpha = 0.05, n_1 = 20, n_2 = 20, t_{\text{val}}(n_1 + n_2 - 2) = t_{\text{val}}(38) = 2.021)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 7.17c:** Results of treatment 7c, 100% printing at agent

212
7.2.6 Conclusions and recommendations

From the model results, we can draw several conclusions that lead to recommendations for Dutch Life. From personal experiences and observations, we can add several other conclusions. We first present the conclusions regarding the treatments for the 'internal' alternatives to be followed by the conclusions relating to the 'chain alternatives'. After that we provide a list of recommendations.

The internal alternatives: treatments 1 to 4

From the results of treatments 1 to 4 two issues immediately arise: the department of medical affairs and the problems relating to the agenda documents. The handling that is done by the department of medical affairs has a tremendous influence on throughput time and workload. For both cases, the origin of the problem lies outside Dutch Life. Medical affairs depends on response times of doctors who perform medical checks on clients and on the clients themselves. In the case of errors made by intermediaries, Dutch Life depends heavily on the response time and accuracy of the intermediary.

The problems relating to medical affairs could be dealt with by investigating the internal processes of this department. Furthermore, the threshold above which an acceptor has to involve medical affairs could be raised. This requires an investigation into the relation between insured value and the volume of applications. Currently, the share of applications that have to pass medical affairs lies at around 35 percent.

The errors made by intermediaries are a tougher problem. To improve this situation, Dutch Life has to coax the intermediaries to use clear forms, be accurate in filling in the forms and in signing them. Possibly, support for the intermediary, in terms of a computer program that helps to fill in the form correctly, might help. Also, electronic data interchange may contribute to improving this situation.
Designing Organizational Coordination

Another issue that draws attention is the fact that throughput times are in no relation to net production times. In fact if all handling times are added up, it takes in effect no more than a few hours to accept a policy. This means that an increase of the capacity of an acceptance team will not lead to a worthwhile decrease of throughput times. In other words: the workload can be handled, but the process has to rely too much on the performance of outside actors. To cope with this problem, the possibility might be investigated of more autonomous production teams with broad expertise within the team.

The chain alternatives: treatments 5 to 7

The chain alternatives are interesting for several reasons. All alternatives decrease cost and throughput time. If, however, we compare the treatments of set 5 and set 6, we conclude that electronic acceptance has no significant positive influence. If the acceptance results are electronically transmitted back to the intermediary, we see a strong decrease of average throughput time, cost and workload. The fact that electronic acceptance gives no better performance than electronic delivery on its own, is due to the high number of applications that have to pass medical affairs. We basically draw the following conclusions:

- Electronic acceptance is only effective if the number of applications that have to pass medical affairs is decreased.
- By changing the acceptance thresholds, performance can be improved.
- The software for electronic acceptance could be used to support the acceptors in their tasks.
- Electronic data interchange between intermediary and Dutch Life could improve performance in terms of throughput times.

Recommendations

Based on the results of the simulation model and on personal experiences and observations, we would recommend Dutch Life to take the following steps:
Applying the design system: three new cases

1. Initiate an investigation into the processes of medical affairs.
2. Initiate an investigation into possibilities to solve the agenda document problem.
3. Instead of waiting for one day, check the results of data entry directly after entering the data.
4. Determine the acceptance thresholds for both an electronic acceptance system and the acceptor. When the acceptor works with a higher threshold, the first filter could be the electronic system (taking all straightforward applications) and the second filter the acceptor (who takes the exceptional applications).
5. Investigate possibilities for electronic data interchange with the intermediary and include the possibility of transmitting back the acceptance data and letting the intermediary print the policy.

The design system

The Dutch Life case study has been a tough test for the design system. The case itself is very complex, and the design system was used not only by the author, but also by the other members of the design team. The conceptual models and the simulation environment did not result in any problems and were used as a basis for many discussions within the design team and with management. The members of the design team perceived the conceptual model types as natural ways to describe the organizational network, the business processes and the coordination within and between these processes.

The simulation model, and especially the animation, were powerful tools in showing the business processes and the way the design team thought about these processes. The animation was even used to inform the statutory workers council about the steps to be taken in the near future.

Another illustration of the success of the design system is the fact that Dutch Life wanted to use it again for a next project. This project is reported on in the next section, where we investigate the mutations as carried out by the acceptors.
7.3 Case 4: Dutch Life 'policy surrender'

For our fourth case study, we stay with the same organization as before. The focus is on a similar acceptance team in the business unit operating in the individual pension market. We regard the processes that have to be carried out if a person wants to surrender his policy. When a policy is surrendered, the insured receives the amount of money that has been accumulated during the time the premium was due. From this amount, some handling costs and interests are deducted.

A policy may be surrendered for several reasons. First of all, the client may decide to surrender because he is no longer able to pay the premium, or when he needs a large sum of money. However, many surrenders occur after a divorce. In some instances, the insurance company or the intermediary decides to advise a client to surrender his or her policy, because another type of insurance generates a higher return.

We chose for the policy surrender process because it is a process that needs a lot of coordination between three parties: the insurance company, the intermediary and the client. In the case of a divorce, the client even consists of two parties. Furthermore, to successfully complete a surrender process, Dutch Life has to coordinate with several parties internally, much like for the acceptance process. All this coordination results in very long throughput times. A second reason for choosing the surrender process was the fact that the surrender process is a mutation that occurs frequently, and costs a lot of money: 10,000 surrenders annually, resulting in 12 man-years of annual costs.

In the following subsections, we investigate the surrender processes with our design system and analyze the alternatives for change. Like in the previous case study, the project was carried out by a design team, consisting of 6 experts of Dutch Life and the author (including acceptance team leaders and insurance experts). The results of the project are discussed along the lines of conceptualization, specification, and solution finding.
7.3.1 Conceptualization

Before we identify the object classes that take part in the surrender processes, we first give a general impression of the process itself and the organizational context in which it is carried out.

The surrender process is carried out by the acceptance teams as described in section 7.2. For this process, they do not need to coordinate with the department of medical affairs, but sometimes they need actuarial advice for a calculation of the amount of money that has to be paid to the client. Another department that is involved is the payment department, which arranges payment to the client.

The process usually starts at the client, who fills in a form for the surrender of his policy (in about 20 percent of the cases the intermediary sends it). The Dutch Life mailroom sends the form to the right acceptance team, where it is handled by the pre-processor. The preprocessor retrieves the original policy from the archive, together with a so-called info-form, that contains the current status of the policy. When the file is complete, he sends it to the team leader who dispatches it to one of his acceptors. The acceptor first reads the contents of the file and determines whether he needs additional expertise from the legal department or another colleague. If so, he contacts these people for further advice. He then has to determine the method of calculation, for which he has two choices: to do it himself by means of the standard computer program or send a calculation request to the actuaries. Once the acceptor has obtained the results of the calculation, he can produce a formal request for surrender which he sends to the client for a signature. Usually complications occur once this form is returned. If the surrender is due to a divorce, the acceptor sends a so called ‘divorce letter’ to the client, asking for a confirmation of the divorce and the arrangements concerning the partition of joint property. If, in the case of a divorce, both parties are beneficiaries, the acceptor has to produce a legal form concerning the partition of the payment, which has to be signed by both parties. Once this form is returned, the acceptor can arrange payment and post-processing, including mutations in the policy system and informing the tax office about the transaction.
Designing Organizational Coordination

Identify object classes

The object classes involved in this process are displayed in tables 7.18 to 7.21. We distinguish between actor, repository, link and item classes. The actors are the same as in the previous case study, except for the computing center and the medical affairs department who do not take part in the surrender process. Furthermore, the legal department and the payment department are added to the actor list.

<table>
<thead>
<tr>
<th>Object</th>
<th>Attributes</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>Identifier</td>
<td>Send request for surrender</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sign forms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Send back forms</td>
</tr>
<tr>
<td>Intermediary</td>
<td>Identifier</td>
<td>Advise client</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Send request for surrender</td>
</tr>
<tr>
<td>Bank</td>
<td>Identifier</td>
<td>Receive instruction for payment</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Make payment to client</td>
</tr>
<tr>
<td>Mailroom</td>
<td>Identifier</td>
<td>Send request to preprocessor</td>
</tr>
<tr>
<td>Preprocessor</td>
<td>Identifier</td>
<td>Get policy from archive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Request policy status from systems</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Send file to team leader</td>
</tr>
<tr>
<td>Team leader</td>
<td>Identifier</td>
<td>Distribute files</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Acceptor tasks</td>
</tr>
<tr>
<td>Acceptor</td>
<td>Identifier</td>
<td>Acceptor tasks (see description)</td>
</tr>
<tr>
<td>Legal department</td>
<td>Identifier</td>
<td>Give juridical advice</td>
</tr>
<tr>
<td>Payment department</td>
<td>Identifier</td>
<td>Arrange payment</td>
</tr>
<tr>
<td>Actuaries department</td>
<td>Identifier</td>
<td>Calculate amount of payment</td>
</tr>
</tbody>
</table>

**Table 7.18: Actor classes**

The repositories are displayed in table 7.19.

<table>
<thead>
<tr>
<th>Object</th>
<th>Attributes</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy system</td>
<td>Identifier</td>
<td>Store policy data</td>
</tr>
<tr>
<td>Guidance system</td>
<td>Identifier</td>
<td>Store document data</td>
</tr>
<tr>
<td>Archive</td>
<td>Identifier</td>
<td>Store paper documents</td>
</tr>
</tbody>
</table>

**Table 7.19: Repository classes**

The links that are used in this process are no different from the previous case study. We display the links in table 7.20.
Applying the design system: three new cases

<table>
<thead>
<tr>
<th>Object</th>
<th>Attributes</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dutch mail</td>
<td>Identifier</td>
<td>Transport documents</td>
</tr>
<tr>
<td>Internal mail</td>
<td>Identifier</td>
<td>Transport documents</td>
</tr>
<tr>
<td>Computer network</td>
<td>Identifier</td>
<td>Transmit policy data</td>
</tr>
</tbody>
</table>

*Table 7.20: Link classes*

The surrender process involves several items that are sent between the different actors and repositories. First of all, there is the request itself, the policy to which it applies, the sheet containing the policy’s status information and the data for the guidance system.

<table>
<thead>
<tr>
<th>Object</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surrender request</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>Status</td>
</tr>
<tr>
<td>Policy documents</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>Policy data</td>
</tr>
<tr>
<td>Policy information sheet</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>Policy status</td>
</tr>
<tr>
<td>Guidance data</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>Client</td>
</tr>
<tr>
<td>Request for legal advice</td>
<td>Identifier</td>
</tr>
<tr>
<td>Legal advice</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>Advice</td>
</tr>
<tr>
<td>Request for calculation</td>
<td>Identifier</td>
</tr>
<tr>
<td>Calculation results</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>Payment amount</td>
</tr>
<tr>
<td>Formal request for surrender</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>Client data</td>
</tr>
<tr>
<td></td>
<td>Status</td>
</tr>
<tr>
<td>Divorce letter</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>Status</td>
</tr>
<tr>
<td></td>
<td>Client data</td>
</tr>
<tr>
<td>Legal partition form</td>
<td>Identifier</td>
</tr>
<tr>
<td></td>
<td>Client / policy data</td>
</tr>
<tr>
<td></td>
<td>Status</td>
</tr>
<tr>
<td>Payment</td>
<td>Client data</td>
</tr>
<tr>
<td></td>
<td>Amount</td>
</tr>
<tr>
<td>Letter for tax office</td>
<td>Surrender info</td>
</tr>
</tbody>
</table>

*Table 7.21: Item classes*
Designing Organizational Coordination

In addition, in the process we identify the formal request, the divorce letter, the legal partition form, and the requests for advice from the legal or the actuaries department. Finally, there is the payment and the note for the tax office.

Construct conceptual model

The objects as identified above form the basis of the network model for surrender processing. From the macro-perspective, the network model is slightly different from the one displayed in figure 7.2, because in this case, most of the interaction between Dutch Life and the client is direct. Furthermore, in this case the bank arranging payment plays a role. In figure 7.9 we show the network model from the macro-perspective.

![Network model from the macro-perspective](image)

**Figure 7.9: Network model from the macro-perspective**

From the meso-perspective, the network model resembles the model of the previous case. In figure 7.10 we see that the department of medical affairs has disappeared and that the department of payment and the department of legal affairs have been added.

In addition, the payment department sends payment requests to the bank which pays the amount of money to the client. We did not incorporate the tax office because this actor is not relevant for calculating throughput times.
Applying the design system: three new cases

Figure 7.10: Network model from the meso-perspective

A global representation of the surrender processes is given in the process model of figure 7.11.

Figure 7.11: General process model

Some of the activities can be detailed further. An example of this is given in figure 7.12 where we display the process model of surrender handling tasks that have to be performed in cases where a surrender is legally complex.
Designing Organizational Coordination

Figure 7.12: Actor model for surrender activities
Applying the design system: three new cases

The conceptual models were used extensively in discussion with the members of the design team and with Dutch Life's management. After two discussion sessions with Dutch Life management and three sessions with the design team, we concluded that a sufficient level of detail was achieved and that the models were representing the surrender processes as carried out by the teams of Dutch Life correctly. Based on the conceptual models and the experience of the modeling process we reformulated the problem and constructed a simulation model, assisted by Dutch Life's management, the design team and some members of the acceptance team involved.

Formulate problem statement

After conceptualization, and in agreement with Dutch Life's management, we formulated the problem as follows:

What measures should be taken to improve the performance of the surrender process in terms of coordination, effectiveness and efficiency? What effect do these improvements have on logistic indicators such as throughput times and handling times related to the surrender process?

In this specific case, the workload was of no interest. This is because the surrender process is one of the many mutation processes carried out by the same team and the total workload of an acceptor consists of the combined workload for all these different mutations.

7.3.2 Specification

Based on the conceptual models and the problem formulation, a simulation model was developed. Parts of the DOC/S template were used for the construction of the model. The specification was carried out by the team that was also involved in conceptualization and included a representation of Dutch Life's management and a representation of the acceptance team involved. We discuss specification in the same way as we did in the previous case studies.
Designing Organizational Coordination

Output variables

As mentioned before, Dutch Life keeps track of several performance indicators, so a numerical validation of the model would be facilitated by incorporating the same indicators as output variables. For this case, we are only interested in throughput times and handling times. A discussion with management and team members concerning the output variables led to the following variables to be recorded by the simulation model:

- Total throughput time.
- Internal throughput time.
- Team throughput time.
- Total handling time, being the sum of all handling times of Dutch Life personnel, related to the surrender process.

The internal throughput time and the team throughput time will be used for replicative validation. These are the only variables for which empirical values can be collected from Dutch Life records.

According to Dutch Life Management, an improvement of the throughput times is the main issue on which the generation of alternatives should focus.

Model reductions

In order to arrive at a simulation model that shows good correspondence to reality, and in order to avoid unnecessary complexity, the design team was engaged in a discussion concerning model reductions. In this discussion, we concluded that it was important that the simulation model should incorporate the whole chain, including the client. The reason for this is that, in this case, the client has an important impact on throughput time, especially where the policy surrender is due to a divorce. The design team identified two types of reductions:
Applying the design system: three new cases

Structural reductions, where certain entities from reality are regarded as black boxes:

- The mail room is represented as an accumulating queue that is emptied each morning at 9 o'clock.
- The clients that send requests for surrender to the team are reduced to a black box which is responsible for generating the requests and reacting on request of Dutch Life.
- The legal department and the calculations department are represented as storages that hold documents for a certain period.
- Documents that end up in the archive or at the client are removed from the system (after gathering statistics).

Numerical reductions, where handling times, routing times and decisions are reduced to stochastic values:

- Document generation is an exponential inter-arrival process.
- Handling times are reduced to triangular or continuous distributions, based on an assessment by team members and team leaders.
- The routing times of documents are represented by triangular distributions, the documents are batched in queues and released according to the times at which the internal mail employees make their rounds through the building.
- All decisions are stochastic values, assessed by members of the design team.

Input data

The values of the input variables were determined by acceptance team members and team leaders. In table 7.22 we present the input values.

Simulation model and animation

The network model served as a basis for the construction of the simulation model. The model consists of code blocks for each actor in the
Designing Organizational Coordination

network models. Exactly like in the previous case, the acceptors are implemented as one generic code block, operating for all acceptors.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of surrenders generated per day</td>
<td>tria(2.8,3.1,3.5)</td>
</tr>
<tr>
<td>Route time from client to Dutch Life</td>
<td>expo(12) hours (during the night)</td>
</tr>
<tr>
<td>Preprocessor</td>
<td></td>
</tr>
<tr>
<td>Time for archive and info retrieval</td>
<td>5 minutes + 1 day of waiting for the policy file from the archive</td>
</tr>
<tr>
<td>Time for input in guidance system</td>
<td>5 minutes</td>
</tr>
<tr>
<td>Time for routing documents to team leader</td>
<td>1 minute per daily batch</td>
</tr>
<tr>
<td>Team leader</td>
<td></td>
</tr>
<tr>
<td>Time for assigning a document to a team member</td>
<td>0.5 minute</td>
</tr>
<tr>
<td>Route time to team member (per daily batch)</td>
<td>1 minute</td>
</tr>
<tr>
<td>Team member</td>
<td></td>
</tr>
<tr>
<td>Time for intermediary check</td>
<td>tria(5,10,15) minutes</td>
</tr>
<tr>
<td>Probability of extra expertise needed</td>
<td>7 percent</td>
</tr>
<tr>
<td>Probability of involvement actuaries department</td>
<td>25 percent</td>
</tr>
<tr>
<td>Probability of making formal surrender form</td>
<td>99 percent</td>
</tr>
<tr>
<td>Time for making formal surrender form</td>
<td>tria(10,15,20) minutes</td>
</tr>
<tr>
<td>Time until client reacts on surrender form</td>
<td>tria(1,4,10) days</td>
</tr>
<tr>
<td>Probability of involving legal department</td>
<td>40 percent</td>
</tr>
<tr>
<td>Probability legal dept. encounters divorce problems</td>
<td>25 percent</td>
</tr>
<tr>
<td>Time for making divorce letter</td>
<td>5 minutes</td>
</tr>
<tr>
<td>Time until client reacts on divorce letter</td>
<td>tria(3,6,15) days</td>
</tr>
<tr>
<td>Probability of legal partition form in case of divorce</td>
<td>50 percent</td>
</tr>
<tr>
<td>Time for making legal partition form</td>
<td>tria(3,5,8) minutes</td>
</tr>
<tr>
<td>Time until client reacts on partition form</td>
<td>tria(3,20,50) days</td>
</tr>
<tr>
<td>Time for administrative post processing</td>
<td>tria(15,20,25) minutes</td>
</tr>
<tr>
<td>Payment department</td>
<td></td>
</tr>
<tr>
<td>Time for handling payment assignment</td>
<td>10 minutes</td>
</tr>
<tr>
<td>Legal affairs</td>
<td></td>
</tr>
<tr>
<td>Time for giving legal advice</td>
<td>tria(10,20,30) minutes</td>
</tr>
<tr>
<td>Actuaries</td>
<td></td>
</tr>
<tr>
<td>Time for calculation</td>
<td>tria(10,15,20) minutes</td>
</tr>
<tr>
<td>Waiting and routing time</td>
<td>2 days (1 back, 1 forth)</td>
</tr>
</tbody>
</table>

**Table 7.22: Input data for the surrender case**

The simulation model has three animation screens: a screen for the external parties, like the client and the bank which pays the money and two screens for Dutch Life’s handling, one of which is shown below.
Applying the design system: three new cases

![Screen dump of Dutch Life handling](image)

**Figure 7.13: Screen dump of Dutch Life handling**

In figure 7.13 we see the preprocessor and the team leader in the lower right section and the team members across the screen. The other screen for Dutch Life’s handling shows the legal department, the payment department and the actuaries department.

**Initial treatment**

For the initial treatment, we have to establish the start-up time, the run-length and the number of replications needed to achieve results that reach a certain level of confidence. Like in the previous case, the system is non-terminating. We estimate the start-up time from a moving average plot of the measured internal and total throughput times. Figure 7.14 shows the two moving average plots and we conclude that a start-up time of 50 days is sufficient. We determined a run length of 500 days due to occasionally very long client response times.
After an initial number of 5 replications, we determined the number of replications needed for a treatment by means of formula (1) and the procedure as sketched in section 6.2.2. Table 7.23 shows the results of this procedure for two output variables and a confidence level of 0.05. The delta values are chosen in such a way that the length of the 95% confidence interval around the mean does not exceed 20 percent of this mean value. We conclude that a total number of 5 replications is sufficient.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>Standard deviation</th>
<th>Delta</th>
<th>Minimum # replications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team throughput time</td>
<td>7.31 (days)</td>
<td>0.54</td>
<td>1.46</td>
<td>5</td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>15.54 (days)</td>
<td>1.02</td>
<td>3.11</td>
<td>4</td>
</tr>
</tbody>
</table>

*Table 7.23: Determining the number of replications*

### 7.3.3 Verification and validation

Verification and validation was carried out in the same way as in the previous case study. Below we discuss both shortly.
Applying the design system: three new cases

Verification

The individual parts of the model as given in the model description above were tested for their behavior by means of a numerical analysis of their functioning and a check of the program code. The animation was helpful in finding errors in the simulation code. The code was tested by manual walkthroughs and by letting individual items follow parts of the code and testing variable values afterwards. The design team did a joint walkthrough of the simulation model which resulted in several improvements of the model.

Validation

The model was validated by means of replicative validation and by expert validation. Simulation results and the animation were discussed by the design team and with members of the acceptance teams. The discussions led to the fine-tuning of several input values.

A replicative validation was carried out by performing a t-test on the difference between the means of the output variables for 5 replications and the means of empirical values found in Dutch Life records. We obtained a file in which the relative frequency distribution of internal and team throughput time was recorded for the period of a year. As the model generates approximately 1500 throughput times per replication, we produced 5 random samples of 1500 throughput times from the frequency distribution of the obtained file, using different random number streams. This was done by constructing a small simulation model that uses the Dutch Life file as input. After obtaining 5 times 1500 values for two variables, we performed a t-test on the difference between the means. Table 7.24 shows the results of the replicative validation where the first mean and variance refer to the samples from Dutch Life records and the second mean and variance refer to the 5 replications of the model. The level of confidence is 95 %, resulting in a t-value of $t_{0.025}(n_1 + n_2 - 2) = t_{0.025}(8) = 2.306$. The | t | values give the half-width of the 95 % confidence intervals of $d / s_d$. We conclude that there is no significant difference between the empirical values and the experimental values.
Designing Organizational Coordination

<table>
<thead>
<tr>
<th>Variable</th>
<th>$x_1$</th>
<th>$s^2$</th>
<th>$x_2$</th>
<th>$s^2$</th>
<th>$d$</th>
<th>$s$</th>
<th>$t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal throughput t.</td>
<td>17.06</td>
<td>0.29</td>
<td>15.54</td>
<td>1.04</td>
<td>1.52</td>
<td>0.82</td>
<td>1.85</td>
</tr>
<tr>
<td>Team throughput t.</td>
<td>7.44</td>
<td>0.028</td>
<td>7.31</td>
<td>0.291</td>
<td>0.13</td>
<td>0.32</td>
<td>0.41</td>
</tr>
</tbody>
</table>

$\alpha = 0.05, n_1 = 5, n_2 = 5, t_{\alpha}(n_1 + n_2 - 2) = 2.306$

**Table 7.24: Replicative validation results**

7.3.4 Solution finding and consistency check

Dutch Life management was confronted with the model results of the treatment of the current situation and the animation. Based on this, the design team and Dutch Life management discussed alternative strategies for change.

From the simulation model and the collected data concerning the current situation, it can be concluded that the client is an important factor influencing the throughput time. Alternatives should therefore be focused on increasing the speed and accuracy with which the client returns his or her forms. Another important issue for Dutch Life management is the total handling time of Dutch Life actors consumed by this process, because handling time (and not throughput time) is related to process costs. From these reflections, the design team concluded that two types of alternatives were needed to improve the process:

1. Better coordination between Dutch Life and clients.
2. Faster handling times within Dutch Life.

An example of the first type of alternative is the development of a “do it yourself package” for surrendering a policy that can be sent to the client by Dutch Life or the intermediary. In this way, the client himself can collect, fill out and sign all documents and send them to Dutch Life at once, without having to refer back to Dutch Life. With this solution, all preconditions for the successful and fast processing of the request for surrender are fulfilled. An example of the second alternative is a better and more straightforward support for the acceptor, including better calculation tools that reduce the number of calculation requests to the actuaries.
Applying the design system: three new cases

These ideas can be tested in the simulation model for their effectiveness. The design team therefore devised the following treatments:

<table>
<thead>
<tr>
<th>Treatment</th>
<th>Alternative</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Current situation</td>
</tr>
<tr>
<td>1a,1b,1c</td>
<td>Accurate and 'one time only' delivery by the client. (20, 50 and 99 % of the incoming surrenders)</td>
</tr>
<tr>
<td>2a,2b,2c</td>
<td>All handling times within Dutch Life are reduced. (Handling times reduced to 90, 80 and 70 % of original value)</td>
</tr>
<tr>
<td>3</td>
<td>All calculations are done by the acceptor himself</td>
</tr>
</tbody>
</table>

*Table 7.25: Treatments for the policy surrender case*

Consistency

The alternatives do not have any influence on the consistency between simulation model and conceptual models. All alternatives can be implemented in the simulation model by changing input values.

7.3.5 Results

Treatment 0

Treatment 0 concerns the current situation and serves as a yardstick for all alternatives. We tested the alternatives on significant change relative to treatment 0, by means of a t-test. In the tables, we note the output variables, their 95 % confidence intervals, the \(|t|\) values and the significance. For treatment 0 we note the variables and their confidence intervals. The simulation results of treatment 0 are displayed in table 7.26.

<table>
<thead>
<tr>
<th>Output variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total throughput time (days)</td>
<td>20.99 ± 0.88</td>
</tr>
<tr>
<td>Internal throughput time (days)</td>
<td>15.54 ± 1.17</td>
</tr>
<tr>
<td>Team throughput time (days)</td>
<td>7.31 ± 0.62</td>
</tr>
<tr>
<td>Handling time (minutes)</td>
<td>104.36 ± 1.46</td>
</tr>
</tbody>
</table>

\(\alpha = 0.05, n = 5\), values determine 95 % confidence intervals

*Table 7.26: Results of treatment 0, current situation*
Designing Organizational Coordination

Treatments 1a, 1b, and 1c

Treatment set 1 concerns the client as an important factor in throughput time. In the treatments 1a, 1b and 1c we assume that respectively 20%, 50% and 99% of the incoming surrenders are complete and accurate in terms of forms and signatures. So for these types of surrenders, no follow-up communication with the client is needed. The results are displayed in tables 7.27a, 7.27b and 7.27c.

<table>
<thead>
<tr>
<th>Output variable (20 %)</th>
<th>Value (rel. change)</th>
<th>t t</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total throughput time</td>
<td>19.13 ± 0.23</td>
<td>1.91</td>
<td>NO</td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>13.67 ± 0.72</td>
<td>1.41</td>
<td>NO</td>
</tr>
<tr>
<td>Team throughput time</td>
<td>6.58 ± 0.89</td>
<td>0.95</td>
<td>NO</td>
</tr>
<tr>
<td>Handling time</td>
<td>102.33 ± 1.05</td>
<td>1.03</td>
<td>NO</td>
</tr>
<tr>
<td>( \alpha = 0.05, n_1 = 5, n_2 = 5, \ t_{\alpha}( n_1 + n_2 - 2) = t_{0.05}(8) = 2.306 )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7.27a: Results of treatment 1a, 20% of forms accurate

<table>
<thead>
<tr>
<th>Output variable (50 %)</th>
<th>Value (rel. change)</th>
<th>t t</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total throughput time</td>
<td>16.73 ± 0.60 (-20 %)</td>
<td>4.27</td>
<td>YES</td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>11.21 ± 0.87 (-28 %)</td>
<td>3.24</td>
<td>YES</td>
</tr>
<tr>
<td>Team throughput time</td>
<td>6.24 ± 0.88</td>
<td>1.39</td>
<td>NO</td>
</tr>
<tr>
<td>Handling time</td>
<td>99.25 ± 1.04 (-5 %)</td>
<td>2.90</td>
<td>YES</td>
</tr>
<tr>
<td>( \alpha = 0.05, n_1 = 5, n_2 = 5, \ t_{\alpha}( n_1 + n_2 - 2) = t_{0.05}(8) = 2.306 )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7.27b: Results of treatment 1b, 50% of forms accurate

<table>
<thead>
<tr>
<th>Output variable (99%)</th>
<th>Value (rel. change)</th>
<th>t t</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total throughput time</td>
<td>12.08 ± 0.40 (-42 %)</td>
<td>9.09</td>
<td>YES</td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>6.59 ± 0.72 (-58 %)</td>
<td>6.78</td>
<td>YES</td>
</tr>
<tr>
<td>Team throughput time</td>
<td>5.51 ± 0.74 (-25 %)</td>
<td>2.43</td>
<td>YES</td>
</tr>
<tr>
<td>Handling time</td>
<td>94.06 ± 0.82 (-10 %)</td>
<td>6.11</td>
<td>YES</td>
</tr>
<tr>
<td>( \alpha = 0.05, n_1 = 5, n_2 = 5, \ t_{\alpha}( n_1 + n_2 - 2) = t_{0.05}(8) = 2.306 )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7.27c: Results of treatment 1c, 99% of forms accurate

From the results of treatment set 1, we see that the influence of the client on total and internal throughput times can only be reduced if a substantial number of clients is prepared to accurately complete and
Applying the design system: three new cases

sign the forms before sending them to Dutch Life. However, if such an alternative succeeds, substantial gains in throughput time can be made.

Treatments 2a, 2b, and 2c

Treatment set 2 concerns a reduction of handling times within Dutch Life. The effects of reductions by 10 %, 20 % and 30 % are calculated in treatments 2a, 2b and 2c respectively. The results are displayed in tables 7.28a, 7.28b and 7.28c.

<table>
<thead>
<tr>
<th>Output variable (90 %)</th>
<th>Value (rel. change)</th>
<th>t</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total throughput time</td>
<td>19.23 ± 0.71</td>
<td>1.74</td>
<td>NO</td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>13.80 ± 0.63</td>
<td>1.32</td>
<td>NO</td>
</tr>
<tr>
<td>Team throughput time</td>
<td>5.44 ± 0.73 (-26 %)</td>
<td>2.53</td>
<td>YES</td>
</tr>
<tr>
<td>Handling time</td>
<td>95.30 ± 1.64 (-10 %)</td>
<td>5.07</td>
<td>YES</td>
</tr>
</tbody>
</table>

\[ \alpha = 0.05, n_1 = 5, n_2 = 5, \ t_{\alpha} (n_1 + n_2 - 2) = t_{0.05} (8) = 2.306 \]

Table 7.28a: Results of treatment 2a, 10% time reduction

<table>
<thead>
<tr>
<th>Output variable (80 %)</th>
<th>Value (rel. change)</th>
<th>t</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total throughput time</td>
<td>18.52 ± 0.67 (-12 %)</td>
<td>2.46</td>
<td>YES</td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>13.09 ± 0.66</td>
<td>1.86</td>
<td>NO</td>
</tr>
<tr>
<td>Team throughput time</td>
<td>4.94 ± 0.89 (-32 %)</td>
<td>3.10</td>
<td>YES</td>
</tr>
<tr>
<td>Handling time</td>
<td>86.16 ± 1.58 (-17 %)</td>
<td>10.43</td>
<td>YES</td>
</tr>
</tbody>
</table>

\[ \alpha = 0.05, n_1 = 5, n_2 = 5, \ t_{\alpha} (n_1 + n_2 - 2) = t_{0.05} (8) = 2.306 \]

Table 7.28b: Results of treatment 2b, 20% time reduction

<table>
<thead>
<tr>
<th>Output variable (70 %)</th>
<th>Value (rel. change)</th>
<th>t</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total throughput time</td>
<td>18.41 ± 0.69 (-12 %)</td>
<td>2.56</td>
<td>YES</td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>13.05 ± 0.61</td>
<td>1.89</td>
<td>NO</td>
</tr>
<tr>
<td>Team throughput time</td>
<td>4.52 ± 0.83 (-38 %)</td>
<td>3.69</td>
<td>YES</td>
</tr>
<tr>
<td>Handling time</td>
<td>77.10 ± 0.85 (-26 %)</td>
<td>16.48</td>
<td>YES</td>
</tr>
</tbody>
</table>

\[ \alpha = 0.05, n_1 = 5, n_2 = 5, \ t_{\alpha} (n_1 + n_2 - 2) = t_{0.05} (8) = 2.306 \]

Table 7.28c: Results of treatment 2c, 30% time reduction

From the results of treatment set 2, we conclude that a reduction of handling times can decrease the team throughput time substantially.
Designing Organizational Coordination

The decrease of total throughput time in treatments b and c can be explained by the fact that a certain number of payments can be sent to the bank one day earlier as a result of faster handling by Dutch Life. However, this has no influence on internal throughput time.

Treatment 3

Treatment 3 concerns an alternative whereby all calculations that are normally done by the actuaries are now done by the acceptor. From the results of table 7.29, we conclude that this has no positive or negative effect on the output values.

<table>
<thead>
<tr>
<th>Output variable</th>
<th>Value (rel. change)</th>
<th>t</th>
<th>Sign.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total throughput time</td>
<td>19.73 ± 0.51</td>
<td>1.27</td>
<td>NO</td>
</tr>
<tr>
<td>Internal throughput time</td>
<td>14.32 ± 0.53</td>
<td>0.93</td>
<td>NO</td>
</tr>
<tr>
<td>Team throughput time</td>
<td>5.88 ± 0.81</td>
<td>1.91</td>
<td>NO</td>
</tr>
<tr>
<td>Handling time</td>
<td>100.56 ± 1.49</td>
<td>2.04</td>
<td>NO</td>
</tr>
</tbody>
</table>

$\alpha = 0.05, n_1 = 5, n_2 = 5, t_{\text{calc}}(n_1 + n_2 - 2) = t_{\text{calc}}(8) = 2.306$

Table 7.29: Results of treatment 3, calculation done by acceptor

7.3.6 Conclusions

From this case study, we can draw some conclusions that lead to recommendations for Dutch Life. We also evaluate the application of the design system to this case.

Recommendations for Dutch Life

From the conceptual models, we learn that the surrender process is a complex process with many interactions with the client. For shortening total throughput time, Dutch Life has to rely very much on its clients. The question is, however, whether this really is a problem. So long as the process is out of the hands of the acceptor, it does not cost Dutch Life anything. On the other hand, if the acceptor has to write to the client three times, he has to pick up the same file three times, resulting in extra handling time. So, if Dutch Life wants to reduce the
Applying the design system: three new cases

handling times and thereby the cost of the surrender process, the handling times for individual tasks should be reduced (treatment set 2) and the client should be stimulated to perform all paperwork accurately and at the time of deciding to surrender the policy (treatment set 1).

Reflecting on these issues, we recommend the following actions to Dutch Life:

1. Making it easy for the client: examine what events may occur with a client that is insured by Dutch Life and induce the actions that are related to these events. For example:

   • Marriage can lead to the addition of a name to the list of beneficiaries.
   • Divorce usually leads to policy surrender.
   • Buying a house usually leads to a different type of policy.
   • Passing certain age limits can lead to policy changes.

   After this investigation, a set of forms and a to-do list can be developed for each event. If a client informs Dutch Life or the intermediary, the forms and the action list can be handed to the client for fast processing. The intermediary may advise the client in filling out the forms. The forms could even be presented on a floppy disk containing a program that helps the client producing the right documents. Internally, Dutch Life could develop a list for each possible mutation of the conditions under which the process can be successfully completed.

2. The development of better support for the acceptor. To reduce handling times, support for the acceptor in terms of information technology, should be developed along the lines of the events that may occur with a client, and in coordination with the forms and to-do lists that are sent to the client. In this way, the client needs to arrange the paperwork only once and the acceptor is supported in a way that matches client handling.
Designing Organizational Coordination

The design system

In this case we have seen a small case study concerning a complex process. The process contains many coordination issues between Dutch Life and the client. The design team felt that the process and its coordination could be modeled accurately by means of both the conceptual models and the simulation tools. By analyzing the simulation model and its outcomes, new ideas for alternatives, like the event- and to-do lists emerged. Another conclusion that can be drawn from this particular case is that the design system does not stand in the way of a fast completion of the project. The case study was carried out within two months from start to end, during which the design team was involved for only twenty percent of the time.

The success of the design system was also recognized by another business unit of Dutch International Group: the general insurance company called Dutch General. The next section shows the application of the design system in a totally different setting, but still in an insurance environment: the logistical handling of incoming mail at Dutch General.
7.4 Case 5: Dutch General Mail Logistics

Dutch General is the general insurance company of the Dutch International group. It is the largest general insurance company of the Netherlands working with independent intermediaries. In the period of 1963 until 1985, the annual premium turnover of the company grew from 595 million guilders to 9.5 billion guilders. Important selling points of the company are high quality service and a good relationship with the intermediaries. Furthermore the company frequently acquired smaller enterprises in the insurance business.

The market for general insurance policies in the Netherlands has been under pressure from direct writing insurance companies, who could offer policies against a lower price by excluding the intermediaries. They took advantage of modern technologies like call centers and of the latest direct marketing strategies. Furthermore, as agreements between the large firms were canceled, the environment became more competitive. During the last ten years, competition came from unexpected corners: banks and pension funds started operating in the insurance market. However, after 1985, market shares stabilized and, once more, the general insurance business is profitable. In 1995, Dutch General is a healthy partner of Dutch International, in which it plays a major role.

Like many other insurance companies in the Netherlands, Dutch General’s organization structure was based on a differentiation of the type of product. Dutch General was organized in sectors, each responsible for a specific type of insurance (e.g. fire insurance, car insurance). In the beginning of the nineties, Dutch General management realized that the organizational structure would curtail flexible responses to market developments. Furthermore, a number of internal SWOT analyses showed that major improvements could be achieved in several parts of the organization. In 1994 these developments led to a major change in organizational structure, business processes and communication channels. The organization turned from product orientation to market orientation. The change took place together with a change in office location.
Designing Organizational Coordination

In the new organizational structure, each department has clearly defined commercial responsibilities. This means that each business unit now serves the intermediaries of a specific region of the Netherlands with the director of this unit being responsible for both the business processes and the results of his unit. Within the bounds set by general policy, the units are free to develop and implement their own thoughts about business processes, unit management and commercial activities. The only functional difference that still remains is the distinction between a private insurance division and a business insurance division. The private division consists of 8 regional business units, headed by two district directors. The business division consists of 11 regional units, headed by four district directors. Each unit is itself headed by a unit director. Furthermore, the organization has several staff departments that serve all regional units. The mail room and internal logistics department, which is the object of this case study, serves all units of Dutch General together with the units of Dutch Health, the health insurance company of Dutch International. The organizational structure of Dutch General is shown in figure 7.15.

![Diagram of organizational structure of Dutch General]

Figure 7.15: Organizational structure of Dutch General
Applying the design system: three new cases

Each regional unit consists of teams, responsible for a part of the business process. Within the business units, a distinction is made between loss claims handling and acceptance tasks (like in the Insuraco case). Usually, a private insurance business unit has three teams of about 10 employees, one team for loss claims handling and two teams for acceptance and mutation tasks. Each team consists of line employees (acceptors and claims handlers) and staff employees (preprocessors and post-processors).

The internal logistics department

Each day, a great number of documents is sent to the Dutch International offices where Dutch General is located. Documents may be sent to Dutch International in several ways: by mail or courier, by fax or electronically, by means of the special insurance data network.

Of all inhabitants of the office location, Dutch General receives by far the most documents, about 7000 per day on average. Most intermediaries send a number of documents with different destinations within Dutch General in one envelope. This grouping of documents is the key to the current arrangements at the mail room: many envelopes have to be opened to examine and sort the documents they contain. Mail for Dutch General is first checked by employees at the mailroom to assess the regional unit it belongs to. After routing to the regional unit, it is re-examined by the preprocessor of the unit to assess the team to which it belongs. After the change in organizational structure, Dutch General management wanted to investigate possible improvements in the logistic chain between delivery by the Dutch Mail and the teams in the regional units.

Problem description

The goal of the project was to analyze the logistic processes and assess the bottlenecks within these processes, to finally arrive at possible improvements of these processes. The reason for starting this project was a lack of insight into throughput times, volumes, failure rates and a number of other logistical indicators by Dutch General management.
Designing Organizational Coordination

Dutch General management specifically wanted insight into the following issues:

- **Duplicate processes**: to what extent are activities carried out twice and how can such a situation be prevented?
- **Process failures**: how many documents go to the wrong destination and start ‘wandering’ through the building?
- **Backlog**: what is the backlog at the mailroom and how can this be decreased?
- **Other performance indicators**: how are we doing in general, and how can this be improved?

This problem situation seems an excellent environment for testing the design system: a complex business process, many actors are involved and the coordination between these actors is key to the success of the business process. Up until this project, the design system was only tested by the author. To test the use of the design system more independently, this project was carried out by a graduate student who was trained in the use of the design system. The graduate student took part in a project team of Dutch General which was advised by the author.

7.4.1 Conceptualization

In this section we develop the conceptual model of Dutch General mail logistics. We identifying the object classes involved, construct conceptual models of the current situation and give a problem statement.

**Identify object classes**

Of all departments at the office location, Dutch General receives by far the most documents. We assume that all mail originates at an intermediary and is delivered by the Dutch Mail. Due to differences in type and volume of mail per region of the Netherlands, we identify 11 instances of the intermediary object class. This results in the following two object classes:
Applying the design system: three new cases

object class INTERMEDIARY
inherits from ACTOR

attributes
region

actions
create documents
send documents

object class DUTCH MAIL
inherits from ACTOR

attributes

actions
collect mail
Dutch Mail processes
deliver mail at Dutch International

The daily volume of mail arriving at Dutch International consists of envelopes or packages. Each of these so-called mail packets has one of the following destinations written on it:

• General mailbox of Dutch General.
• One of the 11 regional mailboxes of Dutch General.
• General mailbox of Dutch Health.
• The Dutch General free post mailbox.
• Registered mail.
• Other destinations.

The 11 regional mailboxes are shared by the respective private and business insurance units. The regional mailboxes were implemented for faster handling (sorting only between business and private).

A mail packet contains one or more documents. Each document has a destination within the Dutch International office location. The average number of documents per mail packet depends on the size of the packet. The mail room distinguishes between 4 different sizes. The documents in a mail packet usually have various destinations. Sometimes, a mail packet can be sent directly to one destination, without opening it. Such a document is called directly addressable.

In the mail room, all documents that are not directly addressable are opened. The contents of a mail packet are then judged by mail room personnel for their destination. Each time a document arrives at a new destination, it receives a date stamp and a destination code. This results in the following object classes:

241
Designing Organizational Coordination

object class MAIL PACKET
inherits from ITEM

attributes
address
size
number of documents
directly addressable

actions

object class DOCUMENT
inherits from ITEM

attributes
destination meant by intermediary
destination assigned
number of date stamps
document contents

actions

The Dutch Mail delivers the mail in mailbags to the mail room. Each mailbox results in one or more mailbags to be handled by the mail room. The main task of the mail room is the assessment of internal destinations of documents from mail packets that are not directly addressable. All mail packets that are directly addressable are being sent to their destination immediately. Employees of the mail room can take on three different roles: sorter, opener or assessor. A sorter manually sorts closed mail packets on size and destination (directly addressable packets). The opener opens up the packets that are not directly addressable with an opening machine. This machine can only handle one of four sizes at a time, therefore the packets must be sorted on size. An assessor determines the destination of the documents contained in a packet by means of several inquiry systems he can use. For example, if a document concerns a loss claim, he can type in the policy registration number and the system returns the regional unit where it belongs. The employees only work with one mailbag at a time to prevent mixing-up of already sorted material. This leads to the following object classes:

object class MAIL ROOM
inherits from ACTOR

attributes

actions
sort packets
open packets
assess destinations
send documents

object class ASSESSOR
inherits from ACTOR

attributes
name
current mailbag

actions
observe document
use inquiry system
assess destination
put document on related bin
Applying the design system: three new cases

<table>
<thead>
<tr>
<th>object class OPENER</th>
<th>object class SORTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>inherits from ACTOR</td>
<td>inherits from ACTOR</td>
</tr>
<tr>
<td>attributes</td>
<td>attributes</td>
</tr>
<tr>
<td>name</td>
<td>name</td>
</tr>
<tr>
<td>current mailbag</td>
<td>current mailbag</td>
</tr>
<tr>
<td>actions</td>
<td>actions</td>
</tr>
<tr>
<td>operate opening machine</td>
<td>sort mail packets</td>
</tr>
</tbody>
</table>

The regional units have their own preprocessors, who account for administrative tasks. The main tasks of the pre-processors are sorting the mail for the teams and preparing the output from the computing center for further processing and sending it to the intermediaries. The preprocessors are responsible for collecting the mail for their regional unit at the mail room several times a day. Mail that is incorrectly dispatched to a unit is sent to a new location by the internal mail service.

Within the teams, the documents are handled by team members. If they encounter a document that is addressed incorrectly, they also send it away with the internal mail service. All mail that is not dispatched to one of the regional units, is directed to one of the other actors inhabiting the office location. Dutch Health is an example. This leads us to the final set of object classes:

<table>
<thead>
<tr>
<th>object class PREPROCESSOR</th>
<th>object class TEAM MEMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>inherits from ACTOR</td>
<td>inherits from ACTOR</td>
</tr>
<tr>
<td>attributes</td>
<td>attributes</td>
</tr>
<tr>
<td>name</td>
<td>name</td>
</tr>
<tr>
<td>regional unit</td>
<td>regional unit</td>
</tr>
<tr>
<td>actions</td>
<td>actions</td>
</tr>
<tr>
<td>collect mail</td>
<td>receive documents</td>
</tr>
<tr>
<td>process documents</td>
<td>handle documents</td>
</tr>
<tr>
<td>send documents</td>
<td>send documents</td>
</tr>
<tr>
<td>receive document</td>
<td></td>
</tr>
</tbody>
</table>
Designing Organizational Coordination

<table>
<thead>
<tr>
<th>object class INTERNAL MAIL</th>
<th>object class OTHER INHABITANT</th>
</tr>
</thead>
<tbody>
<tr>
<td>inherits from ACTOR</td>
<td>inherits from ACTOR</td>
</tr>
</tbody>
</table>

attributes
name

actions
collect mail
deliver mail

actions
receive documents
send documents

Of course the employees we identified as actors make use of many repositories. In fact, each mailbag and sorting bin is a repository. We distinguish between 5 different types of repository:

- Mailbags.
- Outgoing mail bins (bins ready for collection by a preprocessor).
- Sorted but unopened mail.
- Opened mail.
- Returned mail.
- Inquiry systems.

Table 7.30 shows the repositories for mail logistics

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mailbag</td>
<td>General</td>
<td>Mailbag from the general mailbox</td>
</tr>
<tr>
<td></td>
<td>Regional</td>
<td>Mailbags from the regional mailboxes</td>
</tr>
<tr>
<td></td>
<td>Other inhabitants</td>
<td>Mailbag with mail for other inhabitants</td>
</tr>
<tr>
<td></td>
<td>Free mail</td>
<td>Mailbag with business reply documents</td>
</tr>
<tr>
<td>Outgoing mail</td>
<td>Regional units</td>
<td>Assessed documents for Dutch General</td>
</tr>
<tr>
<td></td>
<td>Other inhabitants</td>
<td>Documents for other inhabitants</td>
</tr>
<tr>
<td></td>
<td>Addressable</td>
<td>Bin with directly addressable mail</td>
</tr>
<tr>
<td>Sorted mail</td>
<td>General</td>
<td>Sorted mail from the general mailbags</td>
</tr>
<tr>
<td></td>
<td>Regional</td>
<td>Sorted mail from the regional mailbags</td>
</tr>
<tr>
<td>Opened mail</td>
<td>General</td>
<td>Opened mail from the general mailbags</td>
</tr>
<tr>
<td></td>
<td>Regional</td>
<td>Opened mail from the regional mailbags</td>
</tr>
<tr>
<td>Returned mail</td>
<td></td>
<td>Documents returned to the mail room</td>
</tr>
<tr>
<td>Inquiry system</td>
<td></td>
<td>Systems for assessing destinations</td>
</tr>
</tbody>
</table>

Table 7.30: Repositories for mail handling
Applying the design system: three new cases

The discussion above and the object classes we derive from it, form the basis for the construction of conceptual models of the current situation.

Construct conceptual model

The conceptual model consists of a number of network and process models. We start with the process model from the macro-perspective and zoom in to more detailed models.

Mail is generated at the intermediary, who sends it to Dutch International with the Dutch Mail. At the mail room, the mail is sorted and dispatched to one of the regional units or one of the other departments at the office location. From the preprocessors the mail will either be sent to the team members or to the internal mail service responsible for further dispatching. The other inhabitants and the team members may also send mail by the internal mail service. This can be visualized by the network model of figure 7.16.

![Network model from the macro-perspective](image)

**Figure 7.16: Network model from the macro-perspective**
Figure 7.17: Network model from the meso-perspective (mail room)
Applying the design system: three new cases

In the following models, we zoom in on the mailroom, the assessor and the preprocessor. At the mail room, incoming mail spread over several mailbags is delivered by the Dutch Mail. Each mailbag is handled by one or more sorters who split mail into directly addressable packets and the four sizes of packets that have to be opened by the opener. After opening, the mailbags for the regional units and the mailbag of the general mailbox are left for assessment by the assessors. After being assessed, a document ends up in the outgoing mailbox that is related to the destination in the office. The preprocessors regularly collect these boxes. The network model of the mail room is described in figure 7.17. From the micro-perspective, the network model of the assessor and of the preprocessor are the same (see figure 7.18).

![Network model from the micro-perspective](image)

**Figure 7.18: Network model from the micro-perspective**

We distinguish two process models, one of the assessment process in the mail room and one of the handling in the regional units. In the mail room, we identify two types of assessors. One handles documents that have arrived in one of the regional mailboxes, and one handles mail that has arrived in the general mailbox.

Except for the fact that the assessor for regional mailbox documents has only three boxes to send mail to (the private unit of the region, the business unit of the region, other inhabitants) the processes are the same. The other assessor has 20 boxes to dispatch mail to (the 11 business regional units, the 8 private regional units and other inhabitants). The process model of the assessment process is given in figure 7.19.
Handling at the regional units involves the preprocessors, the team members and the internal mail service. The process model of the process of the regional units is shown in figure 7.20.
Applying the design system: three new cases

Figure 7.20: Process model of the regional units

This concludes the construction of the conceptual models. The models form the basis for the development of a simulation model of mail handling at Dutch General. Before we discuss specification issues, we formulate the problem in terms of the structure and processes we have developed in our conceptual model.
Designing Organizational Coordination

Formulate problem statement

In discussion with Dutch General management, the project team, employees of the mail room and of the regional units, we formulated the problem as follows:

How can coordination between the mail room and the regional units be improved in terms of mail logistics and process quality? How do these improvements affect organizational performance indicators like backlog levels, throughput times, failure rates and process costs.

7.4.2 Specification

Based on the conceptual models and the problem formulation, a simulation model of mail logistics was constructed using the DOC/S template. Specification was carried out by the project team and fed back to the employees of different departments on a regular basis.

The model incorporates logistic handling from the intermediaries in the geographical regions until arrival of the documents at the teams. The primary processes of the teams are not part of the model. We discuss specification in the same way as the previous case studies.

Output variables

To analyze the performance of the logistic processes, we specify four types of output variables:

1. Throughput time of documents. Average throughput time is determined for documents that:
   - Arrive at the right destination the same day they are delivered by the Dutch Mail (weekly average and average for each weekday).
   - Are being sent to another destination by a preprocessor or team member (weekly average and average for each weekday).
   - Arrive at the right destination directly after mail processing.
   - Are in the model (all documents).
Applying the design system: three new cases

2. **Failure rate** (documents that arrive at the wrong destination). Of mail that is apparently misdirected, the following indicators are being collected:
   - Percentage of total volume of mail.
   - Number of documents that have 0, 1, 2, 3, 4, or more date stamps.

3. **Backlog** at the mail room

4. **Process efficiency** (handling time divided by throughput time\(^1\)). Efficiency is determined for documents that:
   - Are sent through to other departments.
   - Arrive at the right location directly after mail processing.
   - Are in the model (all documents).

**Model reductions**

To arrive at a model that shows good correspondence and still ‘fits the purpose’ of developing and analyzing alternatives, the design team was engaged in a discussion about model reductions. The following reductions were determined and carried out:

**Structural reductions:**

- All preprocessors arrive at 08:30 and leave at 16:30. They never make overtime.
- The moment at which the preprocessors collect documents at the mailroom does not depend on the their workload, but occurs at fixed times.
- The internal mail service does not make any ‘rounds’ but all mail that is sent through the internal service starts traveling to its destination at the same, fixed, times.
- Actions such as people walking between different departments to collect mail are not modeled. The related resource is set to inactive and the mail is routed to its destination. When the mail arrives, the resource is again set to active.

\(^1\) See Hammer and Champy (1993).
Designing Organizational Coordination

Numerical reductions:

- Document batches are generated from normal distributions. The parameters for this distribution are determined for each region. The document generators for each region are made dependent by using the same random number stream for each region. In practice, the number of documents arriving from the regions show the same pattern.
- Handling times are modeled as normal, log-normal and triangular distributions.
- The model contains no weekends. The effect of weekend in the throughput times is adjusted by calculating the number of weekends a document has passed when it leaves the system and by adding the time related to this number to the throughput time.

Input data

We identify three types of input data:

- Generators.
- Handling times.
- Probabilities.

The model contains two generators, one for determining the size of a mail packet and the number of documents it contains and one for the number of documents generated per day. Table 7.31 gives the values of the data for the size generator.

<table>
<thead>
<tr>
<th>Share of total volume</th>
<th>Size 1</th>
<th>Size 2</th>
<th>Size 3</th>
<th>Size 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 %</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26 %</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52 %</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># documents / packet</td>
<td>1+Pois(0.21)</td>
<td>1+Pois(0.33)</td>
<td>1+Pois(0.85)</td>
<td>1+Pois(8.54)</td>
</tr>
</tbody>
</table>

Table 7.31: Packet size generator data

Table 7.32 gives the data for the generator of the number of documents for each region and for the general mailbox and registered mail box. It also presents the probability of a certain destination for each region. The North region is a special case, 44% is for the private regional
Applying the design system: three new cases

unit, 23% is for the business regional unit, 4% is for other departments and 23% has the destination of the west 1 business regional unit. Furthermore, 30% of the total number of documents in the general mailbox and 85% of the documents in the registered mailbox is directly addressable.

<table>
<thead>
<tr>
<th>Region</th>
<th># documents</th>
<th>Std.dev.</th>
<th>Private</th>
<th>Business</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>North west 1</td>
<td>139</td>
<td>12</td>
<td>55%</td>
<td>34%</td>
<td>11%</td>
</tr>
<tr>
<td>North west 2</td>
<td>150</td>
<td>13</td>
<td>47%</td>
<td>43%</td>
<td>10%</td>
</tr>
<tr>
<td>North</td>
<td>205</td>
<td>29</td>
<td>44%</td>
<td>23%</td>
<td>4%</td>
</tr>
<tr>
<td>East</td>
<td>203</td>
<td>29</td>
<td>59%</td>
<td>33%</td>
<td>8%</td>
</tr>
<tr>
<td>Central</td>
<td>204</td>
<td>21</td>
<td>64%</td>
<td>28%</td>
<td>8%</td>
</tr>
<tr>
<td>West 1</td>
<td>118</td>
<td>12</td>
<td>52%</td>
<td>36%</td>
<td>12%</td>
</tr>
<tr>
<td>West 2</td>
<td>148</td>
<td>14</td>
<td>60%</td>
<td>32%</td>
<td>8%</td>
</tr>
<tr>
<td>Rhine area</td>
<td>220</td>
<td>19</td>
<td>59%</td>
<td>33%</td>
<td>8%</td>
</tr>
<tr>
<td>South-east 1</td>
<td>155</td>
<td>16</td>
<td>59%</td>
<td>28%</td>
<td>13%</td>
</tr>
<tr>
<td>South-east 2</td>
<td>139</td>
<td>12</td>
<td>57%</td>
<td>33%</td>
<td>10%</td>
</tr>
<tr>
<td>South-west</td>
<td>183</td>
<td>21</td>
<td>54%</td>
<td>34%</td>
<td>12%</td>
</tr>
<tr>
<td>General mailbox</td>
<td>4730</td>
<td>475</td>
<td>85%</td>
<td>15%</td>
<td></td>
</tr>
<tr>
<td>Registered mail</td>
<td>210</td>
<td>10</td>
<td>85%</td>
<td>15%</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.32: Regional generator data

A number of probabilities relate to what happens after a document has been assessed and given a destination at the mailroom. We distinguish between two situations: 1) the mail room or the preprocessor who sends through the document has assigned the right destination to it and 2) the mail room or the preprocessor that sends through a document has assigned the wrong destination to a document.

For the first situation, we specify two different actions to be performed by the preprocessor who receives the correctly addressed document: 1a) he erroneously sends it through to another regional unit or 1b) he sends it to the team (correctly). An analysis of Dutch General records results in the probabilities in table 7.33 (we went as far as analyzing the fourth regional unit that handles the document).
Designing Organizational Coordination

<table>
<thead>
<tr>
<th></th>
<th>Send through (1a)</th>
<th>To team (1b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>First region</td>
<td>2.0 %</td>
<td>98.0 %</td>
</tr>
<tr>
<td>Second region</td>
<td>1.5 %</td>
<td>98.5 %</td>
</tr>
<tr>
<td>Third region</td>
<td>1.0 %</td>
<td>99.0 %</td>
</tr>
<tr>
<td>Fourth region</td>
<td>0.0 %</td>
<td>100.0 %</td>
</tr>
</tbody>
</table>

*Table 7.33: Failure rates*

In the second situation, we distinguish between four options for the preprocessor who receives the wrongly addressed document: 2a) he sends it to the correct destination, 2b) he sends it to a wrong destination, 2c) he sends it to a team in his region (incorrectly) and 2d) he returns it to the mail room. Table 7.34 gives the related probabilities.

<table>
<thead>
<tr>
<th></th>
<th>Send through correct (2a)</th>
<th>Send through incorrect (2b)</th>
<th>To team (2c)</th>
<th>Return to mail room (2d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>First region</td>
<td>52 %</td>
<td>31 %</td>
<td>10 %</td>
<td>7 %</td>
</tr>
<tr>
<td>Second region</td>
<td>40 %</td>
<td>56 %</td>
<td>2 %</td>
<td>2 %</td>
</tr>
<tr>
<td>Third region</td>
<td>56 %</td>
<td>40 %</td>
<td>1 %</td>
<td>3 %</td>
</tr>
<tr>
<td>Fourth region and more</td>
<td>95 %</td>
<td>1 %</td>
<td>0 %</td>
<td>4 %</td>
</tr>
</tbody>
</table>

*Table 7.34: Failure rates (2)*

The handling times of the different actors are displayed in table 7.35.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Handling time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sorting time regional documents</td>
<td>Normal(3.5,0.3) seconds</td>
</tr>
<tr>
<td>Sorting time general documents</td>
<td>Normal(4.5,0.2) seconds</td>
</tr>
<tr>
<td>Sorting time other documents</td>
<td>Normal(3.5,0.3) seconds</td>
</tr>
<tr>
<td>Sorting time preprocessor</td>
<td>Normal(5.0,1.0) seconds</td>
</tr>
<tr>
<td>Opening time regional documents (batch)</td>
<td>Uniform(7,10) minutes</td>
</tr>
<tr>
<td>Opening time general documents (batch)</td>
<td>Uniform(25,30) minutes</td>
</tr>
<tr>
<td>Opening time other documents (batch)</td>
<td>Uniform(7,10) minutes</td>
</tr>
<tr>
<td>Assessment time registered mail</td>
<td>120 minutes (all documents)</td>
</tr>
<tr>
<td>Assessment time regional mail</td>
<td>7 + Lognormal(13,16) seconds</td>
</tr>
<tr>
<td>Assessment time general mail</td>
<td>8 + Lognormal(15,16) seconds</td>
</tr>
<tr>
<td>Assessment time returned mail</td>
<td>9 + Lognormal(18,16) seconds</td>
</tr>
<tr>
<td>Assessment time preprocessor</td>
<td>10 + Lognormal(40,10) seconds</td>
</tr>
</tbody>
</table>

*Table 7.35: Handling times*
Applying the design system: three new cases

Simulation model and animation

The code of the simulation model is grouped into the following parts:

- Generators.
- Dutch Mail.
- Mail room.
- Regional units.
- Internal mail service.
- Other departments.

The generators generate the input of the model: 11 generators for the regions, 1 for the general mailbox, one for the free mail box and one for the other inhabitants. Dutch Mail delivers the mailbags at fixed times, three times a day. The mail room consists of a number of actors that can take on the different roles as discussed in section 7.4.1. After assessment, the documents end up in a regional queue that is emptied by the preprocessor. The send and receive schedule of the internal mail service is implemented by several wait-signal constructs. The preprocessors are implemented identically for each region. The model contains 5 animation screens:

1. An overview of the Netherlands, showing the flow of mail from the regions to the office location. In a sense, it gives an impression of the macro-perspective.
2. Dutch Mail, showing the delivery of mail at the Dutch International office location.
3. Mail logistics at Dutch International, showing the mail room, the regional units, the other units and the flow of mail between these departments. It gives an impression from the meso-perspective.
4. The mail room, showing the different activities that occur in the mail room. This is also an impression from the meso-perspective.
5. A preprocessor. Showing the activities of a preprocessor, from the micro-perspective.

Figures 7.21 and 7.22 show screen-dumps of screens 1 and 3.
Designing Organizational Coordination

**Figure 7.21:** Screen-dump of the macro-perspective

**Figure 7.22:** Screen-dump of the meso-perspective
Applying the design system: three new cases

Initial treatment

For the initial treatment, we have to establish the start-up time, the run-length and the number of replications needed to achieve results that reach a certain level of confidence. Like in the previous case, the system is non-terminating. We estimate the start-up time from a moving average plot of total throughput time. Figure 7.23 shows a moving average plot and we conclude that a start-up time of 10 days is sufficient. With respect to cycle times, we determined the run length at 40 days.

![Diagram showing throughput time over simulation time](image)

**Figure 7.23: Moving average throughput times**

After an initial number of 5 replications, we determined the number of replications needed for a treatment by means of formula (1) and the procedure as sketched in section 6.2.2. Table 7.36 shows the results of this procedure for three output variables and a confidence level of 95%. The delta values are chosen small to get accurate mean values: 5 percent of the average value of the output variable. From table 7.36, we conclude that a total number of 10 replications is sufficient and we ran five extra replications.
Designing Organizational Coordination

<table>
<thead>
<tr>
<th>Throughput time (results after 5 replications)</th>
<th>Mean</th>
<th>Standard deviation</th>
<th>Delta</th>
<th>Minimum # replications</th>
</tr>
</thead>
<tbody>
<tr>
<td>All documents</td>
<td>14:57 hours</td>
<td>2:56 hr.</td>
<td>45 min.</td>
<td>6</td>
</tr>
<tr>
<td>Documents sent through</td>
<td>77:24 hours</td>
<td>5:59 hr.</td>
<td>3:52 hr.</td>
<td>2</td>
</tr>
<tr>
<td>Documents correctly addressed</td>
<td>11.48 hours</td>
<td>2:55 hr.</td>
<td>36 min.</td>
<td>9</td>
</tr>
</tbody>
</table>

*Table 7.36: Determining the number of replications*

7.4.3 Verification and validation

Verification and validation was carried out in the same way as in the previous case study. Below we discuss both shortly.

Verification

The individual parts of the model as given in the model description above, were tested for their behavior by means of a numerical analysis of their functioning and a check of the program code. The animation was helpful in finding errors in the simulation code.

The normal distributions for the generators were determined by applying a best fit algorithm provided by the ARENA input processor to the empirical values found.

The code was tested by manual walkthroughs and debugged by standard step- and trace options of the ARENA run processor. The model was also tested by letting individual items follow parts of the code and testing variable values afterwards. For further information we refer to Bouwman (1995).

Validation

The model was validated by means of replicative validation and by expert validation. Simulation results and the animation were discussed in the project team and with employees of the mail room and the regional teams. The simulation was judged as accurate and correctly representing the current situation.
Applying the design system: three new cases

Replicative validation was quite difficult as there were no empirical data available. To collect data about the current situation, the ‘age’ of documents was measured for two weeks at the preprocessors of each regional unit. The sample that resulted from this investigation was compared to the output of the simulation model by means of a chi-square test. The value of chi-square is determined by formula (6):

\[
\chi^2 = \frac{\sum_k (f_m - f_e)^2}{f_e}
\]

(6)

Where \(k\) represents the number of intervals, \(f_m\) is the model frequency for each of the intervals, \(f_e\) is the empirical frequency for each interval and \(\Sigma_k\) is the sum over all \(k\) intervals. If \(\chi^2 = 0\), there is no difference between model values and empirical values. To test a significant difference between model values and empirical values, \(\chi^2\) can be compared to the critical values as listed in the \(\chi^2\) table for the related degrees of freedom and desired level of confidence. The number of degrees of freedom is equal to \(k-1\). When the value that is found for \(\chi^2\) is smaller than the table value, the difference between model results and reality is not significant. For a discussion on the chi-square test, see Pegden et al. (1995). The chi-square test was carried out on frequency distributions of document age for each day of the week. The desired level of confidence is 0.05 (\(\alpha\)). Intervals that fall within a weekend are not excluded from the test because the frequency is zero for both empirical and model values, which has an undesired positive effect on validation. The interval width is one day. The results of the chi-square test are displayed in table 7.37. The difference between model results and empirical values is not significant in all cases.

<table>
<thead>
<tr>
<th>Day</th>
<th>k-1</th>
<th>(\chi^2)</th>
<th>Table value</th>
<th>Significant?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday</td>
<td>8</td>
<td>5.3</td>
<td>15.5</td>
<td>No</td>
</tr>
<tr>
<td>Tuesday</td>
<td>8</td>
<td>10.0</td>
<td>15.5</td>
<td>No</td>
</tr>
<tr>
<td>Wednesday</td>
<td>8</td>
<td>13.8</td>
<td>15.5</td>
<td>No</td>
</tr>
<tr>
<td>Thursday</td>
<td>8</td>
<td>10.2</td>
<td>15.5</td>
<td>No</td>
</tr>
<tr>
<td>Friday</td>
<td>9</td>
<td>11.1</td>
<td>16.9</td>
<td>No</td>
</tr>
<tr>
<td>(\alpha = 0.05)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table 7.37: Replicative validation results*
Designing Organizational Coordination

As a visual check, we display the frequency distribution of the age of documents that were delivered on Mondays in figure 7.24.

![Graph showing frequency distribution of document age on Mondays](image)

**Figure 7.24: Frequency distribution of document age on Mondays**

7.4.4 Solution finding and consistency check

The project team was engaged in a brainstorm session about possible improvements of the current situation. This brainstorm session was based on results of the model of the current situation and supported by group support facilities such as idea generation and organization tools. Employees of the mailroom, the regional units and several managers participated in the brainstorm session. In this section we give a short and aggregated description of the alternatives; for a complete account of the results we refer to the session report (van Eijck and Bouwman, 1995).

Although the throughput time of documents can still be improved, it was not regarded as the main problem in mail logistics. All participants agreed that the problem rested with the failures made at the mail room and in the regional units. The cause of this problem was inadequate information exchange and coordination between those par-
Applying the design system: three new cases

ties. The reason for this inadequate coordination can seems logical if we take a second look at the organization structure: Dutch General is responsible for the activities of the units, but a general staff department of Dutch International is responsible for mail logistics. All this results in documents being lost in the organization, the sending and re-sending of documents between departments and eventually in very high throughput times for some documents. To be able to cope with this situation, the participants of the brainstorm session developed several alternatives, which we present below in an aggregated format.

**Alternative 1: a help desk for the mail room**

A help desk for the mail room should have insight in all processes that relate to mail logistics. Agreements between regional units about handling documents should be known here (an example is the agreement between the north unit and the business west 1 unit; the latter handles 23% of the documents of the former). Documents that arrive at a wrong destination should immediately be returned to the help desk of the mail room in order to find out quickly the right destination and in order to avoid sending the document around.

**Alternative 2: service corners**

The regional units are concentrated in the building on five floors. It might be more effective to let the regional mail be handled by special service corners that serve a whole floor. However, this only works if the business and private units of a specific region are located on the same floor, which is not the case. Another removal of personnel would do the organization no good, so this alternative was disqualified.

**Alternative 3: a central mail room / sorting on team**

This alternative concerns the development of a central facility in the organization that handles all issues that relate to mail logistics. The actors of the mail room would sort and assess destinations as detailed as teams within a unit. To solve capacity problems, preprocessors of several units could be transferred to this central mail facility.
Designing Organizational Coordination

Alternative 4: shorten handling times by information exchange

For this alternative, all rules and agreements that relate to mail handling should be assessed and stored in an inquiry system. Employees of the mail room can retrieve all information that is needed to correctly address the document from this system by typing in one or more key fields such as policy number or intermediary number. Each assessor is equipped with a printer that prints all relevant information on semi-adhesive labels that are attached to the documents.

The first thing a preprocessor or team member should do is read the label instead of consulting the inquiry system himself. The label should reveal the following information:

- Date.
- Intermediary number.
- Policy or loss claim number.
- Division (business or private).
- Regional unit.
- Product.
- Team number (the actual destination).

The rules and agreements that are recorded in the inquiry system are developed by the regional units, but managed by the mail room. A change in the system concerning rules or agreements of a unit is requested from the mail room which effectuates the change.

An integral solution: alternatives 1, 3 and 4

After several discussions with management, it was concluded that alternatives 1, 3 and 4 were not mutually exclusive. If combined, they could be very effective. This decision led to the request of Dutch General management to test this solution in the simulation model. This results in two treatments: treatment 0 concerning the current situation and treatment 1 concerning the integral solution. The simulation model was changed in the following way to arrive at treatment 1:
Applying the design system: three new cases

- All regional units release one preprocessor each. The preprocessor that is left only sorts on product code.
- The mail room is extended with a help desk where returned documents are assessed. All documents that do not arrive at the right location are sent to this help desk. The help desk is able to send 90% of the returned documents to the right location. Documents that are returned to the help desk twice are correctly addressed in all cases. Assessment time of the help desk employees is a triangular distribution: tria(30 seconds, 1.5 minute, 10 minutes).
- The mail room is increased with a number of extra assessors. Destination is now determined on team level. The assessors are supported by a new inquiry system and adhesive label printers.
- Mail is sent to the regional units every two hours instead of letting the preprocessors collect the mail.

The new situation is shown in the network model of figure 7.25.

**Figure 7.25: Network model of the integral solution**

The costs of this alternative are estimated at DFL. 400,000.=- of annual costs (inquiry system, printers, extra labor costs) and the annual profits are estimated to be more than DFL. 1.5 million (resulting from the discharge of 19 preprocessors). The net annual result of this alternative is therefore estimated at about DFL. 1 million.
Designing Organizational Coordination

Consistency

The network model shown in figure 7.25 shows that the implementation of the alternative in the simulation model has no influence on consistency between the simulation model and the modeling concepts.

7.4.5 Results

In this section we present the results of the two treatments that were run with the model. These treatments are basic treatments, from which general conclusions can be drawn. The model was used later on to fine-tune the number of employees that is necessary to smoothly operate the central mail room.

Treatment 0

Treatment 0 concerns the current situation. We tested the results of treatment 1 on significant change by means of a t-test on the difference of the mean. In the tables, we note the mean value and the 95% confidence interval. In the tables for treatment one we also give the $|t|$ values and the significance. The results for treatment 0 are given in table 7.38 which represents the throughput time values.

<table>
<thead>
<tr>
<th>Output variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documents arrived within one day</td>
<td>7.12 ± 0.01</td>
</tr>
<tr>
<td>Documents correctly addressed</td>
<td>11.80 ± 0.26</td>
</tr>
<tr>
<td>Documents sent through</td>
<td>77.4 ± 0.54</td>
</tr>
<tr>
<td>All documents</td>
<td>14.95 ± 0.26</td>
</tr>
<tr>
<td>$\alpha = 0.05, n = 10$, values determine 95% confidence int.</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.38: Results of treatment 0

Treatment 0 showed also that the efficiency rate as discussed before was below 0.20 percent (!). Average total handling time is at around 95 seconds. So 99.8 percent of the time, a document is waiting or being transported. Furthermore, 95 percent of the documents reach the correct destination directly after being handled by the mail room. This means that, overall, the mail room has a failure rate of 5 percent.
Applying the design system: three new cases

Treatment 1

Treatment 1 concerns the integral solution as discussed above. Number of replications, run-length and start-up time were again determined by the respective procedures and were not changed. The results for treatment 1 are given in table 7.39.

| Output variable                        | Value        | |t| | Sign. |
|----------------------------------------|--------------|---|---|------|
| Documents arrived within one day       | $5.59 \pm 0.007$ (-16 %) | 43.3 | YES |
| Documents correctly addressed          | $8.90 \pm 0.21$ (-25 %) | 5.7  | YES |
| Documents sent through                 | $31.58 \pm 0.47$ (-60 %) | 43.4 | YES |
| All documents                          | $9.98 \pm 0.19$ (-33 %) | 9.9  | YES |

\[ \alpha = 0.05, n_1 = 10, n_2 = 10, t_{\text{nc}}(n_1 + n_2 - 2) = t_{\text{nc}}(18) = 2.101 \]

**Table 7.39: Results of treatment 1**

From these results, it is clear that, next to the financial benefits of the integral solution, an overall improvement in throughput time of about 33 percent can be achieved. Using the model, we calculated the number of extra employees that are needed for the central mail room. After running treatment 1 a few more times with different capacities for the mail room, we concluded that 4 extra employees would be ideal (see Bouwman, 1995).

7.4.6 Conclusions and recommendations

From this final case study, we can derive some conclusions that lead to recommendations for Dutch General. We also evaluate the use of the design system for this case.

**Recommendations for Dutch General**

The processes related to mail logistics are quite complex. The conceptual models show that mail logistics are crucial in the coordination between intermediaries and regional units. The simulation model, however, shows that the performance of this system is not that bad after all. Of all documents, 95 percent is addressed correctly the first time and throughput times are acceptable if they are related to throughput
Designing Organizational Coordination

times of the primary processes that are carried out after mail handling. We conclude that, for Dutch General, part of the problem was a lack of insight and understanding on part of management of the mail logistic processes. The conceptual models and the simulation model contributed greatly to creating a better understanding. Next to this, we were able to suggest some very interesting changes in the processes by using the design system.

Our recommendation of implementing the solution as discussed in section 7.4.4 has led to immediate action taken by Dutch General. After the project was finished, an implementation team was set up to change the processes within three months. Our further advice was to analyze the primary processes of one regional unit using the design system. For this, a new project team was formed, involving one of the acceptance teams of Dutch General. We cannot report on this project because it is still in progress. For a first impression of the results, we refer to Bouwman (1995).

The design system

Although the design system was not used by the author himself in this case study, it caused no problems. However, within the project, we encountered a specific situation related to coordination that so far the conceptual models could not anticipate. The fact that the mail room has a certain number of employees that can take on different roles in different situations is difficult to capture by means of the process models. If necessary, the conditions under which an actor changes his or her role and starts carrying out another task set may be visualized by means of a state transition diagram. For the simulation model, however, role changing is no problem. The actor module of the DOC/S template supports role changing by attaching several different activities to an actor, each being activated by different conditions (see section 5.3.3 and figure 7.6).
8. Epilogue

'And when you really understand dynamic reality you never get stuck. It has forms, but the forms are capable of change.'
- Robert Pirsig

8.1 Introduction

Much like the organizational environments we were allowed to visit during our research, our design system is dynamic by nature and should be capable of change and improvement. In chapter 1 we stated that the inductive-hypothetical model cycle relates to a process of continuous improvement, following closely the goal of the Singerian inquiring system: progress. In this final chapter we evaluate our research and examine the progress we have made from three key perspectives:

1. **The research environment.** The design system was developed based on several observations of practice, and it was tested in practice, too. As the case studies of chapters 2, 6, and 7 have made important contributions to the research, they are evaluated in section 8.2.

2. **The design system.** The core of our research lies in the development of the methodology and the tools that constitute the design system as discussed in chapters 3, 4, and 5. We evaluate our design system in section 8.3.

3. **The research approach and the research philosophy** as discussed in chapter 1 have been of great importance in structuring our scientific efforts and in molding the results. We find that an account of the research as carried out would be incomplete without a reflection on the research approach and the research philosophy. We discuss this subject in section 8.4.
Designing Organizational Coordination

Checkland (1985) refers to the above threefold distinction as the application area ('A'), the intellectual framework ('F'), and the methodology ('M'). We conclude this chapter by presenting an agenda for further research.

8.2 The research environment

The research presented in this thesis is strongly influenced by experience from practice. We argued that a design system that aspires to practical use should start with an investigation of practice and be tested in practice, too. Doing so, the inductive case studies presented in chapter 2 provided the requirements for our design system. We applied the design system in the case studies of chapter 6 and 7. We found both efforts very stimulating and valuable for the research.

Although these processes were already highly standardized, the organizations we visited faced several coordination problems in their primary processes. The case studies show that communication between actors is a key factor in solving these coordination problems. In the Dutch Telecom case, we see that a more structured way of communication can speed up the process. The Dutch Life case studies show that communication between intermediaries and insurance companies can be greatly improved. The Insuraco case shows that unstructured communication (i.e. phone-calls from intermediaries) can disturb the primary processes. The Dutch General and Dutch Life cases show that the sorting and routing of information to the correct destinations can serve as a mechanism for coordinating business processes.

In this short overview, we see that the design system was applied to a variety of problems in several different organizations. In section 8.2.1 we evaluate the merits of the inductive case studies of chapter 2 and in section 8.2.3, we discuss the use of our design system in each of the case studies of chapter 6 and 7. In section 8.3.3 we extrapolate the possible application of the design system in environments other than discussed in this work.
8.2.1 The inductive case studies of chapter 2

Nedlloyd

Although Nedlloyd is one of the big players in worldwide shipping, commercially and technologically they are faced with strong competition. In the logistics sector too, organizations experience the need to become 'lean and mean' and to re-evaluate their business processes. An investigation of the problems and opportunities of this organization identified the need for a clear approach to analyze and change business processes, aiming for better coordination.

From the investigations of the Nedlloyd situation we learned that coordination between central and distributed parts of an organization can be difficult and costly, leading to high investments in information technology and high operational and travel costs. We concluded that coordination in business processes is an important issue for logistic organizations and that solving coordination problems can greatly help these organizations in improving their performance.

The large bank

The investigation of the large bank in section 2.3 surprised us, as the problems in the business process were very similar to those of Nedlloyd. The comparison between the bank and Nedlloyd led to the idea that a logistic approach toward the design of administrative business processes might be feasible and valuable.

We also learned that the distribution of activities leads to intensive and explicit communication in order to coordinate processes that involve more than one office location. Again, this was attacked by making large investments in information and communication technology.

From the first two case studies we concluded that, in order to solve coordination problems, our design system should address coordination as well as communication and should focus on processes before focusing on structure.
Designing Organizational Coordination

Dutch Telecom

At first sight, the problems we identified at Dutch Telecom seemed small and insignificant. However, once more, we looked at a situation where a logistic approach could be worthwhile. Again, communication between actors was a key factor in the coordination of the business process. Although Nedlloyd and the large bank showed standardization as the main coordinating mechanism, the Dutch Telecom case reveals that mutual adjustment between the phone operators and the service engineers leads to poor performance of the business process.

Insuraco

Insuraco showed us coordination of business processes in its full complexity. The design system could have been applied to many activities of this organization. We chose to show the full complexity of the organization in chapter 2 and pick out an administrative process to which we applied the design system in chapter 6. From the analysis of the Insuraco situation, we learned that our design system should not aim for solving all coordination problems that can be found in one organization. Instead, the design system should be developed as a tool that can be used to improve individual business processes. This resulted in the emphasis on problem demarcation in our design system.

Discussion

The similarity in coordination problems of the, in other respects very different organizations we visited in chapter 2 convinced us that a general approach for identifying and solving coordination problems in business processes is feasible and useful. The cases showed four major issues that need to be examined to improve coordination in business processes: the processes themselves, the existing mechanisms of coordination, organizational performance measures and technology for coordination support. The issues are discussed in chapter 3 and used for developing our design system in chapters 4 and 5. As these were also profound issues in the case studies in chapters 6 and 7, we conclude that our inductive case studies are a relevant frame of reference.

270
8.2.2 The case studies of chapters 6 and 7

Dutch Telecom (revisited)

We used the Dutch Telecom case for a first test of the design system. The performance of Dutch Telecom’s service process was poor in terms of response times, the number of waiting assignments and the fact that the telephone lines to the dispatchers were frequently busy. Devising a relatively simple information and communication system solved the problem. In terms of the coordinating mechanisms of Mintzberg (1979), there was a shift from mutual adjustment to standardization.

Instead of using Mintzberg’s terms, we prefer to use a distinction between implicit and explicit coordination. We define embedded coordination as built-in coordination of the business process (in this case accomplished by a work process and technology). Explicit coordination takes place where coordination of the business process is a separate task, carried out by specific actors and (in this case the task of the dispatcher in the old situation is an example). At Dutch Telecom, explicit coordination performed by the dispatcher is reduced to identifying the service engineer that can best do the job and assign the job to this engineer. The rest of the explicit coordination tasks (mutual adjustment with the engineers) is now embedded in the procedures and the technology. Also, the coordination between engineers and the billing department is now embedded in the business process.

Insuraco (revisited)

The Insuraco case study is interesting for several reasons. First, explicit coordination between the intermediaries and the loss claims department (in terms of telephone calls) has a disturbing effect on the performance of the business process. By introducing actors whose sole task is coordination, the performance of the primary business process (handling loss claims) can be greatly increased.

---

1 One can refer to these actors as coordinators.
Designing Organizational Coordination

Second, by embedding several coordination tasks in the process by means of information and communication technology, the performance of several business processes can be improved.

Third, the Insuraco case taught us about an aspect of business processes that is hard to see in practice, but can easily be revealed by simulation: stability. We showed that the Insuraco business process is not very stable and that small changes may lead to significant results.

Dutch Life policy acceptance

Dutch Life provided an excellent environment to test the design system. The case study was somewhat different from the others because of the design team consisting of people from several different backgrounds. In this case study we could test the value of the conceptual models. After all, many people in the organization had to provide input into the models and judge them on correspondence to their situation.

Using the design system we showed that Dutch Life experienced several coordination problems within their acceptance process. First of all, internally, the coordination with the medical affairs department could be improved. Second, communication with the intermediaries to clarify errors slowed down the process. Coordination with medical affairs was already embedded in the process but the problem was not the actual coordination, but the slow handling at medical affairs. Coordination with the intermediary is explicit, but can be embedded by stimulating the intermediaries to use electronic data entry systems that check the input on certain types of errors. The EDI alternatives show that the process can be improved by means of electronic communication, but Dutch Life expected these alternatives to perform much better.

Dutch Life policy surrender

In the policy surrender case the design system was applied by an independent design team. The author merely advised the design team on the use of the design system and constructed the simulation model. The other activities were mainly done by Dutch Life employees.

272
Epilogue

The case shows that coordination with the client is a cumbersome process that involves exchanges back and forth. We showed that embedding coordination by means of giving the client clear instructions of what information to provide (in terms of request forms) can improve the performance of the process. We argued that coordination can be improved even further by linking policy mutations to client events. As a result of these links, better support for the acceptor could make his tasks easier.

Dutch General mail logistics

Dutch General mail logistics taught us that the design system can also handle complex processes. In this case, the design system was applied by an undergraduate student so as to test the design system independently of the author. He judged the design system as easy to use. His experience was that the conceptual and empirical models were very clear and useful representations of the logistical processes he encountered in the project. The project showed a high degree of involvement of both management and employees responsible for mail logistics. According to Dutch General management the animations played an important role involving the employees (one of the managers of Dutch General even used the word stimulation models).

The case of Dutch General showed a logistic process where the mail room is the coordinator for the back office. However, due to differences in back office organization, it is difficult for the mail room to handle all mail correctly. The introduction of a central coordinator (the mail room help desk and a number of extra sorters) to replace the activities of the pre-processors effected improvement in throughput time and process costs.

In addition, a modest information system that prints useful information on the documents could support the coordinators of the mail room in doing their job more effectively and could embed further coordination (in terms of routing and handling) in the business process.
Designing Organizational Coordination

Discussion

One can argue that the sample of our case studies is limited because they are all in the service sector. However, the various case studies clearly show very different processes and characteristics. Despite the differences between the cases, the processes we examined all fall within our definition of a business process (def. 3.1).

We showed in chapters 6 and 7 that the design system can be successfully applied to reveal and solve coordination problems. The design system was capable of supporting the improvement of business process performance for all case studies in chapters 6 and 7. This is not only our opinion, but also the view of the managers and employees who took part in the case studies. The requests for new projects by several companies of Dutch International are very rewarding in this respect.

8.2.3 Using the design system in other environments

The case studies that were presented as a five-course meal in chapter 6 and 7 lead to the desire for tasting other dishes in the same restaurant. In section 8.2.1 we argued that our design system should be applied to processes that fall within the definition of a business process we gave in chapter 3. This does not mean that the design system cannot be applied outside the environments as discussed in this thesis. Some examples where we expect that our design system can also be applied are:

- The financial industry: administrative processes in banks, insurance companies, pension funds and investment trusts.
- Government: processes in the tax office, the unemployment benefits office, local governments.
- The logistic service industry: logistic and administrative processes for shipping, transshipment, truck companies and courier services.
- Other administrative and logistic processes: in all companies where large-scale administrative or logistic processes are important, we feel that the approach as demonstrated in this thesis can be applied.
We define the requirements of an environment where the design system can be applied as follows:

1. There should be a problem that involves operational business processes, as expressed in definition 3.1, and their coordination.
2. These processes should be carried out on a large scale or be highly repetitive.
3. It should be possible to form a project team with operational and managerial involvement, creating sufficient support for the project.

8.3 The design system

We stated in section 1.4 that 'the aim of our research is the development of an inquiry system for the design of coordinated business processes.' Following Sol (1982), we defined an inquiry system as a structured set of instruments, to be used as a context for problem solving activities, expressing a methodology in view of a problem area.

Throughout this thesis we called our inquiry system 'the design system', consisting of a methodology, proposed in chapter 4 and a set of tools, discussed in chapter 5. In the previous section we concluded that the design system meets the demands of practice. In this section we evaluate our design system on its own merits and conclude that it meets the requirements as set out in general terms in chapter 1, and in more specific terms in chapter 3 and 5.

8.3.1 The methodology

The central question we now have to answer about our research is: have we constructed a methodology that really supports the design of coordinated business processes? The affirmative answer is based partly on the results of the case studies in chapter 6 and 7, and partly on an evaluation of the methodology as such. For this evaluation we reflect on the way of thinking, the working method, the modeling approach and the management approach.
Designing Organizational Coordination

Way of thinking

In section 4.2 we stated that our way of thinking rests on three important foundations: a worldview (1), a focus on the problem domain (2) and a methodological perspective (3). We comment on the merits of this way of thinking for the use of our methodology.

1. Our worldview is characterized by the identification of organizations as dynamic systems of interacting objects. The organizational designer benefits from this worldview in making conceptual and empirical models. In Pirsig's terms this provides a knife that can be used to split the problem area into useful parts (see Pirsig, 1974). In the case studies the object descriptions were effective in representing both the structure and the processes of the situation. We conclude that an object-oriented worldview is appropriate for capturing both the structure and the process of the system under consideration.

2. Our focus on the problem domain demarcates the problem area in which the design system should be applied. We defined our problem domain as interdependent business processes in organizations and selected the case studies within the boundaries of this domain focus. As a result of this domain focus, the design system relates more to organizational processes than to organizational structure. After conducting the case studies we are encouraged in our opinion that structure serves process and not the other way around. This also means that, in our view, an organizational structure should be built to support and maintain the desired organizational processes.

3. Two principles are central to our methodological perspective: the view of business process design as a problem-solving effort, and the fact that the existing situation should explicitly be taken into account. The problem-solving approach splits the process of using the design system up into four phases and six activities (see figure 1.4). The distinction between understanding and designing in modeling gives the organizational designer a yardstick for generating alternatives, a thorough insight into how things work and a vehicle for communication. During the case studies we discovered that this insight was very important for the designers as well as the managers and employees of the organization under consideration.

276
Working method

Our working method is closely related to the problem solving cycle and basically consists of four activities:

1. Conceptualization.
2. Specification.
3. Verification and validation.
4. Solution finding and consistency check.

The goal of conceptualization is to provide a vehicle for communication about the problem situation. The three activities of conceptualization: delineation of the problem area, construction of the models and deriving a problem statement, were carried out for each case study and helped the respective design teams to focus their minds and activities. Each time, the results of conceptualization were laid down in a conceptualization document, as a milestone and for later reference. The conceptual models were effective means of communication and were also useful as an aid for the construction of the simulation models. See below for a discussion on the model types.

The specification phase consists of 6 activities, eventually laid down in a specification report. Bots (1989, p.34) argues that specification is essentially no more than translating the conceptual model. However, our case studies prove, however, that there is more than that to specification. As illustrated in section 4.4.3, the translation as such is relatively straightforward and leads to a simulation model devoid of data. We find that the core of specification does not lie in the translation itself but in the other 5 activities. We have shown in chapter 5 that, for the greater part, the construction of a simulation model can be automated and supported. However, the specification of output variables and model reductions, the collection of input data, the construction of an animation and the specification of an initial treatment are, in our opinion, more crucial in arriving at a valid model than the mere translation of conceptual models. Therefore our advice is to do the specification thoroughly and accurately, with an emphasis on model reductions and the collection of empirical data.
Designing Organizational Coordination

In carrying out the case studies we discovered that verification and validation is thought of as being relatively unimportant by the managers and employees concerned. The case organizations would all rather focus on model construction and gathering results from the simulation models. However, this does not mean to say that verification and validation should not be performed. As the models are meant as yardsticks for evaluating alternatives, they must show a high degree of correspondence to the actual situation. Therefore, verification and validation should be performed accurately. Whenever possible, structural and replicative validation techniques should be used. We conclude that, where the design system is used in projects in organizations, there is a risk of underestimating the need for thorough verification and validation, also because of the statistical expertise needed for a correct replicative validation.

With respect to solution finding, we distinguished between generating alternatives and evaluating alternatives. The design system does not give extensive support to the generation of alternatives. In the case of Dutch General Mail Logistics we used a Group Decision Room session for alternative generation, based on the model of the current situation. An analysis of the empirical model of the current situation usually reveals bottlenecks and opportunities for change. The animation may help to visualize the processes and thereby make people think of alternative ideas. We found that it is very important to confront several stakeholders in the problem with the models and the model results of the existing situation (i.e. management and employees).

The evaluation of alternatives has been extensively demonstrated in the case studies and shows the actual power of simulation in business process design. The case studies show that a check of the consistency between the alternatives and the modeling concepts did not lead to any problems. We conclude that the expressive power of the models was sufficient for both the exiting situations encountered and the alternatives that were generated.
Epilogue

Modeling approach

The modeling approach is the core of the design system. In each case study we developed conceptual and empirical models according to the framework and model types of section 4.4 (see figure 4.2). In all case studies it was feasible and useful to construct and validate the conceptual and empirical models we needed for an accurate description of the current situation. We did not encounter situations that needed a 'work-around' or a solution that lies outside the expressive power of the models. Several observations lead us to the conclusion that our modeling concepts provide good support for the analysis and design of well-coordinated business processes:

1. The network, process and actor models provide clear insight into the business processes as encountered in the 5 case studies of chapter 6 and 7. During the projects, the conceptual models were used extensively in communication with the organization involved.

2. We have demonstrated that a generic transformation of conceptual models into empirical models is feasible and straightforward. We applied the transformation procedure successfully in the Dutch Telecom case (for other applications of the transformation procedure see Vreede, 1995).

3. The empirical models were evaluated by several problem owners as good dynamic representations of interdependent business processes. The undergraduate student who applied the design system in the Dutch General case, too, judged the models as ‘easy to construct and use, sufficiently expressive and an appropriate means of representing the business process under consideration’ (see Bouwman, 1995, p. 90).

These observations may not lead to the conclusion that the design system can (or should) be used in every organizational process. The design system was built, based on our definition of a business process (def. 3.1) and is intended to be used for well-defined, or definable organizational processes.
Designing Organizational Coordination

Management approach

With respect to the management approach, we conclude that the three reports as discussed in section 4.5 can serve as milestones in a project with the design system. For smaller projects we advise to report on the activities in one single report, containing the elements as mentioned in section 4.5.

We stick to our opinion also that, if necessary, any common project management approach can be used. For smaller projects, the three reports, associated with constraints regarding time, expenditure, quality, organization and information will do as a management instrument (see also Wijnien et al. 1990).

8.3.2 The tools

Next to the methodology as discussed in chapter 4, we developed two supporting tools in chapter 5: the DOC/C tool to support conceptualization and the DOC/S tool for specification. In this section we evaluate the use of these tools.

The DOC/C tool

In the case studies in chapter 6 and 7, the conceptualization tool is used extensively for the construction of network models, process models and actor models. The tool provides an environment in which these models can be constructed quickly and in a straightforward manner. The models display a high graphical quality and are easy to read.

Working with the tool did not generate any problems, neither for the author, nor for the other members of the various design teams. Since we do not have any reasons to change our modeling approach, we do not find it necessary to adapt the tool with respect to the model types it supports either.

Due to time constraints, we did not invest any efforts in developing automatic interaction between the DOC/C tool and the DOC/S tool.

280
Epilogue

However, we feel that it is technically possible to construct an interface that transforms the conceptual models made with the DOC/C tool into code that can be interpreted by the DOC/S tool. Further research could point in the direction of integrating the conceptual and empirical modeling tools so as to arrive at a more comprehensive support environment.

The DOC/S tool

Although the DOC/S tool is still a prototype, we have met almost all requirements as specified in chapter 5 (see table 5.11). Further developments should point in the direction of meeting the requirements that are left, such as handling prioritized items and temporary capacity changes for actors. Furthermore, we feel that a specification tool should support a wide range of statistical functionality, which we did not implement due to time constraints. Statistical features can be easily attached to DOC/S in the future, in order to make validation and alternative analysis easier.

8.3.3 Further improvements to the design system

Often, an evaluation results in asking if there are any open ends to be filled in. An answer to this question begins about fifteen years ago when research in the direction of dynamic modeling to support the development of information systems started. In fifteen years of research, design systems, methodologies and tools have been developed in which dynamics, process perspective, object-orientation and problem solving are key words. The design systems have been applied in many cases and environments. From the cases where problems were encountered we have learned and improved our approaches.

From the above, one might conclude that there is little room for improvement with respect to our design system. However, apart from the further refinements to the tools as discussed above, we feel that more research has to be done with several objectives:
Designing Organizational Coordination

1. From the experiences of working with the design system in the case studies of chapter 6 and 7 we conclude that our approach toward the design of business processes is mature but still in a test phase. The design system should now be used, by organizational designers in practice. The support tools should be used by others for a more independent test.

2. To broaden the problem domain of the design system, research should be conducted in which it is applied to organizational processes that lie outside our definition of a business process. The design system could, for example, be applied to policy development within governmental departments, strategy formulation processes within large organizations, or budgeting processes at universities. Application in these environments and onto these processes may lead to useful extensions of the design system, developing new concepts in order to arrive at a more generic design system for well-coordinated organizational processes.

3. Finally, a comparative study should be conducted of methodologies that aspire to support the design of business processes in order to assess the relative value of our own design system. Such a study would require another research philosophy and a more deductive research approach.

8.4 Research philosophy and research approach

A purposeful activity is characterized by two elements: an eventual goal to which the activity is supposed to lead (the product) and a set of assumptions and rules that define the way in which that goal should be pursued (the process). The process must be chosen in such a way as to increase the chances of arriving at the desired product. A better fit between product and process leads to an even more purposeful activity.

---

2 More purposeful in terms of arriving closer to the goal or reaching more of the goals (relating to effectiveness) or in terms of reaching the goal faster or cheaper (relating to efficiency).
Organizations are in constant search for a better fit between their products and their processes. It is not the primary business process itself that leads to progress and improvement, but the reflection on this business process in terms of its fit with the product it is designed to achieve.

During the four years in which this research was carried out we have been trying to establish a good fit between our product (the design system) and our process (the inductive-hypothetical model cycle). In establishing this good fit, the reflections demanded by our research approach have indeed led to progress in terms of our design system.

Science should be in constant search for a better fit between its processes and its products. It is not the scientific product that makes science so interesting, but the reflection on its process.

A scientific environment allows more freedom for reflection than an organizational environment. Therefore, in many cases, organizations have to rely on science for improvement. If, as in our case, the object of scientific activity is organizational activity, then science has to rely on organizations for the input into its process. It is this mutual interdependence that we have found so stimulating in the past four years. From this interdependence between our scientific activities and the organizational activities which we analyzed, we conclude that the Singerian inquiring system, and the inductive-hypothetical model cycle we derived from it, fit the purpose of developing a design system for organizational coordination.

Taking into account the design system and the application of it in practice, we conclude that we have succeeded in creating the cooperative environment that forms our third reason for choosing the Singerian inquiring system (section 1.5.1, p. 21): where we arrived at that goal that will help organizations attaining their goals (see Churchman, 1971 p.200).

Summarizing, we have contributed to a better fit between product and process in three different ways:
Designing Organizational Coordination

1. Our design system is able to help organizations find a better fit between business process and business product. The emphasis on understanding in our design system stimulates organizations to reflect on their existing situation and the ways in which they can strive for a better fit as described above.

2. By demonstrating the value of the inductive-hypothetical model cycle in our research, we have shown that the Singerian inquiring system (on which the inductive-hypothetical model cycle is based) fits our objective of developing a design system.

3. By reflecting on the way of working in our design system as well as on our scientific method, we have managed to improve our personal fit between our objectives and our activities.

8.5 Further research

By following the inductive-hypothetical model cycle, by investigation and application in an interesting and complex problem domain, we have reached at our goal: a design system for organizational coordination. In this section we indicate several directions for further research from what we learned.

In the first place, the design system has only been applied on the limited scale of the case studies carried out within this research project. The design system could now be investigated for its applicability to other problem domains and to processes that fall outside our definition of a business process. We hypothesize that, if only partly, our modeling approach will be applicable to these other processes as well.

Second, the issue of coordination in business processes could be investigated from other perspectives. Research could be directed to the relationships between certain process, structure or chain aspects of organizations and certain mechanisms of coordination (see for example Straatman, 1996 and Bruggeling, 1996). Our design system might be used to identify these relationships within case studies.
Further research could also relate to other uses of the design system. The models and animations made with the design system can be used for training purposes. It is already possible to use simulation models as engines that support parts of business processes on a real time basis. Research could be directed so as to extend the DOC/S tool for the purpose of real time process support.

Another direction worth investigating is the use of the design system by design teams as groups. The DOC/S and DOC/C tools are currently stand-alone applications, but have the potential of being used as group support tools. We hypothesize that combinations of simulation and animation modeling and group support facilities can be very effective in improving organizational processes. Research into this subject may even lead to a group design system for organizational coordination.

In this final chapter, we looked back on the design system, its development process, its application in practice and possible directions for further research. We are confident that we were able to build on the very interesting research tradition of dynamic modeling, and to lay the groundwork for answering the questions that remain about designing organizational coordination.
Designing Organizational Coordination
References

"The scientists of the Institute thus discovered the driving force behind all change, development and innovation in life, which was this: Herring Sandwiches. They published a paper to this extent which was widely criticised as being extremely stupid."
- Douglas Adams


ADAMS, D., Life, the Universe and Everything, Pocket Books, New York, 1982.


Designing Organizational Coordination

BIRTWISTLE, G.M., B. LIBLONG AND B. WYVILL, SIMULA Bibliography, Report 84/159/17, Computer Science Department, University of Calgary, Calgar, 1984.
BOOCH, G., Object-oriented analysis and design with applications (2nd ed.), Benjamin Cummings, Redwood City, 1994.
References


Designing Organizational Coordination


proceedings of the 1993 System Dynamics conference (pp 93-102), ISBN 92-9129-003-3, CanCun, Mexico, 1993


Designing Organizational Coordination


References


JOOSTEN, S AND G. AUSSEMS, Met de stroom mee ?, in: Computable, 16 december, 1994 (in Dutch).


KIESER, A. AND H. KUBICEK, Organisation 2. auflage, Walter de Gruyter, Berlin / New York, 1983 (in German)


KUIJLAARS, G.F.S.J., Coordinatie bij projecten georganiseerde crimina-

Designing Organizational Coordination

References


MEER, K. VAN DER, Documentaire Informatiesystemen, 2e herziene druk, NBLC uitgeverij, Den Haag, 1994 (in Dutch).


MITROFF, I.I., F. BETZ, L.R. PONDY EN F. SAGASTI, On managing science in the systems age: two schemas for the study of science as a whole systems phenomenon, TIMS Interfaces, vol. 4, no. 3, 1974


PANKO, R.R., End User Computing, management, applications and technology, John Wiley and Sons, New York, 1988


Designing Organizational Coordination

PICOT, A., AND R. REICHWALD, Bürokommunikation: Leitsätze für anwender, CW-Publikationen, München, 1984 (in German)


SHANK, J.K., AND V. GOVINDARAJAN, Strategic cost analysis, the revolution from managerial to strategic accounting, Irwin, Homewood, 1989.


References


STRENG, R.J., C.F. EKERING, E. VAN HECK AND J.F. SCHULZ (EDS), Scientific Research on EDI; bringing worlds together, Samsom publishers, Alphen a/d Rijn the Netherlands, 1992.

297
Designing Organizational Coordination

TURNER, W.S., Project Auditing Methodology, North Holland, Amsterdam, 1980.
References


Designing Organizational Coordination

### Author index

'It's publishing magic. Once you've got a name like that, then whether you can actually write or not is a minor matter.'

- Douglas Adams

<table>
<thead>
<tr>
<th>Author</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aalst</td>
<td>92</td>
</tr>
<tr>
<td>Ackoff</td>
<td>6, 62, 79</td>
</tr>
<tr>
<td>Adams</td>
<td>27, 77, 143, 185</td>
</tr>
<tr>
<td>Anderson</td>
<td>72</td>
</tr>
<tr>
<td>Attema</td>
<td>36, 41, 52, 170, 175</td>
</tr>
<tr>
<td>Aussems</td>
<td>72</td>
</tr>
<tr>
<td>Ballou</td>
<td>56</td>
</tr>
<tr>
<td>Benbasat</td>
<td>24</td>
</tr>
<tr>
<td>Benjamin</td>
<td>74</td>
</tr>
<tr>
<td>Bolter</td>
<td>15</td>
</tr>
<tr>
<td>Booch</td>
<td>23, 80</td>
</tr>
<tr>
<td>Bosman</td>
<td>4, 21</td>
</tr>
<tr>
<td>Bots</td>
<td>9, 56, 57, 63, 71, 80-84, 93, 96, 101, 103, 277</td>
</tr>
<tr>
<td>Bouwman</td>
<td>88, 120, 131, 135, 258, 260, 265</td>
</tr>
<tr>
<td>Brache</td>
<td>67</td>
</tr>
<tr>
<td>Bruggeling</td>
<td>284</td>
</tr>
<tr>
<td>Clark</td>
<td>24</td>
</tr>
<tr>
<td>Cox</td>
<td>69</td>
</tr>
<tr>
<td>Creemers</td>
<td>36</td>
</tr>
<tr>
<td>Crosby</td>
<td>70</td>
</tr>
<tr>
<td>Crosslin</td>
<td>92</td>
</tr>
<tr>
<td>Crowston</td>
<td>7, 70</td>
</tr>
<tr>
<td>Davenport</td>
<td>10, 56</td>
</tr>
<tr>
<td>Dennis</td>
<td>56</td>
</tr>
<tr>
<td>Deurvorst</td>
<td>72</td>
</tr>
<tr>
<td>Drucker</td>
<td>10</td>
</tr>
<tr>
<td>Dur</td>
<td>10, 57, 62, 77, 80, 93, 99, 101</td>
</tr>
<tr>
<td>Eijck</td>
<td>17, 36, 41, 52, 74, 77, 105, 118, 150, 161, 170, 175, 203, 260</td>
</tr>
<tr>
<td>Emery</td>
<td>6, 79</td>
</tr>
<tr>
<td>Faludi</td>
<td>1</td>
</tr>
<tr>
<td>Farace</td>
<td>12</td>
</tr>
<tr>
<td>Flores</td>
<td>70</td>
</tr>
<tr>
<td>Fried</td>
<td>56</td>
</tr>
<tr>
<td>Galbraith</td>
<td>5, 10</td>
</tr>
</tbody>
</table>
Designing Organizational Coordination

Goldratt ........................................... 69
Gore ............................................. 115
Govindarajan .................................... 67
Grimmett ........................................ 69
Gulick ........................................... 4, 6, 8

H
Hammer .......................................... 10, 12, 55, 251
Hee ................................................. 92
Herik ............................................. 88
Hirschheim ....................................... 72
Holt ............................................... 7, 92
Horngren ........................................ 67
Huston ........................................... 97

J
Joosten ........................................... 72

K
Kaplan ............................................ 56
Keen ............................................. 2, 12, 14, 15
Kickert ........................................... 7, 10
Kieser ........................................... 6, 11, 52
Kramer ........................................... 79
Kubicek ........................................... 6, 11, 52
Kuijlaars ......................................... 120, 131, 135

L
Leeuw ............................................ 79
Linden ........................................... 67
Lochovsky ........................................ 96

M
Malone ........................................... 7, 70
Mangurian ........................................ 12
March ............................................. 97
Mason ............................................ 21
Mattessich ....................................... 22
Meel ............................................. 10, 19-20, 55, 77, 85-88, 141
Meer ............................................. 15, 72
Meinsma .......................................... 116
Mintzberg ....................................... 6, 8, 11, 271
Mitroff .......................................... 21, 81
Mooney .......................................... 4, 10
Motshagen ...................................... 77, 88
Muirdock ......................................... 56

N
Nedlloyd ......................................... 28
Norusis .......................................... 141

O
Opper ............................................ 70
Ören ............................................. 86

P
Panko ............................................. 15, 71, 73
Pegden .... 87, 103, 104, 105, 116, 130,
............................................. 149, 259
Petri ............................................. 92, 93
Picot ............................................. 14
Pirsig ............................................ 55, 70, 267, 276

Q
Qureshi ........................................... 97

R
Reichwald ....................................... 14
Reiley .......................................... 4, 10
Reisig .......................................... 92
Rummler ......................................... 67

S
Schrijver ....................................... 15, 77
Shank ........................................... 67
Shapeware ....................................... 116
Short ........................................... 5, 18, 20, 81, 97
Simon ........................................... 5, 18, 20, 81, 97
Author index

Slyke.............................................15
Smit.............................................79
Soest.................................87, 151, 152, 153
Sol... 2, 9-10, 16-22, 41, 52, 56, 62, 71,
77-86, 92, 103, 117-121, 140, 151, 203,
.............................................275
Sprague.................................15, 73
Stallings......................................15
Stirzaker.................................69
Straatman.................................284
Streng.........14, 15, 58, 74, 75, 77, 104

T
Theeuwes.................................67
Thompson.................................5, 8
Turner.......................................111

U
Uijlenbroek...............................72, 73
Urwick....................................4, 8

V
Verbraeck.................................72, 77, 110
Verhoef....................................19
Veryard.....................................7
Vreede 70, 77, 86, 88, 90, 97, 105, 110,
.............................................117, 141, 279

W
Wierda.................................20, 74, 75, 77, 88
Wijers.....................................19
Wijnen.....................................280

Y
Yin..........................................24, 27
Yourdon...................................87

Z
Zeigler.....................................86
Designing Organizational Coordination
Subject index

'When traditional rationality divides the world into subjects and objects, it shuts out Quality and when you're really stuck it's Quality, not any subjects or objects that tells you where you ought to go.'
- Robert Pirsig

A

B
basic classes ......................... 93, 94 bounded rationality ................. 81 business unit .. 1, 42-47, 163, 187, 216, .................................................. 236, 238, 239, 247

C

D
decision........... 8, 60, 70, 88-89, 96-100,
Designing Organizational Coordination

107-110, 124, 128, 187, 62
decision makers 88, 110
decision making 8
decision rules 96, 108
delay 96, 106, 127-133, 190
descriptive empirical model 22, 25, 82
design approach 1

design system 16-28, 37, 40-41, 51-77, ...
... 83, 138-143, 163, 183-188, 215-216, ...
... 234-240, 265-285
direct supervision 6, 9
dynamic modeling 77, 115, 281

E
empirical data 103-104, 120, 171, 259
empirical model 22-26, 80-91, 103, ...
... 110-115, 273, 278-281
experiment 85-86, 91, 103-113
experiment model 91, 110-113
experimental design 91, 103, 141
expressive power 279

F
feedback coordination 6, 52
financial industry 274
fit 20, 55, 258, 283-284
future coordination 6

H
heuristics 16
hierarchy 97, 132

I
information system 7, 19, 35, 44-46, ...
... 52, 71-77, 115, 273, 281
information technology 3, 53, 235, ...
... 269
initial treatment 86, 123, 151, 174, ...
... 202, 227, 257, 277
initialization conditions 86, 111, 123,

inquiring system 16, 20-24, 267, ...
... 283-284
inquiry system 16, 189, 242, 262-263, ...
... 275
insurance company 11-12, 25, 33, ...
... 37-45, 59, 65-66, 163, 185-186, 216, ...
... 236-238
intellectual framework 268
interdependence 5, 10, 53, 59, 65, 81, ...
... 283
intermediary 13-14, 164-167, 177-179, ...
... 186-217, 226, 230, 235, 240-245, 262
item 57-60, 68-73, 93-96, 101, ...
... 103-109, 120, 124, 126-130, 145, 153, ...
... 166, 172, 190-191, 200, 218
item processor 93, 94, 96

K
key factor 13, 268, 270
keystroke 128, 129

L
liaison manager 45-50, 59, 164-166, ...
... 170-173
link 40, 94, 96, 105-107, 126, 129-130, ...
... 165, 190-191, 218

M
management approach 17-19, 78-79, ...
... 111, 280
market 1-2, 32-35, 46-47, 185-187, ...
... 216, 237
mental model 110
message 12, 46, 64, 75, 93, 96, 105
methodology 16-20, 77-81, 111,
... 115-125, 138, 140, 162, 267-280
middle out 111
model reductions 85, 121-122, 149, ...
... 171, 198, 224, 251, 277
modeling approach 17-19, 67, 77-92,
Subject index

R
reciprocal interdependence..................5
reduction .. 121-124, 176-179, 197, 233
reduction operators ......................121-124
replication..........................151, 204, 229
replicative validation...............87, 153, 171,
                 175-176, 197, 203-204, 224, 229, 258,
repository..... 95-96, 100, 106, 125-130,
                    145, 165, 190, 218, 244
research approach16, 19, 267, 282-283
research objective.................4, 17, 19
research philosophy..............267, 282
run control........................86, 111, 123, 151

S
sequential interdependence..........5
service industry........................1, 2, 274
set of instruments.....................16, 17, 275
simulation code. 85, 107-108, 119, 153,
                    175, 203, 229, 258
simulation experiment...............86
simulation language...............85, 91,
                               103-104, 119, 150
simulation run........................151
Singerian inquiring system21, 24, 267,
                                283
specification filter...............91
standardization ..................6, 8, 270, 271
standardization of outputs.........6
standardization of skills...........6
standardization processes...........6
structural validation..........87, 110
substantive rationality..........81
switch..........................57, 131, 134, 145-150
systems approach...................5, 79

T
task structure...57, 66, 93-96, 101-103
TeamGraphics........................117
template ..116-120, 124, 130-141, 150,
Designing Organizational Coordination

........................................ 196, 223, 250
theory ................................... 4, 20, 23, 53
tracing .................................. 87, 189
transformation .......................... 279
treatment 45, 70, 86-89, 111, 123, 151,
... 174-181, 202-212, 227-235, 257-265,
.............................................. 277

U

units .......... 4, 8, 42, 81, 187, 238-255,
............................................. 260-265
user interface... 71, 120, 123, 129-132,
.............................................. 141

V

validation........... 82- 87, 110-112, 123,
.. 141-143, 153, 171-176, 188, 196-204,
............................................ 224-230, 258-259, 277
verification... 86-87, 110-112, 123, 129,
............................................ 143, 153, 188, 278

W

way of thinking........... 17, 78, 119, 276
Weltanschauung............ 17, 23, 78
workflow management...... 71- 74, 187
working method... 17-19, 67, 78-79, 83,
.......................................... 89-90, 111-116, 277
worldview .................. 20, 23, 79, 276
Het ontwerpen van organisatorische coördinatie

'En op de vraag, waarmee het vlugschrift eindigt: "Wilt u hier meer van weten?"
zie wij zo vrij te antwoorden: "Gaarne. Maar zoudt ge dat ook zelf niet willen?"
- Godfried Bomans

Inleiding

De laatste tien jaar heeft de dienstverlenende industrie sterk aan belang gewonnen. Door gestage groei en door vele fusies zijn grote organisaties ontstaan in de dienstverlenende branches zoals in de financiële sector (banken, verzekeringmaatschappijen en pensioenfondsen), in de logistieke sector (luchtaartmaatschappijen, rederijen en transportorganisaties), en recentelijk ook in de media- en amusementswelond. Typerend voor deze industriën is het feit dat informatie een grote rol speelt in de primaire bedrijfssprocessen. Het is voor deze organisaties cruciaal dat de juiste informatie op het juiste tijdstip op de juiste plaats is.

Ter behoud van een strategische voorsprong op de concurrentie zorgen veel dienstverlenende bedrijven voor nauwe samenwerking in de keten van dienstverlening. In een dynamische en soms bedreigende omgeving ontstaan samenwerkingsverbanden en fusies, ook wanneer het internationale markten betreft. Veelal leidt dit tot samenwerking op afstand, hetgeen invloed heeft op de onderlinge afstemming van de verschillende bedrijfssprocessen. Het belang van informatie in deze bedrijfssprocessen stelt eisen aan de te gebruiken technologieën voor communicatie en informatieverwerking.

---

ISBN: 90-5638-007-9
Designing Organizational Coordination

Naast de ontwikkeling van samenwerking in ketenverband zien vele organisaties tevens een mogelijkheid om hun structuur en interne processen efficiënter en effectiever in te richten. Bij deze veranderingsprocessen krijgen organisaties te maken met het oplossen van coördinatievraagstukken rond hun primaire processen, zowel ten behoeve van de interne bedrijfsvoering als in het kader van samenwerking in de keten. Voor de oplossing van deze coördinatievraagstukken is het niet voldoende om alleen de organisatiestructuur in kaart te brengen. In dit soort veranderingsprocessen staan met name de bedrijfssprocessen zelf ter discussie. Voor het effectief veranderen van deze bedrijfssprocessen is het van belang dat de dynamiek van deze processen expliciet in beeld wordt gebracht.

Bij de aanpak van coördinatievraagstukken kan naast het onderscheid tussen proces en structuur tevens een onderscheid worden gemaakt naar verschillende perspectieven van waaruit het vraagstuk kan worden beschouwd. Vanuit het *macro-perspectief* beschouwen we structuren van verschillende organisaties die met elkaar samenwerken en worden processen gezien als ketens van activiteiten die op elkaar moeten worden afgestemd. Vanuit het *meso-perspectief* beschouwen we één enkele organisatie met haar afdelingen die elk hun eigen processen uitvoeren. Vanuit het *micro-perspectief* zien we werkplekken en medewerkers die, ondersteund door bedrijfsmiddelen, hun taken uitvoeren en afstemmen op andere taken.

In dit proefschrift is een op simulatie gebaseerde methode voor de aanpak van genoemde coördinatievraagstukken ontwikkeld en getoetst. Ter ondersteuning van de methode is tevens een tweetal computergereedschappen ontwikkeld. Tezamen worden methode en gereedschappen aangeduid als het ‘ontwerp-systeem’. Met het ontwerpssysteem kunnen coördinatievraagstukken in organisaties worden geanalyseerd alsmede alternatieven worden bepaald en beoordeeld op effectiviteit. Met de methode worden bedrijfssprocessen en hun onderlinge coördinatie in model gebracht alsmede de prestaties van deze processen vastgesteld. Met behulp van de met de gereedschappen ontwikkelde simulatiemodellen kan vervolgens een analyse worden gemaakt van de effecten van mogelijke alternatieven.
Samenvatting

Doelstelling en methode van onderzoek

Doel van het onderzoek is de ontwikkeling van een ontwerp-systeem voor de aanpak van coördinatievraagstukken. Een ontwerp-systeem moet een organisatieontwerper ondersteunen in de analyse van organisatieproblemen en de oplossing daarvan in termen van mogelijke alternatieve inrichtingen van processen en structuren in de organisatie.

Uitgangspunt voor de ontwikkeling van zo'n ontwerp-systeem is Sol's definieëring van een inquiry system als een gestructureerde verzameling instrumenten die kan worden gebruikt als context bij het probleemoplossen. Een inquiry system formuleert een methodologie binnen een bepaald probleemgebied. Analoog hieraan formuleert ons ontwerp-systeem een methode (en gereedschappen) voor de aanpak van coördinatievraagstukken. Samenvattend wordt de volgende onderzoeksopdracht geformuleerd:

Ontwikkelen van een zo volledig mogelijke verzameling instrumenten en methoden (een ontwerp-systeem) ter ondersteuning van het ontwerp van gecooördineerde bedrijfsprocessen.

De methode binnen het ontwerp-systeem is ontwikkeld aan de hand van een raamwerk waarbij aandacht wordt besteed aan de achterliggende denkwijze, de te gebruiken werkwijze, de gebruikte modeltypen (modellerwijze) en de beheerswijze. De gereedschappen zijn ontwikkeld vanuit de eisen die de methode aan deze gereedschappen stelt.

Voor de ontwikkeling van het ontwerp-systeem staat de door Churchman (1971) gepropageerde Singeriaanse onderzoeksfilosofie centraal. De keuze voor deze filosofie heeft een drietal redenen:

1. De filosofie legt sterk de nadruk op doelgerichtheid en voortgang, en sluit derhalve nauwkeurig aan bij onze wens om probleemoplossen te ondersteunen.
Designing Organizational Coordination

2. De filosofie stimuleert het gebruik van verschillende disciplines binnen het onderzoek zodat voor elk deelprobleem in het onderzoek een adequate oplossing kan worden gevonden.

3. De filosofie propageert een omgeving waarin A (lees: de onderzoekker) op zoek is naar een manier om B (lees: organisaties) meer doelgericht te laten functioneren. Dit sluit naadloos aan bij de geformueleerde onderzoeksdoelstelling.


**Vier inductieve casusposities**

Aan de hand van een aantal interviews, rapporten, archiefmateriaal en andere bronnen is de situatie van een viertal Nederlandse dienstverlenende organisaties geanalyseerd. De organisaties betreffen

- Een grote Nederlandse rederij (Nedlloyd Lijnen).
- Een grote Nederlandse bankorganisatie.
- Een onderhoudsdienst van PTT Telecom.
- Een grote Nederlandse schadeverzekeraar.
Samenvatting

Van de situatie bij Nedlloyd leren we dat coördinatie tussen centrale en decentrale delen van de organisatie kostbaar en ingewikkeld is. De noodzaak tot coördinatie van de schepenbeheerder, de beheerder van de scheepvaartlijnen en de verschillende regionale kantoren leidt tot hoge investeringen in informatie- en communicatietechnologie alsmede reiskosten. Een goede oplossing van het coördinatieprobleem zou de effectiviteit en efficiëntie van deze organisatie sterk verbeteren.

De bankorganisatie leert ons dat er grote overeenkomsten bestaan tussen haar eigen coördinatievraagstukken en die van Nedlloyd. Ook hier leiden coördinatieproblemen tot grote investeringen in informatie- en communicatietechnologie. Communicatie speelt een belangrijke rol in de afstemming van bedrijfsprocessen, zeker als het om afstemming tussen verschillende locaties gaat.

Uit de eerste twee casus concluderen we dat het gebruik van logistieke concepten bij de analyse van bedrijfsprocessen en coördinatie in dienstverlenende organisaties haalbaar en zinvol lijkt.

De PTT Telecom case leert ons dat een effectieve uitvoering van bedrijfsprocessen ook op kleinere schaal in de weg kan worden gestaan door coördinatieproblemen. De afstemming tussen de centrale opdrachtuitgifte en de decentraal opererende servicemonteurs liet te wensen over. Opnieuw was communicatie tussen medewerkers een cruciaal gegeven. Bovendien leende de situatie zich uitstekend voor het toepassen van logistieke concepten.

De grote verzekeraar toonde ons de combinatie van een aantal uiteenlopende coördinatievraagstukken. Door de complexiteit van deze organisatie werd het duidelijk dat het ontwerp-systeem gericht moet worden op duidelijk afgrensbare problemen en een beperkte verzamelingsamenhangende processen. Tevens werd duidelijk dat het van groot belang is om eerst de huidige situatie goed te kennen voordat tot het genereren van alternatieven kan worden overgegaan.
Designing Organizational Coordination

Eisen aan een ontwerp-systeem voor coördinatie

Uitgaande van de onderzochte theoretische inzichten rond coördinatie en de bezochte praktijksituaties kunnen eisen worden geformuleerd waaraan een ontwerp-systeem voor coördinatie moet voldoen. Deze eisen worden geformuleerd aan de hand van een conceptueel raamwerk waarin een viertal aspecten aan de orde komen:

1. **Bedrijfsprocessen** moeten kunnen worden beschouwd vanuit de drie genoemde perspectieven. Dit betekent dat vanuit het micro-perspectief taken en beslissingen moeten kunnen worden geanalyseerd, vanuit het meso-perspectief activiteiten en middelen moeten worden onderscheiden en vanuit het macro-perspectief transacties moeten worden gezien.

2. **Coördinatie en coördinatievraagstukken.** Uitgaande van de bedrijfsprocessen zoals gezien vanuit de verschillende perspectieven, moet het vervolgens mogelijk zijn om de verschillende aspecten van het coördinatievraagstuk in kaart te brengen. Ook coördinatievraagstukken kunnen vanuit de drie verschillende perspectieven worden beschouwd.

3. **Prestatiemeting van bedrijfsprocessen.** Om inzicht te verkrijgen in de waarde van veranderingen in de bedrijfsprocessen, is het noodzakelijk om de prestaties van deze processen te kunnen vaststellen. Hierbij wordt onderscheid gemaakt tussen het kostenaspect, het tijdsaspect, het capaciteitsaspect en het kwaliteitsaspect van bedrijfsprocessen.

4. **Technologie voor coördinatie.** Evenals de bedrijfsprocessen en de coördinatievraagstukken kan de mogelijk te gebruiken technologie voor het oplossen van coördinatievraagstukken ook vanuit de drie perspectieven worden beschouwd. Vanuit het micro-perspectief speelt ‘personal computing’ een rol, vanuit het meso-perspectief kan de aandacht worden gevestigd op ‘workflow management systemen’ en vanuit het macro-perspectief is ‘electronic data interchange’ van belang.
Samenvatting

Een methode voor het ontwerpen van coördinatie

Op basis van de vier elementen van het voornoemde conceptueel raamwerk is gekozen voor een op discrete-event simulatie gebaseerde methode voor het ontwerpen van coördinatie. De methode wordt toegelicht aan de hand van de achterliggende denkwijze, de werkwijze, de modelleerwijze en de beheerswijze.

Denkwijze

In de methode worden organisaties en bedrijfsprocessen gezien als systemen van interacterende objecten. Het analyseren en ontwerpen van bedrijfsprocessen wordt gezien als een proces van probleemoplossen.

Werkwijze


Modelleerwijze

In de modelleerwijze wordt onderscheid gemaakt tussen conceptuele en empirische (simulatie-) modellen. De methode onderscheidt een drietal conceptuele modelltypen: met het netwerkmodel worden de actoren en opslagplaatsen van objecten ('repositories') in kaart gebracht, met het procesmodel kunnen bedrijfsprocessen worden gemodelleerd en met het actormodel kunnen de taken van een specifieke actor worden beschreven. Basis voor de modelltypen is een in dit proefschrift uitgewerkte verzameling objectklassen, gebaseerd op de inductieve casus.
Designing Organizational Coordination

Het empirische model heeft betrekking op de simulatiecode, de empirische gegevens en een experimentele opzet. Het empirische model kan worden geanalyseerd aan de hand van animatie (animatie model) alsmede aan de hand van experimentele uitvoer (experiment model).

Beheerswijze

Binnen de methode wordt niet gekozen voor een specifieke beheerswijze. Er zijn vele project management methodieken op de markt die geheel of gedeeltelijk kunnen worden gebruikt. Het wordt aanbevolen om de probleemsituatie zo nauw mogelijk te definiëren en slechts bij uiterste noodzaak uit te breiden. Projectteams kunnen klein worden gehouden en het is wenselijk om slechts op essentiële zaken te rapporteren.

Ondersteunende gereedschappen

Ter ondersteuning van de methode zijn voor zowel de conceptualisatiefase als de specificatiefase gereedschappen ontworpen. De conceptualisatiefase wordt ondersteund door een grafisch modellerprogramma waarmee de drie genoemde modeltypen kunnen worden geconstrueerd. Het programma bevat zogenaamde templates voor elk van de modeltypen waarin de bouwstenen voor dat modeltyp zijn ondergebracht.

Voor de ondersteuning van de specificatiefase is een template gebouwd binnen de ARENA™ simulatie omgeving. Hiermee kan de specificatie van de netwerkmodellen, de procesmodellen en de actormodellen eenvoudig worden gerealiseerd. Ook het ontwikkelen van animatiemodellen is met dit gereedschap zeer eenvoudig.

Toetsing van de methode in casusposities

Het ontwerp-systeem is toegepast en getoetst in een vijftal casusposities. Het ontwerp-systeem is toegepast op twee eerder besproken inductieve casus: de PTT Telecom casus en de grote verzekeraar.
Samenvatting

De PTT casus liet zien dat met relatief kleine ingrepen (de uitrusting van service monteurs met draagbare telefoons en laptop computers) de coördinatie en daarmee de prestaties van het bedrijfsproces aanzienlijk kunnen worden verbeterd.

De casus rond het acceptatie- en schadeafhandelingsproces van een grote schadeverzekeraar liet zien dat door betere coördinatie en inbedding van coördinatie in het bedrijfsproces, de verstorende invloeden van inkomend telefoonverkeer sterk kunnen worden verminderd. De casus geeft verder aan dat EDI alleen interessant is met betrekking tot de prestaties van het bedrijfsproces als daarmee de taakduur van individuele medewerkers structureel wordt verminderd.

Naast de reeds inductief beschouwde cases is een drietal nieuwe case studies uitgevoerd bij twee verschillende verzekeringsmaatschappijen van een groot financieel concern in Nederland. Het betrof hierbij respectievelijk het acceptatieproces en het afkoopproces bij een levensverzekeraar en de postlogistiek bij een schadeverzekeraar.

Bij het acceptatieproces voor levensverzekeringen bleek zowel de coördinatie met externe partijen (verzekeringstussenpersonen) als met interne actoren (met name met de afdeling medische zaken) verbeterd te kunnen worden. Met name door de inbedding van coördinatie in het primaire proces (door middel van procedurele afspraken en gemoderniseerde informatietechnologie) is het mogelijk om de gemiddelde doorlooptijd van een aanvraag met meer dan 50 procent te verbeteren.

Coördinatieverbetering in het afkoopproces spitst zich met name toe op de relatie tussen de verzekeraar en de consument. Door de consument beter inzicht te geven in de te volgen procedures bestaat er minder behoefte aan communicatie (en coördinatie) en kan de gemiddelde doorlooptijd sterk worden teruggewerkt (tot minder dan 60 % van het oorspronkelijke gemiddelde).

Toepassing van het ontwerp-systeem op de logistiek van een grote schadeverzekeraar leidde tot een verbeterde inrichting van de sorteerprocessen op de postkamer en een betere afstemming tussen de post-
kamer en de afdelingen die de tussenpersonen in de regio bedienen. Door een kleine investering worden extra gegevens op de poststukken afgedrukt waardoor de verdere sortering, routering en behandeling efficiënter en effectiever kan geschieden.

Conclusies en verder onderzoek

Met de ontwikkeling en succesvolle toepassing van het ontwerp-systeem is de eerder geformuleerde onderzoeksdoelstelling gehaald. Door toepassing van het ontwerp-systeem was het mogelijk om bedrijfprocessen en coördinatie in vijf verschillende casus te analyseren en te verbeteren. Na de uitvoering van de vijf genoemde casus is het ontwerp-systeem opnieuw op twee praktijksituaties toegepast.

Met betrekking tot de denkwijze achter het ontwerp-systeem, kan worden gezegd dat de objectgerichte benadering zinvol is voor de structurering van zowel probleemsituatie als alternatieven. De nadruk op probleemopslossen als kern voor de werkwijze geeft de ontwerper houvast in het ontwerptrject. Met name het onderscheid tussen kennen en maken is hierbij van belang.

Binnen de werkwijze blijft validatie een belangrijk aandachtspunt. Het is van belang dat het ontwerp-systeem wordt toegepast met voldoende statistische achtergrondkennis, zodat een goede validatie kan worden gewaarborgd.

De in de modelleerwijze gedefinieerde conceptuele en empirische modellen hebben sterk bijgedragen aan een goed begrip van zowel huidige als toekomstige situaties bij alle betrokkenen. Ook in de afbakening en structurering speelt de modelleerwijze een belangrijke rol.

Met betrekking tot de beheerswijze kan worden gesteld dat een eenvoudig opgezette methodiek goed voldoet. In kleinere projecten kan worden volstaan met het documenteren in een enkel rapport.
Samenvatting

Na de ontwikkeling en meervoudige toepassing van het ontwerp­systeem, kunnen we stellen dat het geschikt is voor verder gebruik in de dienstverlenende sector. Wellicht dat in de toekomst het specificatiegereedschap kan worden aangesloten op bijvoorbeeld workflow management systemen, zodat geen dubbele ontwikkeling hoeft plaats te vinden.


319
Designing Organizational Coordination
Curriculum Vitae

"The fortunes of those who have figured in this tale are nearly closed. The little that remains to their historian to relate is told in few and simple words"
- Charles Dickens

Daniel T.T. van Eijk was born in Arnhem on December 18, 1968. After graduating from the 'Katholieke Gelders Lyceum' in Arnhem in 1987, he studied Computer Science at Delft University of Technology, where he graduated from the Information Systems department in 1992. His Masters project concerned the improvement of logistics and administrative processes for a stevedore company in the port of Rotterdam. After graduation he worked as a research assistant at the newly founded school of Systems Engineering, Policy Analysis and Management where he completed his dissertation. His research was published in several journals and at international conferences in the United States, Mexico, Sweden, Greece, Belgium and the Netherlands. While being a research assistant he was a member of the faculty board. He co-developed the undergraduate course on telematic systems, and was involved as a teacher in the information systems course. He assisted four masters students in their projects and gave several invited lectures on simulation and business engineering. Currently he is working as a senior management consultant at KWW management consultants in Rotterdam.
Designing Organizational Coordination