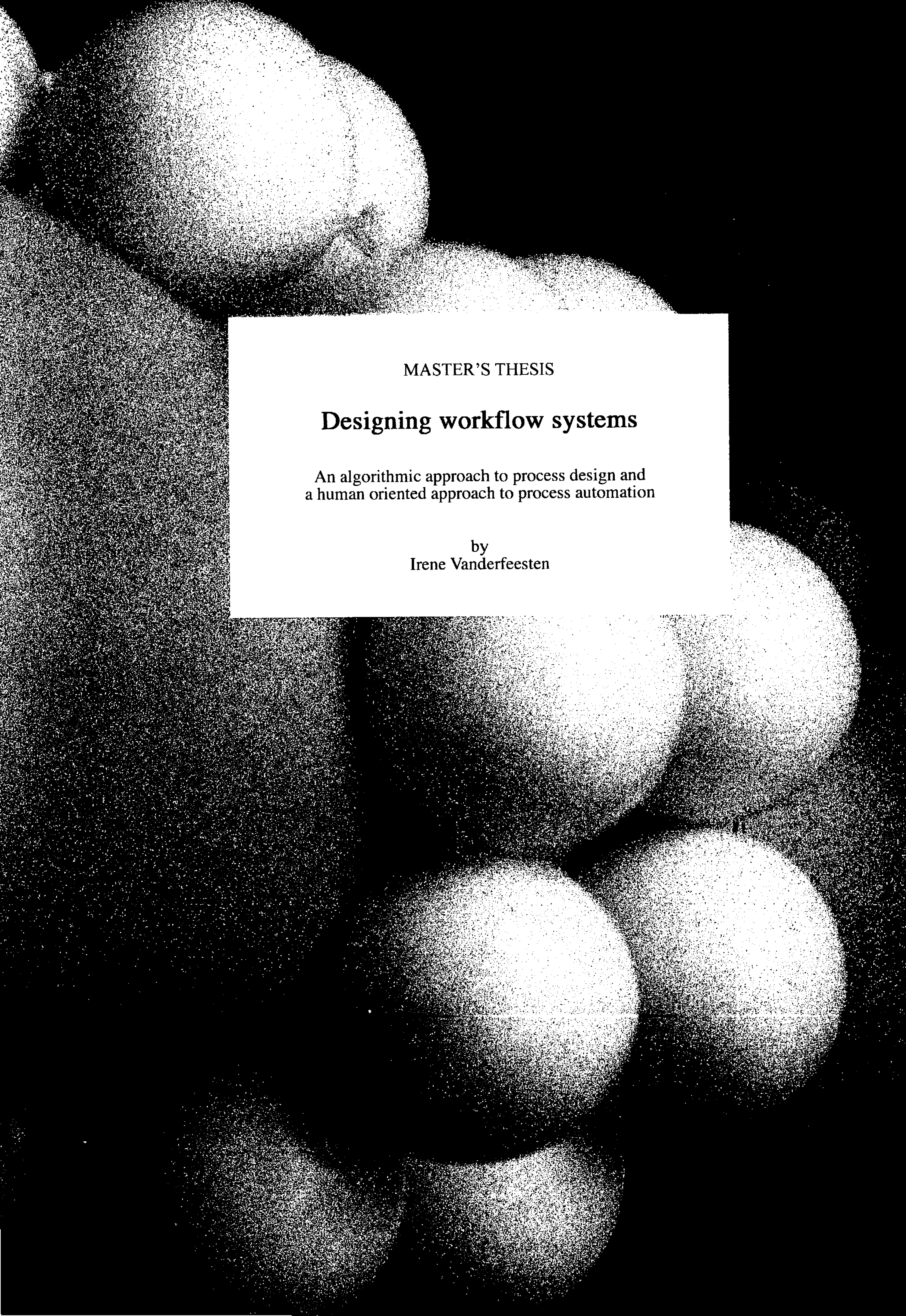
MASTER

Designing workflow systems

an algorithmic approach to process design and a human oriented approach to process automation

Vanderfeesten, Irene

Award date:
2004



MASTER'S THESIS

Designing workflow systems

An algorithmic approach to process design and
a human oriented approach to process automation

by
Irene Vanderfeesten

TECHNISCHE UNIVERSITEIT EINDHOVEN
Department of Mathematics and Computing Science

MASTER'S THESIS

Designing workflow systems

An algorithmic approach to process design and
a human oriented approach to process automation

by
Irene Vanderfeesten

Supervisors:
dr.ir. H.A. Reijers (TM-IS)
dr. M. Voorhoeve (W&I-AIS)
dr. W. van Eerde (TM-HPM)

Eindhoven, August 9, 2004

Abstract

This thesis elaborates on two approaches to workflow system design.

The first method concerns the design of the process model at buildtime. Workflow designers experience considerable freedom in designing the smaller steps (or activities) within a process. An operational notion of activity cohesion and coupling may help them to design more well-structured workflow activities. Inspired by resemblances between software programs and workflow processes, the applicability of software engineering quality metrics to workflow process design is examined. We adopted an important rule of thumb from this field, which states "strong cohesion, loose coupling". This rule of thumb is used to develop a method for supporting the design of well-structured activities in workflow processes. A well-structured activity is an activity containing a content that strongly belongs together and having as little connections to other activities as possible. Our method takes an information processing view on the process and considers the content of activities by means of their underlying operations. In this way, a workflow design is the clustering of operations into activities. New cohesion and coupling metrics for workflow design are introduced, which are integrated in a design heuristic. These metrics determine the cohesion (the degree to which the content of activities in a design belongs together) and the coupling (the degree to which he activities in the design are connected to each other) of a workflow process design. Based on these metrics the heuristic can determine the best design among alternatives and can be used by workflow designers to identify the strongly cohesive and weakly coupled process designs among several alternatives. The application of this approach is illustrated with some realistic examples.

In the second approach we consider the runtime execution of a workflow process. Here, the people that have to work within the workflow system are in the centre of attention. After all, the implementation of a workflow management system in an actual situation can only succeed when people accept the system and can perform their work properly. A danger of automation is the erosion of work, resulting in boring and monotonous work. In this part of the research we will develop some ideas to make small adaptations to the workflow system in order to make the execution of the work more pleasant. The basis of these ideas is the combination of two models from literature. The first model, from the area of organisational psychology and job design, is the Job Characteristics Model, which describes five core dimensions of a job. The second model, about assignment and synchronisation policies in workflow management systems, shows which parts of the system can be "tuned" in what way. It mainly describes the distribution and assignment of work items. A change in the tuning of the system can have a positive or negative impact on the employee who has to work in the system. By combining these two models a number of ideas are generated to improve an employees perception of the work (s)he has to perform. Most of the ideas try to affect an employee's autonomy in a positive manner. After a description of the generation of the ideas, they will be validated by carrying out a qualitative expert validation, resulting in a list of six best ideas. Finally, we will identify which workflow management systems facilitate the support of the best ideas or, when they do not, which extensions to the workflow management systems can be made to facilitate the ideas.

Preface

This Master's Thesis is the result of eleven months of research at the sub department of Information Systems of the department of Technology Management of Technische Universiteit Eindhoven. With this thesis, I conclude my study of Computer Science and I will obtain the Master of Science degree.

The past six years have never been easy for me. The greatest part of this period I have been struggling with doubts about my interest in and suitability for this field of study. From the very beginning I was interested in human and business aspects of computer systems, but already soon after starting this study I found out that its line of approach was not that broad. It took a long way about to eventually find the right place. Finally, I found an interesting area in the field of business process modelling and workflow management.

Now, in the end, I am glad that I persevered in going on with this study. I think I have been able to combine a reliable background in Computer Science with useful knowledge of industrial engineering and psychology.

While working on the research involved with this Master's Thesis, I found a very nice working environment and elaborated on interesting research questions. I am grateful for the opportunities the IS department gave me to do my final assignment, to explore more of the scientific world outside computer science, and to develop my own interests and personality. I am very proud of the final result, which lies right in front of you. I am even more proud, because the working on this research already led to the acceptance of a paper to the Business Process Management (BPM) conference (see also [69]).

I would like to take this opportunity to express my thanks to a number of people in particular, for the help and support I received during this final assignment. First of all, I would like to thank Hajo Reijers, my supervisor in this project, for convincing me not to give up at the moment I needed it most, and for our nice and close collaboration. Without his support I never would have been able to finish my study. I also want to thank my other supervisors, Marc Voorhoeve and Wendelien van Eerde, for their effort and everlasting enthusiasm in sharing their ideas with me.

Next, I would like to thank all of my colleagues at the IS department. They really made me feel at ease in our working environment, which has been a very important factor to succeed in doing this final assignment. Especially, I want to thank Boudewijn van Dongen, for his help in the realization of the tool and Eric Verbeek, for the technical support concerning the use of some workflow management systems.

I am also grateful to a lot of people who helped me out to find case studies for the cohesion and coupling metrics (from the first part of this thesis) and those who were willing to give their opinion on my ideas during the interviews I took in the second part of this research. Unfortunately, I cannot mention them all by name because of privacy reasons.

Finally, I would like to express many thanks to my relatives and friends. They have always been ready to help me, to give advice and to assist me in work and deed. They helped me to get back on my feet in rough times. Without their loving and support I would have never been able to find my way through this period of study and conclude it with a happy end. Especially, I appreciate the work Joyce has been doing to design a very nice cover for this thesis, using her own association with the subjects of this thesis.

Irene Vanderfeesten

Eindhoven, August 9, 2004.

Contents

1 Introduction

The subject of this Master's Thesis is Business Process Redesign (BPR) and Workflow Management (WfM). These two concepts are highly intertwined with each other. Usually with BPR we mean the process of (re)assessment, analysis, modelling, definition and subsequent operational implementation of the business process of an organisation. One way to do this is by the use of workflow management and its accompanying concepts of, for instance, the process model and the workflow management system.[1]

In this introduction, the reason of this research, our research questions, and the structure of this thesis will be clarified. First, the background of BPR and workflow management is explained by elaborating on the history of automation systems[2]. Afterwards, we will shortly indicate the sub area within workflow management that will be our focus. Finally, the research questions that are addressed in this thesis are introduced and the structure of this document will be clarified.

During the past decades, a shift took place in the development of information systems. Increasingly, generic tasks have been taken out of programs and put into decomposed management systems. Van der Aalst and van Hee ([5]) describe this evolution in four steps:

- **1965-1975: decompose applications.**
  During this period, information systems comprised decomposed applications, each with its own databases and definitions. The applications ran directly on the operating system and had just a simple command line interface or custom-made graphical interface.
- **1975-1985: database management – "take data management out of the applications".**
  This period is characterised by the rise of the database management system (DBMS). A database is a permanently available, integrated collection of data files, which can be used by many applications. The use of databases has the advantages that data managed by different applications can be combined, that data structures only need to be defined once, that the organisation of data can be handed over to a database management system, and that the same data item only needs to be stored once.
- **1985-1995: user-interface management - "take the user interface out of the applications".**
  During this period, the user interface was extracted from the application program. Originally user interfaces were designed by the developers screen by screen, field by field. Not only did this take up a lot of time, but also each designer had her own style, which meant that every system had to operate in a different way.
- **1995-2005: workflow management - "take the business processes out of the applications".**
  Now that data management and user interfacing have largely disappeared from applications themselves, it seems that much of the software is devoted to business processes (procedures) and the handling of cases. Therefore, it has become attractive to isolate this component and find a separate solution for it. Not only can this accelerate the development of information systems, but it also offers the added advantage that the business processes become easier to maintain. A workflow management system (WfMS) manages the workflows and organizes the routing of case data amongst the human resources and through application programs.

A state-of-the-art application should mainly coordinate between its components. We advocate a special view on automation, based on workflow management and its accompanying concepts. Although the specific terms that are used in the workflow management field can be a little confusing here, we will first give a general introduction before going more deeply into the concepts of workflow management. The next chapter will elaborate further on this subject. There, our extended model of a

[1] An explanation of the difference between BPR and Workflow management can be found in Appendix A.

[2] In chapter 2, we will further elaborate on BPR and workflow management concepts.

workflow system, which will turn out to be an important basis for the research of this thesis, will also be explained.
Nowadays, many companies use workflow management to automate and support their business processes. This is done by the use of a workflow management system. A workflow management system has a number of functions that can be used to define and track workflow processes, making both the progress of a case through a workflow and the structure of the flow itself easier to revise. It therefore is clear that workflow management systems have become the ideal tool for achieving BPR.

As the influence of workflow management and BPR grew during the past ten years, much research has been done. This thesis, elaborates on two parts. The first part is concerned with (re)designing of process models. The second part explores the human aspects of workflow systems. Following, the research of this thesis will be further introduced and the research questions will be elaborated.

At the moment, one of the pressing issues in business process (re)design is how to design a process model. To position our contribution to BPR within current research we use the classification by Kettinger et al ([48]) that gives an overview of all methods that can be used for the diagnosis, redesign, modelling and evaluation of business processes. This classification consists of three levels of abstraction (see for a description also [65]):

- **Methodology** – A methodology is the highest level of abstraction and is defined as a collection of problem-solving methods governed by a set of principles and a common philosophy for solving targeted problems. Many of these methods focus on the project management and organisational issues of process design. Their scope is either too broad or their contents are too vague to actually support the process designer. However, there are some exceptional methods on BPR that are of interest. Roughly, they can be distinguished in two categories ([65]):
  - *evolutionary approaches*: the existing process is taken as starting point, which is gradually refined or improved by using a set of best practices or rules to transform that process.
  - *revolutionary approaches*: a clean-sheet of paper is taken to design the complete process from scratch.
- **Technique** – The term technique is defined as a set of precisely described procedures for achieving a standard task. In their survey, Kettinger et al. ([48]) identified a set of 72 techniques targeted at designing processes (including for instance data flow diagramming, brainstorming, simulation, out-of-the-box thinking, workflow design).
- **Tool** – A tool is a computer software package to support one or more (BPR) techniques.

In the first part of this research, we will focus on a newly introduced revolutionary methodology to the (re)design of business processes and develop a new technique for supporting BPR. By taking an information processing view on the process, we question the size of activities (or process granularity) with respect to their content (of information processing). The research question can be formulated as follows:

How can activities in workflow processes be designed best based on the processing of information in a process? **(1)**

In other words, the aim of this research question is to develop an analytical method to design well-structured activities. A well-structured activity in this view means an activity that is cohesive itself and has as little exchange of information with other activities as possible. This is according to a rule of thumb we adopted from research on quality metrics in the area of software engineering: "strong cohesion, loose coupling". The information processing view of a process that is on the basis of this method focuses on the content of the activities by means of the underlying operations on information of the process.
Considering this view on process design, a process model is the (most favourable) clustering of operations into activities. Our research question now is accentuated to:

How can operations be clustered best into activities? **(2)**

Our method is described in chapter 3 and extends the work of Reijers on cohesive activities in workflow process design ([66]). Related work addresses Product Based Workflow Design (PBWD), (see for example: [1], [64], [65]) and quality metrics in the field of software engineering (for example: [21], [47], [89]).

The other part of this thesis will focus on human aspects of workflow system design. An important limitation to most business process redesign projects or automation projects is the abstraction from critical factors to the success of the implementation of a system, like for instance[3]:

- the degree to which the process model reflects the actual process.
- the degree to which the business goals (like improvement in throughput time, waiting time, occupation rate, etc.) are achieved by the new system.
- the quality of the end product and service level that can be achieved using the workflow system.
- user acceptance of the introduced changes and the new system.
- user-friendliness of the workflow system.

We complement our research on the size of activities in the process model by exploring the human side of workflow design (i.e. the user-friendliness of the workflow system). Our initial research question was as follows:

How can activities in workflow processes be designed best with respect to a user's experience of the work he or she has to do? **(3)**

In other words, the goal of this research question is the development of some guidelines to the design of workflow activities that should be followed to design activities that people like to work on.

A literature study revealed that our research question was too detailed. Usually a worker' contribution to a process is not limited to a single activity, but comprises a range of activities within several processes. Thus, in analysing an employee's function and discussing the quality of his or her work, we have to consider all of his or her duties together. Considering this, it is not possible to develop guidelines to the design of activities separately from other activities, because the quality of the designed activity will depend on the other activities an employee performs.
Besides this, literature on design of work within automated systems appeared to be scarce. Most of the related research is focused on for example the process of designing a system, not on the designing itself (see for more detailed information chapter 4).
Because of this marginal theoretical basis to develop guidelines on the design of activities, a less detailed view on the human approach to workflow process design is justified.

We adopted another view on the work an employee working in a workflow system has to perform, because our background in psychological theory and practices is not sound enough to examine the content of an entire job within the workflow system. The second part of this thesis will focus on the way in which the employee has to execute the work within the workflow system. (To be more clear: the focus is not on the *content* of the work (what he has to do), but on the *means and methods*

[3] The points mentioned are based on a general impression from literature. They do not necessarily give a proper overview of success factors to business process redesign.

(supported by the workflow system) for the execution of work within a certain business process (how he has to do it).) The research question for the second part of this thesis can be reformulated as follows:

In what way can a workflow system be adapted to improve an employee's experience of the work he or she is doing within the system? **(4)**

The related research to this question is that of task analysis, job design and task characteristics from socio-technical theory and organisational psychology (see for example [8], [28], [58]) on the one hand, and assignment and synchronisation policies within workflow management systems ([54]), that describe the possible ways to "tune" a workflow management system, on the other hand. The combination of these subjects has led to the development of some ideas to make small adaptations to the workflow system in order to improve an employee's experience. Particularly, by providing workers with more autonomy in their work.

The structure of this thesis is as follows. First, the meaning of workflow management and its related concepts is introduced in chapter two. As can be clear from the title, we adopted a very broad view on workflow systems that will be explained in more detail in this chapter too. Especially, our extended view of the workflow system is elaborated in section 2.4.
After that, the first research question about the design of activities is elaborated in chapter three. In chapter four, the human side of workflow systems is discussed. Finally, chapter five contains a summary of the conclusions and directions for future work.

2 Introduction Workflow Management

In this chapter some of the backgrounds on workflow management are explained. This context will make it easier for the reader to understand this thesis.
According to Van der Aalst and Van Hee ([5]), the term workflow management refers to the ideas, methods, techniques, and software used to support structured business processes. The objective of workflow management is to achieve streamlined and easy-to-maintain work processes.
This definition refers to a broad spectrum of related subjects and issues. In this chapter, the parts of workflow management that are relevant to this thesis will be clarified. As the terminology in the field of workflow management is not fixed, this thesis includes a glossary. As much as possible, we adhere to the definitions of the Workflow Management Coalition (WFMC) ([86]) and of Van der Aalst and Van Hee ([5]), although they are inconsistent with each other in some respects. In these cases, we selected the most appropriate definition or made a definition of our own, based on [5] and [86].

Workflow management systems are designed to support business processes. A business process consists of a number of steps (activities) that can be executed automatically, manually, or using a combination of these two. A workflow management system does not only support the execution of activities, but also takes care of the distribution and assignment of work items to employees, and provides the possibility to keep track of cases and to gain statistics (for instance management information) of the process and the workers. The execution of these tasks is based on a process model (or process definition in other words).

This chapter first elaborates on business processes. Next, it is explained how (the steps of) a business process can be modelled using Petri net techniques. In the third section, the Workflow Management Coalition Reference Model, a framework for workflow management systems, will be presented. Finally, our extended view on the workflow system (which is needed to describe the human and organisational aspects of workflow management) is introduced and it is clarified on which parts of this model our research focuses.

2.1 Business processes

Workflow management systems are related to business processes, as explained before. Van der Aalst and van Hee ([5]) define a business process as follows:

> *"A business process is one focused upon the production of particular products. These may be either physical products, such as an aircraft or bridge, or less tangible ones such as a design, a consultation paper, or an assessment. In other words, the "product" can also be a service."*

Thus, many kinds of business processes exist. Besides the examples mentioned in the definition above, you can think of for instance the production of bottles of soft drinks, the processing of an insurance claim, the hiring of a new employee, the assembly of a car, the ordering of office supplies, etc. Every company has its own processes. Some of them are core processes, such as the assembly of a car for a car company. Some of them are supporting processes, like the processing of invoices or the ordering of office supplies at the car company. In literature ([5]), business processes are split up into three categories of processes: primary, secondary and tertiary processes[4]. The processes that are typically supported by a workflow management system are primary and secondary processes. A workflow management system typically deals with administrative processes, such as in banks, insurance companies, and governmental agencies.

[4] For an explanation of the meaning of these terms, see Appendix B.

The processes that are supported by workflow management systems are called workflow processes. According to Reijers ([65]), a workflow process is a specific type of business process, with the following features:

- It is *case-driven*: each execution of a step in a workflow process can be attributed to exactly one specific case; there is, for example, no batch-processing within a workflow.
- It is a *make-to-order* process: a specific order or trigger starts the initiation of the process; there is, for example, no production-to-stock.

The core actions of workflow processes involve the processing of large amounts of information and therefore can be characterised as "information intensive". Compared to production (or manufacturing) processes, dealing with physical parts and products, the following remarks can be made. Although there are many similarities, there are also some logistical aspects in which an administrative process differs from a typical manufacturing process ([1]):

- *Making a copy is easy and cheap.* In contrast to making a copy of a product like a car, it is relatively easy to copy a piece of information (especially if it is in electronic form).
- *There are no real limitations with respect to the in-process inventory.* Informational products do not require much space and are easy to access (especially if they are stored in a database).
- *Quality is difficult to measure.* What is the quality of the decision to accept an insurance claim for instance?
- *Quality of end-products may vary.* A manufacturer of cars has a minimal quality level that any product should satisfy. However, in an administrative process it might be attractive to skip redundant checks to reduce workload.
- *Transportation of electronic data is timeless.* In a network information travels almost at the speed of light.

In short, a workflow process is a typical kind of business process. It can be supported using typical techniques, methods and systems, which will be explained in the following sections.

2.2 Workflow definition

To be able to support workflow processes, a static representation of the process and its resources has to be made. We refer to this static representation as a workflow definition. Such a workflow definition can be divided into two important parts: the process model and the resource classification. The process model contains the various steps and states of the process and the resource classification indicates the kind of work an employee is allowed to perform. Both concepts are explained in section 2.2.1.

Based on the workflow definition the dynamic execution of the process can be controlled. When a particular case comes in, it has to follow a route through the steps of the process before it is finished. In section 2.2.1, the design of the process and resource classification at build-time is clarified. After that, we describe the execution of workflow processes at runtime in section 2.2.2.

2.2.1 Build-time: process model and resource classification

The static representation of the process is designed during the development of the system. One of the techniques to model a business process is the use workflow nets ([5]). (A workflow net is a special kind of Petri net.)

We will give an example of the modelling of a process and a short explanation of this model. Consider for instance the process of sending a bill and receiving the payment at the financial department of a company. First the bill is sent by a member of the administrative staff of the financial department. Then, the payment is awaited. If a certain period, e.g. a month, has elapsed without receiving the payment, a secretary of the financial department will send a reminder. When the payment is received, it is checked by a member of the administrative staff. He or she looks up the accompanying bill and checks the amount paid. When it is OK, the payment can be archived immediately.

When it is not OK, two things can happen. The first one is that there is too little money paid (the amount on the bill was higher than the amount that was paid). Then, a letter with a new bill has to be

sent to the customer by a member of the administrative staff. After that, the process starts again in some way. First they have to wait until a payment is received, eventually by sending some reminders. Then, when the payment is received, it has to be checked, and the loop can be repeated eventually.
It is also possible that the customer paid too much. If that is the case, a member of the administrative staff has to do two things: the remaining amount has to be paid back and a letter with an explanation has to be sent. After that, all things can be archived.

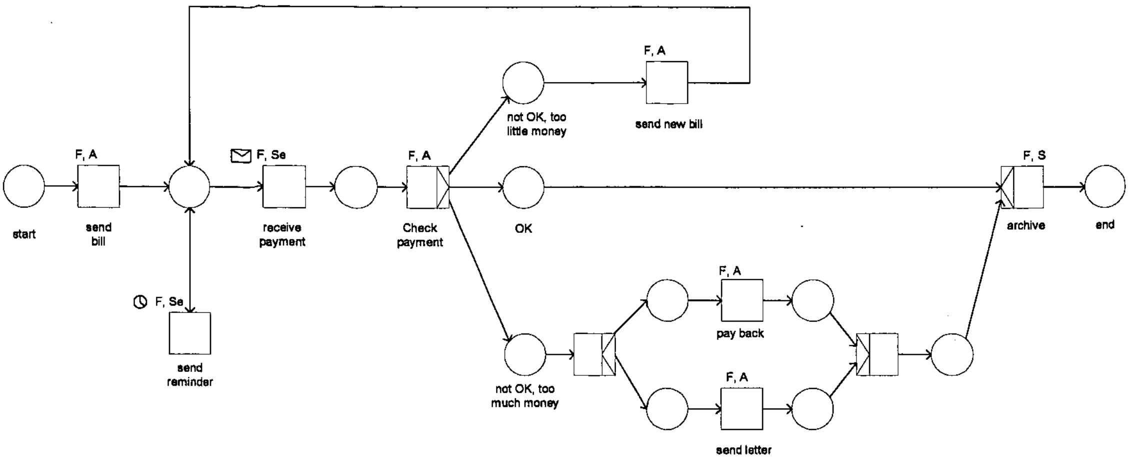


Figure 1: A workflow net for the processing of bills and receipt of payments

The workflow net for this process description can be found in Figure 1. Each activity, that is described in the text above, is represented by a square. The states between activities (also called "places" in Petri net models) are depicted as circles. Above each activity a combination of characters and symbols can be found that refers to the resource requirements and triggers of that activity. For the research of this thesis, triggers are left aside, and resource requirements will be explained below.

We will not extensively elaborate on the theory of Petri nets and workflow nets, and refer to [5] for more information. Nevertheless, some of the typical constructions that can occur in the process model of a business process are considered ([5]):

- **Sequential routing** – We refer to sequential activity execution when a number of activities are performed one after the other (cf. sending a bill and receiving the payment in the example).
- **Parallel routing** – Two or more activities related to a specific case may be carried out in parallel if, by definition, the process contains an AND-split and an AND-join. The AND-split allows more than one task to be initiated at the same time. Upon completion, the parallel workflows are resynchronised using the AND-join. A parallel execution of two activities is possible if they are not dependent on each other, i.e. the one has not to be finished before the other can start (cf. the part of the example after "check payment" was not OK, because there was too much money paid).
- **Conditional routing** – Because most cases need to be able to handle various types of cases, not all cases proceed through a given process in the same way. In other words, there may be various routes through a process. In order to ensure that – dependent upon a case's characteristics – a particular route is chosen, we can make use of the (X)OR-split or the (X)OR-join. For each case, an OR-split selects one or more from a number of alternative activities for each case. It is not determined beforehand which alternatives and how many alternatives are selected. These different routes can be reconverged using an OR-join. When a XOR-join is used to model a certain part of the process, a number of alternatives are present, but only one of them will be selected[5]. Afterwards the alternative routes are brought together by a XOR-join. (cf. the construction after the "check payment" activity. This is a XOR-split, XOR-join: only one of the alternative routes can be chosen. Either the amount of money paid is enough, is too much or too small.)
- **Iterative routing** – An iterative routing is possible within a workflow if its structure permits one or more activities to be performed repeatedly. An iteration may for example, result from a quality

[5] XOR means eXclusive OR.

control: as long as the result of the activity is unsatisfactory, it must be repeated (cf. the iteration when the customer has paid too little, a new bill is sent, and the process starts 'again').

An activity in a workflow process has to be executed by someone or something, this someone or something is called a resource. A resource can be a person in case of a manually or partially automated execution of the activity, or it can be a computer (i.e. an automatically invoked application) in case the activity can be performed automatically.
When the resource to execute an activity is a person, the determination of which resource is allowed to perform an activity is based on a resource classification. The *resource classification* contains the assignment of roles to physical employees. The assignment of roles to employees is dependent on for instance skills, education and authorisation, but it can also depend on the organisational unit or the geographical place an employee belongs to. An employee can have several roles. The classification can contain some hierarchy: for example the head of a department is allowed to do all the work of his or her subordinates (like Mary and John can do the work of Lisa and Jack in Figure 2).

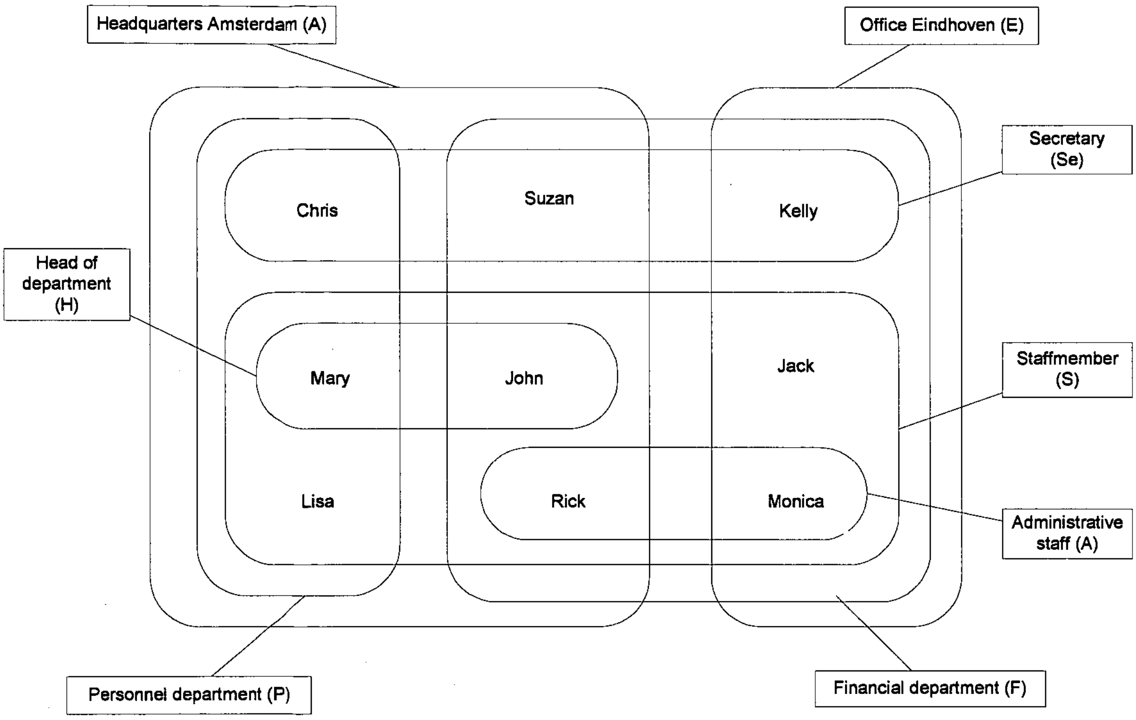


Figure 2: A sample resource classification

When we consider the workflow example and sample resource classification above, we can see that the activity "send bill" has to be executed by someone from the administrative staff at the financial department. (This is indicated by the "F, A" above the activity square). In this case that would be Rick or Monica. There are no requirements on the geographical position, so either of them is allowed to perform the activity. To show the hierarchical part of a resource classification, we take a look at the "archive" activity. It has to be performed by some staff member of the financial department ("F, S" is indicated as resource requirement for this activity). Jack is a staff member, so he is allowed to perform the activity. But so are John, Rick and Monica. They have more competencies than Jack, but they still are allowed to do the work a staff member is doing.

2.2.2 Run-time: cases and work items

As we explained above, in the process model all static information on the activities and resources is depicted. This model is designed at build-time and is difficult to change at runtime. The process model is developed to support the process at runtime, when many cases have to follow the static structure at the same time. Step by step the end product is produced for each case. To describe the cases that are running through the process, some terminology is used that can be rather confusing.

Therefore, we will explain the terms that are used in this thesis below. They are also clarified by a picture (Figure 3).

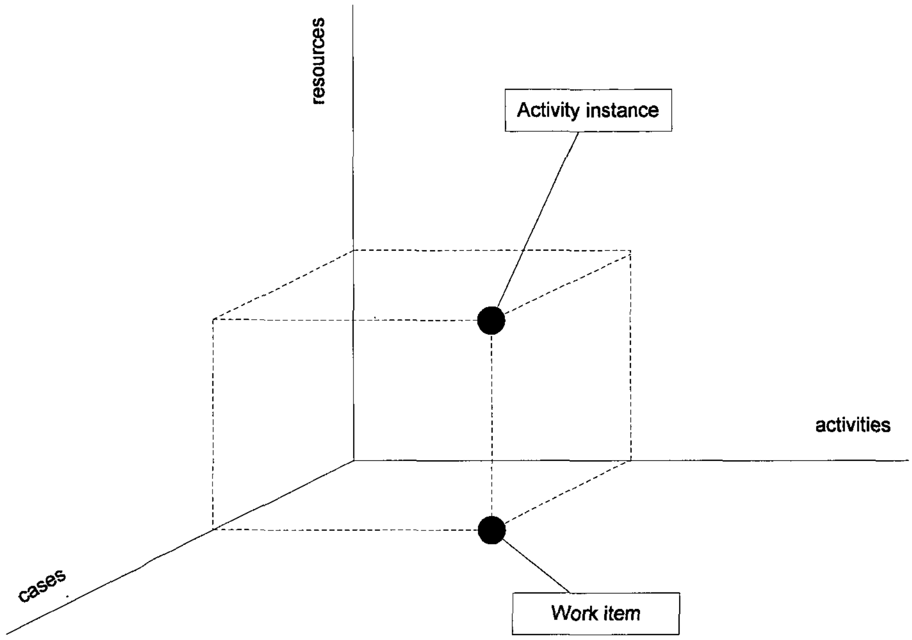


Figure 3: The relationship between cases, activities, and resources (from [1])

A *work item* is a piece of work that has to be performed when a certain activity has to be executed for a particular case. (For clarity of Figure 3: when an activity from the process model is combined with a case, that activity has to be performed for that case. This specific piece of work is called a work item.) Subsequently, an *activity instance* is a work item that is assigned to a resource (by the workflow management system).

For the second part of this research we typically focus on the distribution of work items among the employees (resources) and the assignment of the work items to an employee.

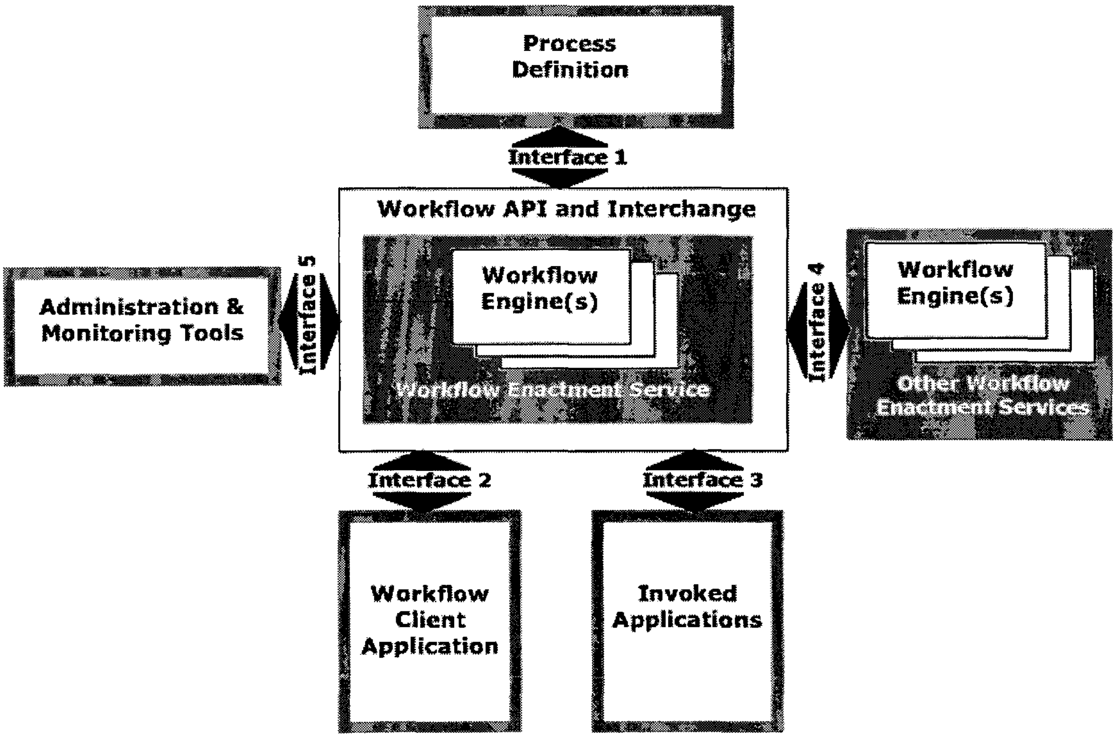

Process Definition
Interface 1
Workflow API and Interchange
Workflow Engine(s)
Workflow Enactment Service
Administration & Monitoring Tools
Interface 5
Interface 4
Workflow Engine(s)
Other Workflow Enactment Services
Interface 2
Interface 3
Workflow Client Application
Invoked Applications

Figure 4: The WFMC Reference Model (from [87])

2.3 WFMC Reference model

The Workflow Management Coalition (WfMC) has developed a workflow management reference model (see Figure 4) in order to create a general framework for the development of workflow management systems. The reference model illustrates the major components and interfaces of a workflow management system. We will have a short look at this model to get more insight in the components and functionality of a workflow management system.

2.3.1 Workflow enactment service

The *workflow enactment service* is the heart of a workflow management system, as is clear from Figure 4. The workflow management coalition ([85]) defines the workflow enactment service as a software service that may consist of one or more workflow engines in order to create, manage and execute a workflow process.

Van der Aalst and Van Hee ([5]) amplify on this definition of the workflow enactment service as follows. This part of the system pumps – as it were – the cases through the organisation. It creates new cases, generates work items based upon the process description, matches resources and work items, supports the performance of activities, and enables the recording of particular aspects of the workflow. In short, this means that the workflow enactment service ensures that the right activities are carried out in the right order and by the right people.

For technical reasons, the workflow enactment service may consist of several *workflow engines*. Such a workflow engine provides those facilities that are required for the logistical completion of cases. Its definition from the WfMC ([85]) is: a workflow engine is a software service or "engine" that provides *the runtime execution environment* for the support of a workflow process. In certain cases, several workflow engines operate alongside one another. Each of them handles a portion of the cases and/or processes.

Therefore, the workflow engines (or together called the workflow enactment service) are the core of the workflow management system.

2.3.2 Interfaces

The workflow enactment service manages the workflow process at runtime. In order to perform the process, the workflow enactment service interfaces with other components ([5], [85]). Note that the WfMC does not consider all of these components as part of the workflow management system.

- **Process definition (tool)** – A workflow engine operates based on one or more workflow definitions. In the workflow reference model, the tools for constructing these workflow definitions are known as process definition tools.
- **Workflow client applications** – Those employees who are only involved in the actual execution of a workflow process will never use the process definition tools. The only contact they have with the workflow management system is through the workflow client applications. Each employee has a worklist, which forms part of the workflow client applications. The workflow engine uses this worklist to show which work items need to be carried out.
- **Invoked applications** – The performing of an activity may result in the starting up of one or more applications. These do not form a part of the workflow management system because they are associated with the actual performance of work, not to its logistical management. Such applications do belong to the workflow system, though, as will be explained in section 2.4. These applications can be fully automatic applications or interactive applications (to which the intervention of an employee is needed, like for instance a word processor or spreadsheet).
- **Other workflow enactment service(s)** – A workflow system may contain several workflow engines. These come under the same management and use the same workflow definitions. Besides, it is also possible to link several autonomous workflow systems with one another.
- **Administration and monitoring tools** – The workflow enactment service ensures the processing of cases based upon workflow definitions. The supervision and operational management of these flows (including the resources) are done using administration and monitoring tools. These can be divided into tools for operational management of the workflows and tools for recording and reporting.

The research in the first part of this thesis focuses on the process definition. The second part mainly deals with the runtime execution environment and particularly with the workflow enactment service, the client applications and administration and monitoring tools.
With this reference model, the technical structure of a workflow management system and its interfaces is clarified. However, we feel this is not enough to support our research on the human aspects. In the next section a solution to this problem is presented.

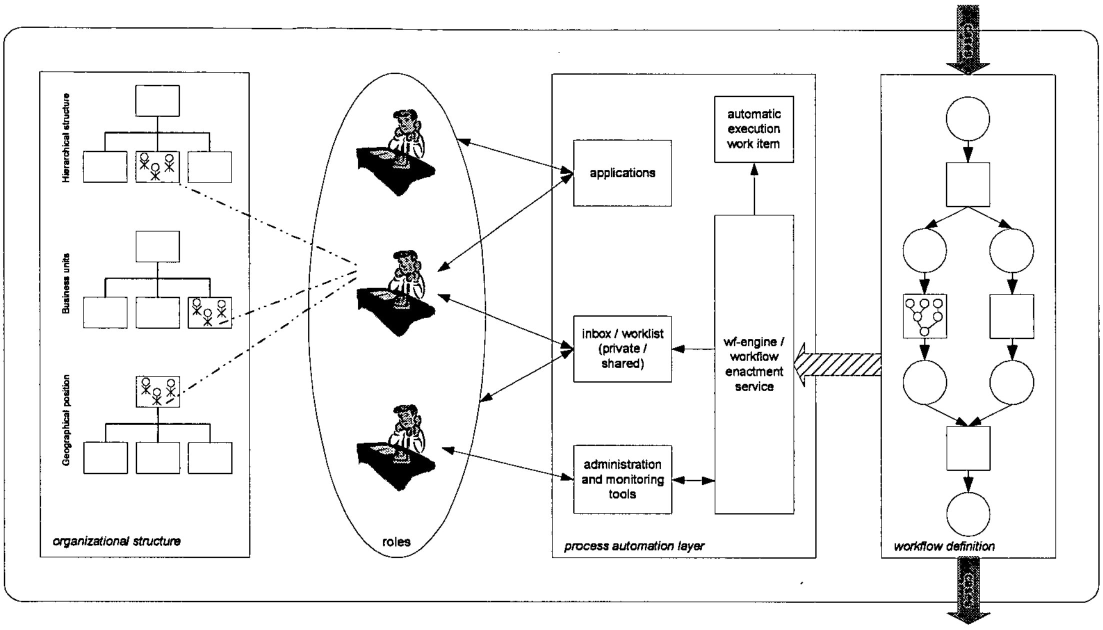


Figure 5: The workflow system

2.4 The workflow system

Now we have examined a basic background on workflow management, our extended view on the workflow system as a whole is presented. Therefore, a new model for the workflow system is introduced and explained in this section.
Note that the term 'workflow system' can cause some confusion because different people use it with different meanings. Therefore, we explicitly remark our view is just one of many, but for the correct understanding of this thesis it is important to know what we mean by the term 'workflow system'[6].

When a workflow management system is implemented in a company or department, the technical parts have to form a unity with the other parts of the organisation concerned in the system. In this research we consider the system within a company as a whole, including the employees and organisational structure for example. This overall system is called the *workflow system*. Inspired by zur Muehlen's "role resolution procedure model" ([54]), we have developed a model of the workflow system, consisting of four levels (see Figure 5). In this section, the structure of this model is clarified. The four levels can be described as follows:

- **Organisational structure** – In this level the structure of the organisation is established. This includes for example, the division in departments or business units, hierarchical structure, functions, physical employees, geographical position, competencies, authorisation and rules.
- **Roles** – The next level are the roles that can be performed by employees in the organisation. To fulfil a certain role an employee has to meet the accompanying requirements (e.g. concerning competences or organisational unit). When an employee is performing a certain role, he or she is authorised to execute the activities that should be executed by someone with that role. This

[6] In the glossary, definitions of the terms that are used in this thesis are collected.

resource-information is stated in the workflow definition. However, in some systems, these roles are not represented and work items are directly assigned to "physical" employees.

- **Process automation layer** – The process automation layer is the third level in the model. It can be divided into two parts: the distribution of work items and the automatic (and computer supported) execution of work items. This includes: the workflow enactment service, the automatic execution of work items (or automatically invoked applications), the computer applications an employee needs to perform an activity, the shared or individual worklists and the administration and monitoring tools. (See also section 2.3)
- **Workflow definition** – The last level is the workflow definition. This is a static representation of the process, as described in a previous section. It consists of the process model, the resource classification and the relationships between those two. Here, you can find which steps have to be executed in which order and who is authorised to perform the execution. Incoming cases follow a certain route (possibly based on conditions). During this course the steps of the process are executed and the workflow enactment service allocates work items to the shared worklists of the authorised roles or to the private worklist of the employee (performing the authorised role). This allocation is done based on the workflow definition. Using this definition, the workflow enactment service can decide to which role or employee a work item should be sent.

In comparison with the workflow reference model of section 2.3, we remark that this model appears separated in two parts in the workflow system model of Figure 5. The process automation layer of the workflow system model takes care of the runtime execution of a process and therefore consists of all parts and interfaces that are described in the reference model, *except* for the process definition tool. The process definition from the workflow reference model is split off because this tool is used at build time. In this thesis, mainly traditional workflow management systems are considered. In these systems, the workflow definition is used as a basis to the execution of the process, but the definition cannot be changed during runtime.

2.4.1 Identification of the parts of the model covered by this research

Our research is based on the workflow system model described above. As explained in the introduction, we focus on two specific research questions that are involved with particular parts of the workflow system.

The first part of this research is only concerned with the last level of the workflow system, the workflow definition, and in particular with the process model. In chapter 3, a method to obtain well-structured process models, based on an information processing view on workflow processes and an operational notion of coupling and cohesion for workflow processes can be found.

The second part mainly focuses on the process automation layer and its interfaces with the users (i.e. the application, inboxes and administration and monitoring tools). In chapter 4, some ideas to "tune" these parts of the workflow system are elaborated, with the intention to make the working in a workflow system pleasant to the employees.

3 Cohesion and coupling metrics for process design

The subject of the first part of this Master's Thesis is an algorithmic approach to the design of a process model as the sub title of this thesis reveals. With this approach, a particular view on the workflow process is adopted. By focusing on the processing of information and considering the content of activities (i.e. their underlying operations) it is possible to support the design of well-structured activities. The design of a process model based on this information processing view is considered as the most favourable grouping of operations into activities. In this chapter we address our first research question as introduced in chapter 1:

How can operations be clustered best into activities?

The processing of information (i.e. the operations on information elements) of the process can be represented by an information element structure, as will be explained in this chapter. Based on these information element structures, a method is developed to define cohesion and coupling metrics to determine the cohesion within the activities of the design and the coupling between the activities. These metrics are inspired by quality metrics in software engineering. The rule of thumb we adopted from this field is that of "strong cohesion, loose coupling". For workflow process design, this means that the operations in one activity should strongly belong together and that there should be as little connections as possible between separate activities.

Next, a heuristic is introduced to support the decision between several workflow designs, based on the cohesion and coupling metrics for workflows. This heuristic helps workflow designers to decide about the size of activities. Using the heuristic, they can choose whether to separate or combine several activities.

This chapter starts with an introduction to the research. Then, we will focus on software engineering metrics and their applicability to workflow processes. Next, our cohesion and coupling metrics for workflow process design will be presented. Their applicability will be tested on two case studies. Finally, we will conclude this chapter with some general remarks and directions for future work.

3.1 Introduction

Administrative business processes are considerably more flexible in their layout than manufacturing processes (see e.g. [63]). Because their focus is on the processing of *information* instead of physical parts, various degrees of freedom exist. For example, making digital copies of reports or documents is simple, which in principle enables concurrent processing. Furthermore, as the transportation of information can take place instantaneously and there are no real limitations with respect to the in-process inventory, activities may be ordered more freely.

While freedom in restructuring administrative processes may be exploited to improve the performance of administrative business processes (see e.g. [3]), it may be difficult to cope with this freedom in other respects. One issue is the proper size of the individual activities in a process (the process granularity). Badly chosen boundaries between activities may affect process performance negatively. For example, small activities increase the number of hand-offs between activities, with a corresponding increase of errors ([73]). On the other hand, activities that are too large may cause inflexibility, since their underlying operations must be performed regardless of their merits under specific circumstances ([3]).

This chapter addresses the issue of activity design in *workflow processes*. The general idea of such a metric is as follows. By focusing on the content of an activity, i.e. its underlying operations, it can be quantitatively expressed how these operations "belong" to each other within one activity or, in other words, how *cohesive* such an activity is. In addition, it is also important to what extent the activities are independent from each other or, conversely, how much they are *coupled*. The inspiration for these metrics comes from software engineering, where an old aphorism is to strive for "strong cohesion, loose coupling". Workflow processes are quite similar to software programs:

- They both focus on *information processing*. Within each step, one or more outputs are produced on the basis of one or more inputs.
- They are alike in their *structure of components*. Workflow processes and software programs have a similar compositional structure (see Table 1). A program – functional or object-oriented – can be split up into respectively modules or classes. Every module consists of a number of statements, and every statement has a number of variables and constants. Likewise, a workflow process has activities. Every activity is built out of a number of elementary operations and each operation uses one or more information elements.
- Their *dynamic execution* follows a static structure. In instantiating either a software program or a workflow process, an execution flow of their elements takes place in accordance with their static representation. This flow may involve consecutive executions, concurrency, conditional routings, etc.

Software programs	**Workflow processes**
Program	Workflow process
Module/Class	Activity/Task
Statement/Method	Operation
Variable/Constant	Information element

Table 1: Resemblances between workflow processes and software programs

Previously, Reijers ([66]), compared the application of a simple workflow quality metric to various design dilemma's with the decisions of 14 experienced workflow designers. The outcomes matched, supporting the validity of the metric and the viability of the underlying idea. However, some limitations to this metric can be identified:

- The cohesion metric from Reijers ([66]) lacked facilities to handle *conditional alternatives* to achieve the same output (XOR-construction), a construct very common in business processes and in administrative processes in particular ([6]).
- The metric did not support the use of *resource information*. Only qualified and authorized employees may execute specific activities and their accompanying operations ([5]).
- Moreover, we think the quantitative notion of cohesion should be balanced with a similar notion of *coupling*, a more or less natural span in software engineering literature ([74],[89]).
- Finally, it was not extensively studied whether *existing metrics* in the software engineering domain could be applied in workflow design.

These four issues are addressed in this chapter, resulting in an extended workflow cohesion metric (containing conditional alternatives and resource information), the addition of a workflow coupling metric, a heuristic based on these metrics to compare several design alternatives, and an overview of software quality metrics as well as an evaluation of their application to workflows.
The structure of this chapter is as follows. First, an overview of existing software quality metrics and their merits is given. Secondly, the domain of workflow processes is explored and our assumptions for the use of cohesion and coupling metrics are presented. Thirdly, the applicability of existing software metrics to the workflow process domain is assessed. Following is the formal introduction of our cohesion and coupling metrics and the assessment of the heuristic, based on these metrics, within a realistic workflow setting and an actual redesign project. The chapter ends with a discussion of the limitations of the presented approach and directions for further research.

3.2 Software quality metrics

In the area of software engineering a variety of software quality metrics has been developed. These metrics seem to be good quality indicators for software design. The purpose of software quality metrics is to get better structured designs. The most important advantages of a well-structured design are, according to [21]:

- **Comprehensibility** – Both the programmer and the user can more easily understand the overall program logic.
- **Manageability** – Managers can more readily assign personnel to modules and responsibility is more localized.
- **Efficiency** – Implementation effort should be reduced.
- **Error reduction** – Testing modules independently should be easier.
- **Reduced maintenance** – Identification of the modules should be easier, since different functions are performed by different modules.

Software quality research used to focus on procedural programs and later turned towards object-oriented software. According to Conte, Dunsmore and Shen [[21]] and Troy and Zweben ([82], and in [75]) the quality of a design is related to five design principles:

- **Coupling** – Coupling is measured by the number of interconnections among modules. It is the measure of the strength of association established by the interconnections from one module of a design to another. The degree of coupling depends on how complicated the connections are and on the type of connections. It is hypothesized that programs with a high coupling will contain more errors than programs not showing high coupling.
- **Cohesion** – Cohesion is a measure of the relationships of the elements within a module. It is also called *module strength.* It is hypothesized that programs with low cohesion will contain more errors than programs not showing low cohesion.
- **Complexity** – A design should be as simple as possible. Design complexity grows as the number of control constructs grows, and as the size in number of modules grows. The hypothesis is that designs with high complexity will contain more errors.
- **Modularity** – The degree of modularisation affects the quality of a design. Over-modularisation is as undesirable as under-modularisation. The hypothesis is that low modularity indicates more errors.
- **Size** – A design that exhibits large modules or a large depth of the structure chart is considered undesirable. It is hypothesized that programs of large size will contain more errors than programs of lesser size.

In literature coupling and cohesion seem to be the most important metrics for software quality, although researchers do not agree on the difference in importance between those two. Troy and Zweben ([82], and in [75]) have found some results of analyses that indicate that coupling is the most influential of the design principles under consideration. Myers ([55]) considers cohesion and coupling as equally important.
The most popular metrics from software engineering, which make cohesion operational, are as follows:

- **Logical relationships** – Inspired by the SMC-cohesion rules of Stevens, Myers en Constantine [81] and Lakhotia [50], Bieman en Kang [15] introduce a number of 'Association-based Cohesion Measures'. They define an input-output dependence graph (IODG) of a program module and six logical relationships among pairs of output elements. To each of these relationships a value is addressed. Based on these values the module cohesion is determined.
- **Functional cohesion** – Bieman en Kang [15] also discuss cohesion metrics based on the computation of data slices for every output variable in a program. The data slices are used to find the number of (super)glue tokens and to compute the value of the cohesion metric.
- **Design level functional cohesion** – The design level functional cohesion metrics, that are also discussed in [15], can be used during the design phase of a program. The program code is not needed. However, the IODG of the program is used to determine the number of isolated and essential components. These numbers again are used to calculate three cohesion metrics.
- **Distance based cohesion** – Simon, Löffler en Lewerentz [76] introduce a distance metric for computing cohesion in object-oriented programs. This metric is based on the theory of similarity and gives a rate of the common elements compared to all elements.
- **Cohesion based on a data flow graph** – Emerson [24] gives a software cohesion metric based on a data flow graph of the program. On the basis of the number of times a certain variable is called, the cohesion of the program is determined.

Although coupling in literature seems quite a promising metric, we have just found a few metrics that determine coupling between modules:

- **Number of data bindings.** – Selby and Basili ([74]) define coupling as the number of data bindings between routines within a cluster and those outside of it. In other words, the number of connections from the module to other modules of the system is counted. This number is combined with the strength of a module to get a coupling/strength ratio for the module.
- **Six types of coupling.** – Fenton and Melton introduce in [25] six types of module coupling, including a ranking on these types from good to bad (or from weak to strong). Every pair of modules in a software system has one of these types of coupling. When a pair has more than one connection, the connection having the strongest coupling counts. In case there is more than one connection between two modules, this metric can be extended by a metric on an ordinal scale to take the number of connections into account. Based on the coupling metrics between two modules an average coupling metric is defined on a system of modules.
- **Object-oriented coupling.** – A large number of object-oriented software quality metrics is available. Among these metrics just a few are focused on coupling ([89]):
  - *Coupling between objects.* This metric counts the number of classes a class is coupled with.
  - *Class coupling.* Class coupling measures connections between classes based on the messages they exchange.
  - *Coupling factor.* The ratio of the maximum possible number of couplings in the system to the actual number of couplings not due to inheritance.

Various researchers have carried out studies to prove that the use of these quality metrics is justified to improve the quality of a software design, but the number of available studies is not overwhelming. Using some small examples, Bieman and Kang ([15], [46], [47]) show how cohesion metrics can be used to restructure a software design. Selby and Basili, [74], give some evidence that low coupling and high strength (cohesion) are desirable. They calculated coupling/strength ratios of a number of routines and found out that routines with low coupling/strength ratios had significantly fewer errors than routines with high coupling/strength ratios. The subject of their study was a new release of an internal software library tool. In [17], Card, Church and Agresti examine a number of Fortran modules from a National Aeronautics and Space Administration project. They found out that fifty percent of high-strength (high cohesion) modules were fault free, whereas only 18 percent of low-strength modules were fault free. There was no relationship observed between fault rate and coupling. Their results indicate that modules with more descendants (modules that call a lot of other modules) have a higher fault rate. Among their conclusions are two important results. High cohesion reduces fault rate. And modules with many descendants are more fault prone than those with few. Here, cohesion seems to be the most significant quality metric, in contrast to the view of Troy and Zweben we mentioned earlier.

3.3 Cohesion and coupling for workflows

Based on the resemblances between software programs and workflow processes that we discussed in the introduction, we hypothesize that the concepts of cohesion and coupling can also have a positive application in the area of workflow process design. Baresi et al. already suggested to apply this hypothesis to workflow process design ([10] and [11]). Although the metrics in software engineering try to give an indication of the overall quality of a program design, we aim for a more modest goal with respect to workflow process design. Clearly, an overall quality concept of a workflow design must encompass elements such as implementability and the effect on customer satisfaction. We will, however, focus on a particular part, i.e. the execution of a workflow process design. For this purpose, we consider "high cohesion, low coupling" as a valuable design adagium. Similar to a well-designed software program, a workflow process that consists of loosely coupled activities will experience fewer errors during runtime, because less communication and exchange of information is needed. The execution of cohesive activities will result in higher quality, because each of them is a coherent part. From an organizational perspective, it should become clearer what it is that needs to be done and who or which department is responsible for it. Several researchers already indicated that intraorganizational dependencies and shared responsibilities should be avoided in workflow (re)design (see for example [13] and [71]). Besides these high-level advantages, we also expect that employees who work within a well-designed process on well designed activities may like their work more and can work more efficiently.

These expectations, obviously, still need validation. A first step is to define a cohesion metric, which is able to handle the elementary characteristics of workflow processes and the addition of the notion of coupling to the area of workflow processes. Before we introduce the cohesion and coupling metrics in the following sections, we will explain the foundations of these metrics and the underlying structure of a workflow process they assume.

3.3.1 An input-output perspective on workflow processes

Underneath the process model of a workflow process design, we distinguish another important structure. Every workflow process takes some input to produce some output. For example, when someone applies for a loan at a bank, he or she has to fill out some forms with personal information, like name and address, but also needs to provide information on income, property, other loans, the new loan amount, etc. Based on this input information the bank-employee decides whether or not this person gets the requested loan, the output. A similar input-output function can be distinguished for each of the smaller steps in the workflow process as well.

During the production of a workflow's output, several smaller steps are executed. The smallest elements that can be identified in such a process are *information elements*. A number of information elements are needed as input to the process. In the activities of the process these information elements are used to produce new information elements and in the end the output is produced in the form of a number of information elements. An *activity* in a workflow process consists of a number of *operations* on information elements. Every operation has one or more input information elements and one or more output information elements. An operation is a basic processing step and cannot have "half-assembly" products, i.e. intermediate information element values. An activity, on the contrary, is built out of one or more operations. Therefore it can have "half-assembly" products. The issue we raised of defining activities can now be reformulated as the proper clustering of information elements and operations into activities. Or from another point of view, it can be said that it is the proper distribution of operations across a workflow's activities.

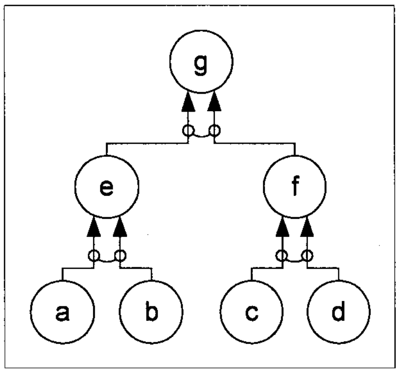


Figure 6: An example activity with four input information elements ("a", "b", "c", and "d"), two semi-manufactured information elements ("e", "f"), one output element ("g") and three operations. To make information element "g" elements "e" and "f" are needed. To get "e" we need information elements "a" and "b" and to get "f" we need "c" and "d".

In Figure 6, a schematic representation of an activity is given, displaying its information elements and operations. The information elements are represented by circles and the operations by one or more arrows. If two arrows are linked to each other, the input elements on the beginning of the arrows are both needed in the operation (this will be explained in more detail in the next section). For example, to get information element "g" we need both "e" and "f". The production of "g" out of "e" and "f" is one operation, where "e" and "f" are the input elements and "g" is the output information element. In order to determine "e", information elements "b" and "c" are needed. And to make "f", information elements "c" and "d" are used. These are exactly the three operations that are contained in this activity. In this structure of an activity, four input information elements can be distinguished ("a", "b", "c", and "d"), one output information element ("g") and two "half-assembly" information elements ("e", "f").

Now suppose, for example, that "a" represents the fixed assets (goodwill, (in)tangible assets) of a company, "b" the current assets (debtors, cash at bank and in hand), "c" the long-term liabilities (loans, mortgages, etc.) and "d" the short-term liabilities (for example creditors). Then based on "a" and "b" the assets ("e") and based on "c" and "d" the liabilities ("f") of the company can be determined. Out of "e" and "f" "g", the balance, is made. In this way we get structures that are rather similar to a Bill Of Material (BOM) [57]. In this context we will call such a structure an *information element structure.*

The information input-output perspective on workflow processes we discussed is similar to that of the methodology of product based workflow design (PBWD) [1], [64], [65]. Also, some contemporary Workflow Management Systems adopt a comparable information-centred modelling and execution approach (see e.g. [4]).

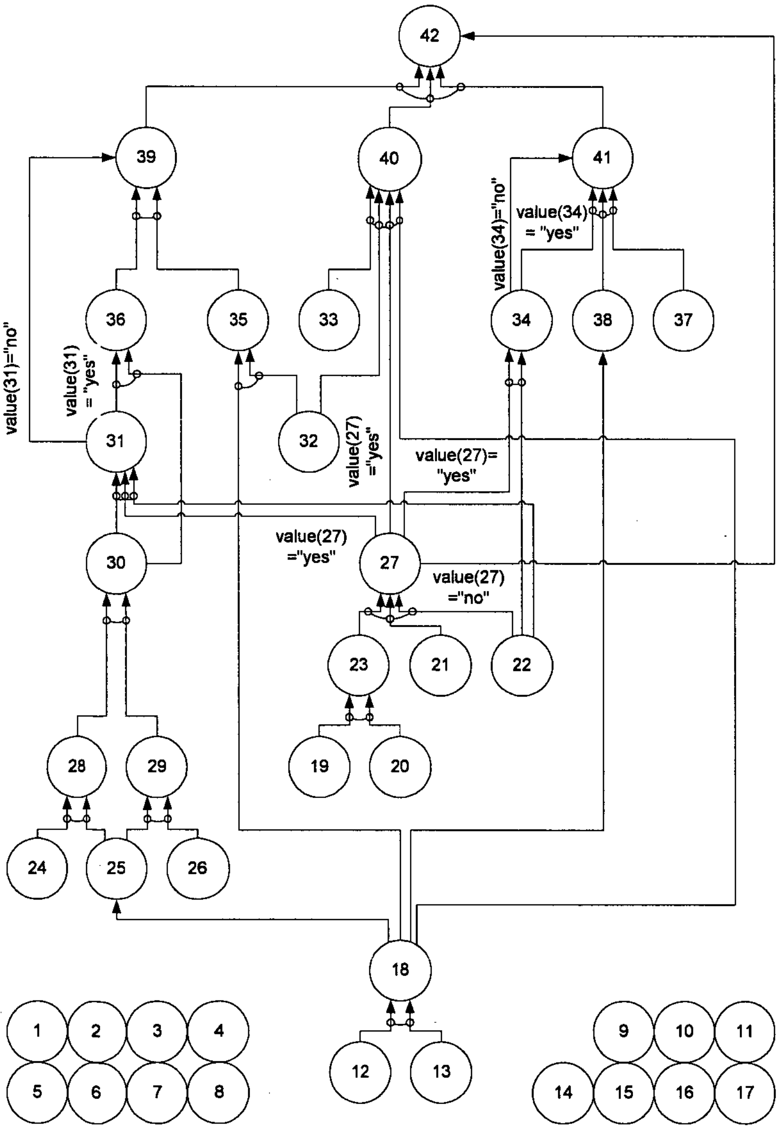


Figure 7: Information elements structure for the process "Request for governmental student grant".

3.3.2 A workflow example

As a running example for the application of our cohesion and coupling metrics and as an illustration for the concepts we introduced, we will present here a workflow process model and its underlying information element structure. Both models deal with the way how requests for governmental student grants are handled in the Netherlands. The presented workflow is a simplified version of the actual procedure as implemented by the Informatie Beheer Groep (IBG) under the authority of the Ministry of Education, Culture and Science.

The essential output of the workflow process is the calculation of the scholarship a new student will get from the government. The amount of money is dependent on the student's background, his parents' income, his living situation, the type of health insurance and the kind of study grant he applied for. The total amount of student grant is composed of three parts. In Figure 7, the complete information element structure can be found. The individual information elements are described in Appendix C.

As the reader can observe in Figure 7, some of the information elements are not integrated in any of the operations, i.e. information elements 1-11 and 14-17. The kind of information they represent is typically required for proper identification, registration, and communication with various parties, but plays no important role in the workflow's information processing. For completeness sake, they have been included in the example.

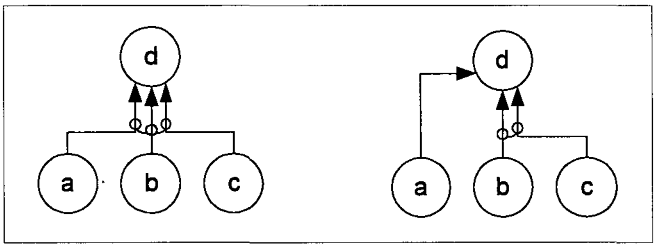


Figure 8: Representation of an information element structure with AND- and XOR-construction. The structure on the left-hand side is the AND-construction: "a", "b" and "c" are needed to get "d". The right-hand structure is the XOR-construction. To make "d" information element "a" is needed or information elements "b" and "c" are needed.

Also note that the presented information element structure incorporates an interesting characteristic, namely a *conditional alternative*. In general, this means that different routes have to be followed to produce the end product based on a specific condition (i.e. the value of an information element). The notion of the conditional alternative is represented in the information element structure by using different notations. In Figure 8 this is clarified. The structure on the left-hand side is an AND-construction. All three input elements are needed to determine the value of the output element of this operation. On the contrary, the structure on the right-hand side is a so-called eXclusive OR-construction (XOR). The output information element can be produced either out of "a" or out of "b" and "c" together.

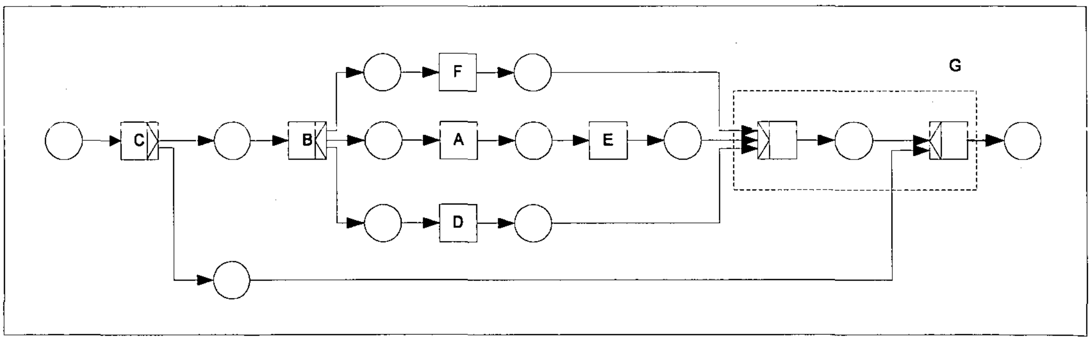


Figure 9: The workflow net of "Request for governmental student grant".

Having explained this notation, it may now become clear from Figure 7 that the value of information element '42' can be determined in two different ways. If the value of information element '27' is "no", information element '42' can be produced directly. When the value of '27' is "yes", a number of other steps have to be executed before the value of information element '42' can be determined.

Activity	Description
A	Determine the income of the parents of the applicant
B	Determine the period/year of student grant and the reference year for tax authorities
C	Determine if the applicant has a right to receive governmental student grant
D	Determine the amount of basic scholarship
E	Determine the amount of supplementary scholarship
F	Determine the amount of loan
G	Determine the total amount of student grant

Table 2: Description of the activities in the workflow net of Figure 9.

In Figure 9, the current workflow process that implements the information structure of Figure 7 is represented as a workflow net ([5]). Individual activities, as described in Table 2, are represented as rectangles. The exact relation between the operations of the information element structure and the workflow process will be explained after we have formally defined both concepts in a later section.

3.4 Mapping software quality metrics on workflows

At first sight, the cohesion and coupling metrics we identified in the software engineering domain may seem to have potential for application to information element structures. After all, relations between variables and modules could be translated into relations between information elements and activities. A closer look, however, reveals several subtle difficulties, as we will discuss in this section. This assessment of the software quality metrics in the workflow domain is based on two pillars. In the first place, we have examined the empirical evidence Reijers gathered from workflow designers in [66]. In this way, we could simulate the outcomes of a software quality metric and compare its outcome with design decisions of experienced workflow designers. In the second place, we examined various realistic processes and two real-life case studies, which used information element structures in deriving workflow process models (see [65]). In this way, we have some confidence on the general characteristics of workflow processes and their underlying information processing structure. The case studies involved the workflow processes of respectively a large social security agency and a multinational bank.
In our study of realistic workflow processes, we identified the following characteristics:

1. Most of the information element structures have only one output information element (i.e. only one end product when the workflow process is finished).
2. Information element structures do not contain cycles.

Considering the first characteristic, we can conclude that the functional and design level functional cohesion metrics and the logical relationships (explained in the section about software quality metrics) are less applicable to information element structures. For example, the design level functional cohesion metrics will have value 1 by definition when there's only one output information element. And logical relationships consider pairs of output elements. The orientation of these metrics is based on the assumption that computer programs usually have more than one output element.
A workaround is that we consider all "half-assembly" products as some kind of output elements, i.e. all intermediate information element values that have to be determined. After all, they can be the output element(s) of an activity (and input of another). Although design level functional cohesion metrics can now be applied on operations from an information elements structure, another problem has to be faced. Bieman and Kang ([15]) do not provide a heuristic that supports the decision of which design is best. Using the design dilemmas we presented to workflow designers earlier and the assumption that strong cohesion within a module or activity is desirable, it turned out that the outcome of this particular cohesion metric did not correlate with decisions actual workflow designers take. Probably this has to do with the fact that our workaround of considering all "half-assembly"

products as output elements leads to much more output elements than is usual for a software program. Moreover, the metrics are basically concentrated on the flow of data. This means the orientation is to the stream of data from input to output and not to all side branches.

The second characteristic we identified, the lack of cyclic structures, has also negative effects on the application of software quality metrics. This cyclic structure in software programs means the update of the value of a variable. It can appear in two forms. In the first place, self-references are quite common in computer programs (like "`sum:=sum+1`"). This can be compared to an operation that changes the value of one of its input information elements. However, we consider information elements as static. They can only be used to make another information element, but they cannot be updated. In other words, a variable in a software program can occur a number of times, while an information element will only appear once in an information element structure. Therefore, information element structures cannot contain this kind of cyclic structure.
Secondly, loops or iterations – often combined with computing the values of arrays – are often applied. An array is a data structure that has no equivalent in information elements. Moreover, in a loop certain computations are executed more than once and most of the time the value of the input elements is different. The loop-construction is comparable to the repeated execution of an operation in terms of workflow design. But in the design view of workflows we do not have a repetition of certain operations. Because of the assumptions that the value of an information element can never change, an operation can only be executed once. (Every other time the operation would be executed, the value of the output element would be exactly the same). Therefore, the execution of a loop in a software program can be compared to the runtime execution of several instances in a workflow process and it cannot be compared to any build-time construction.
There is no good counterpart for either of these cyclic structures in the information processing perspective on workflows. As a result, the functional cohesion metrics and the cohesion metrics based on a data flow graph are not applicable to information element structures.

A third observation is that in information element structures it is allowed to specify alternative ways to produce an output information element (see the example in the previous section). However, software engineering cohesion metrics do not consider the combination of *alternative* statements less cohesive than the combination of *non-alternative* (e.g. sequential) statements. This conflicts with our intuition. After all, some operations and information elements may be part of an activity that is not actually used in some instance.
On this matter, all cohesion metrics seem to fail and cannot be used or have to be adapted, except for the logical relationships. In this latter case the problem with the logical relationships is that there should be a number of different relation types to have good metrics. The idea is that there are different kinds of connections, from very strong to very weak. The metric is based on this difference between the relationships, but information element structures only have three types of relations between elements: no relation, a conditional relation, and a functional relation. Therefore, also this type of metric is not really applicable.

Finally, distance based cohesion is one of the metrics that indicates cohesion in object oriented systems. Other quality metrics in object-oriented software are for example: the number of children of a class and the depth of the (hierarchical) tree. We have found great differences between workflow processes and object oriented programs. An object-oriented program has a hierarchical structure, which cannot be recognized in an information element structure. Moreover, this hierarchy has certain properties, like inheritance, overriding of methods and calling methods from other classes that do not have any equivalence in workflow process design. For example, the calling of a method from another class should be equivalent to using an operation from another activity, which is something that cannot occur in workflow processes. Therefore we did not consider object-oriented metrics.

In short, there are too many dissimilarities to make software cohesion metrics directly applicable to workflow processes. This motivates us to come up with a new cohesion metric, as will be discussed in the next section.

On the other hand, there is a second software quality metric we considered: coupling. On this matter we can say that the previously presented coupling metrics are not applicable in a very straightforward way. Some are too dissimilar, but others have given us plenty inspiration for a new coupling metric.

The six types of coupling are not applicable to our situation, because no different types of connections exist. Some of the coupling metrics from the area of object-oriented software are neither applicable, due to the difference in structure as is explained above. As will be presented in the next section we made a combination of several characteristics of the other software coupling metrics.

3.5 A cohesion and coupling notion for workflow activities

Inspired by the cohesion and coupling metrics in the area of software engineering we developed an extension of the formalization of the information element structure and the cohesion metrics, as presented in Reijers' paper ([66]). This was done by adding the possibility of having several alternative ways of producing an information element and by the addition of resources to the operations and activities of a workflow process. In addition, we also defined a new metric for coupling in workflow processes, a way to combine the use of the cohesion and coupling metrics and a heuristic to determine the best of several designs based on these metrics. The idea of the extension with a coupling metric came from Selby and Basili ([74]) and from Xenos et al ([89]). We also adopted the combined use of a cohesion and coupling metric as in [74].

In the information element view on processes, the job of process design is to impose on a set of operations a number of activities that partition it (in such a way that the rule of thumb of "strong cohesion, weak coupling" is implemented best). Therefore, we introduce the concept of an operations structure.

Definition 1 (Operations structure). An *operations structure* is a tuple (D, W, O) with:

- D: the set of *information elements* that are being processed,
- W: the set of *resource classes* or *roles* that are available to the process. On these resource classes a relation '≤' is defined. 'v ≤ w' means that a person with role 'w' is allowed to do all the work 'v' is allowed to do (and more). For instance, if the head of a financial department is allowed to do the jobs of his accountant and of his secretary (besides his own jobs involved with being the head of the department), it can be formalized as follows: 'accountant ≤ head of financial department', 'secretary ≤ head of financial department'. The '≤'-relation is a partial order because it is transitive and reflexive, but not symmetric (see Appendix D for an explanation of these characteristics).
- $O = \{(p, w, cs) \in D \times W \times P(D)\}$ is a set of *operations* on the information elements, such that there are no 'dangling' information elements and no value of an information element depends on itself:
  - $R = \{(p, w, s) \in D \times W \times D \mid \exists\, (p, w, cs) \in O\colon s \in cs\}$ is connected and acyclic.

where p denotes the output information element of the operation, w the resource class that is allowed to perform the operation, and cs the input elements of the operation.

We have to remark here that the conditional alternative construction as introduced previously is represented by multiple operations with the same output element (as many as there are alternatives). The formalization of the operations of the situations given in Figure 8 is then as follows. (We assume that there is only one resource (*"1"*) to execute the operations.) The construction on the left is built of one operation that is represented by the tuple: $\{(d, 1, \{a, b, c\})\}$. The XOR-construction on the right is built out of two operations that can be formalized by $\{ (d, 1, \{a\}), (d, 1, \{b, c\}) \}$.

The activities that partition an operations structure should satisfy a basic notion of correctness.

Definition 2 (Valid activity). Given an operations structure (D, W, O), we can define a *valid activity* on the operations structure, or simply an *activity*, as a tuple (t, e), where t is any subset of operations, $t \subseteq O$, and e is a resource class, $e \in W$, having the following characteristic:

$$\forall (p,w,cs) \in t : w \leq e \,. \qquad \textbf{(1)}$$

Definition 3 (Valid activity ordering). Given an operations structure (D, W, O), the tuple (T, F) is a valid activity ordering on that operations structure iff:

- T is a set of valid activities, $T = \{ (t,e) \in P(O) \times W \}$, such that:

$$\forall o \in O : \left(\exists (t,e) \in T : o \in t \right). \tag{2}$$

- F is a partial ordering on T, $F \subseteq T \times T$, such that:

$$\forall (t,e),(u,f) \in T : \left(\left(\exists (p,w,cs) \in t, (q,x,ds) \in u : q \in cs \right) \Rightarrow \left((u,f),(t,e) \right) \in F^{*} \right) \tag{3}$$

Within this definition it is expressed by (2) that all operations from the operation structure should appear at least once in an activity. This condition ensures the completeness of the activity design. Condition (3) enforces that when one operation depends on the output of another operation, then the respective tasks they are part of are ordered such that they respect this dependency. In other words, if the execution of an activity requires the value of an information element for one of its operations to be executed, this information element has been established as part of executing a preceding activity. This condition ensures the correctness of the ordering.

A new *cohesion* metric, which allows for the assessment of conditional alternatives in an information element structure, can now be defined as follows. Its first component, the *relation cohesion*, quantifies how much the different operations within one activity are related. It does so by determining for each operation of an activity with how many other operations it overlaps by sharing an input or output, i.e. a non-empty intersection of information elements. In this determination conditional alternatives are treated as separate operations, because in an instantiation only one of the alternatives will be executed. The overlap between these operations (they have the same output element) is therefore not considered. This explains why the expression ($p \neq q$) is in the formula. Then, the average overlap per operation is computed by dividing the total amount of overlaps by the number of operations. Finally, note that all overlaps are counted twice, because we considered all pairs of operations separately (distinguishing as different pairs for example (p, cs), (q, ds) and (q, ds), (p, cs)). Therefore, to get a relative metric between 0 and 1, the average overlap per operation over all operations within an activity is divided by the maximal overlap, i.e. the number of operations minus 1.

Definition 4 (Activity relation cohesion). For a valid activity (t, e) on an operation structure (D,W,O), its relation cohesion $\lambda(t)$ is defined as follows:

$$\lambda(t) = \begin{cases} \dfrac{\sum_{(p,w,cs) \in t} \left| \left\{ (q,x,ds) \in t \mid \left(\left(\{p\} \cup cs \right) \cap \left(\{q\} \cup ds \right) \right) \neq \varnothing \wedge (p \neq q) \right\} \right|}{|t| \cdot \left(|t| - 1 \right)}, \text{ for } |\mathrm{t}| > 1 \\ 0 \text{ , for } |\mathrm{t}| \leq 1 \end{cases} \tag{4}$$

The other component of our cohesion metric, the activity *information cohesion*, focuses on all information elements that are used either as input or output by any operation within the respective activity. It determines how many information elements are used more than once in proportion to all the information elements used. It does so by counting all different information elements that appear in the intersection of a pair of operations, considering all pairs. This number is divided by the total number of information elements in the activity.

Definition 5 (Activity information cohesion). For a valid activity (t, e) on an operation structure (D,W,O), its information cohesion $\mu(t)$ is defined as follows:

$$\mu(t) = \begin{cases} \dfrac{\left| \left\{ d \in D \mid \exists (p,w,cs),(q,x,ds) \in t : d \in \left(\{p\} \cup cs \right) \cap \left(\{q\} \cup ds \right) \wedge (p \neq q) \right\} \right|}{\left| \left\{ d \in D \mid \exists (p,w,cs) \in t : d \in \left(\{p\} \cup cs \right) \right\} \right|}, \text{ for } |\mathrm{t}| > 0 \\ 0, \text{ for } |\mathrm{t}| = 0 \end{cases} \tag{5}$$

The total cohesion of an activity is now given as the product of both the relation and information cohesion. This is to reflect that in our opinion an activity has to score high on both cohesion metrics to say it is cohesive in total. In other words, the operations should be inter-related to each other and information should be shared to a certain degree.

Definition 6 (Activity cohesion). For a valid activity *t* on an operation structure (*D*, *W*, *O*), its cohesion *c(t)* is defined as follows:

$$c(t) = \lambda(t) \cdot \mu(t) \ . \tag{6}$$

The cohesion of the process in total can then be determined by the average activity cohesion.

Definition 7 (Process cohesion). For a process, which consists of a number of valid activities on the operations structure (*D*, *W*, *O*), the average cohesion, or process cohesion (*c*), is defined as follows:

$$c = \frac{\sum\limits_{(t,e)\in T} c(t)}{|T|} \tag{7}$$

As an extension and a natural counterpart of cohesion we also define a metric for *coupling* in a process. Coupling focuses on how much the activities in a process are related, or connected, to each other. A certain activity is connected to another iff they share one or more information elements.

The coupling metric determines the number of related activities for each activity. First the average coupling is determined by adding up the number of connections for all activities and dividing this number by the total number of activities. Now, all pairs of activities have been counted twice. To get a relative metric, the average coupling is divided by the maximal number of coupling, i.e. the number of activities minus 1.

Definition 8 (Process coupling). For a process, which consist of a number of valid activities on the operations structure (*D*, *W*, *O*), the process coupling *k* is defined as follows:

$$k = \begin{cases} \dfrac{\sum\limits_{(t,e),(u,f)\in T} connected\big((t,e),(u,f)\big)}{|T|\cdot(|T|-1)} & \text{, for } |\mathrm{T}|>1 \\ 0 & \text{, for } |\mathrm{T}|\leq 1 \end{cases} \tag{8}$$

where

$$connected\big((t,e),(u,f)\big) = \begin{cases} 1, \text{ if } \big(\exists (p,w,cs)\in t \wedge (q,x,ds)\in u : (\{p\}\cup cs)\cap(\{q\}\cup ds) \neq \varnothing\big) \\ 0, \text{ if t = u} \\ 0, \text{ otherwise} \end{cases} \tag{9}$$

Inspired by the work of Selby and Basili ([74]), as mentioned in the section about software quality metrics, we also define a *coupling/cohesion ratio*. This ratio enables the comparison between various design alternatives.

Definition 9 (Process coupling/cohesion ratio). For a process, which consists of a number of valid activities on an operations structure (*D*, *W*, *O*), the process coupling/cohesion ratio ρ is defined as follows:

$$\rho = \frac{k}{c} \ . \tag{10}$$

The previously defined metrics can be used to find the best workflow design considered among a number of alternative designs. The design with the lowest process coupling/cohesion ratio, is the best

design. Note that we do not describe how the alternative designs can be determined. The ratio can only help to choose the best alternative between already devised options.
To support the calculation of the metrics and the application of the heuristic, we have developed a tool, which will be described in Appendix G.

3.6 Applications

In this section, some applications of our above presented heuristic will be shown. First, the example of the request for student grant as introduced earlier will be used. After that, a recent redesign project is extended with an assessment of the original and new process model using our heuristic and the issues related to this assessment are discussed.

3.6.1 Request for student grant

In this application example, we will focus on the previously presented process of requesting student grant. First, the process cohesion of the partition of activities as given in the workflow net of Figure 9 is computed. Afterwards, we will give two alternative designs for this process, one with smaller activities and one with larger activities. Our heuristic will help to determine which of the three designs is best, implementing our hypothesis that activities should neither be too small nor too large. This is the insight, which we derived from our experiences with workflow designers (see [66]).

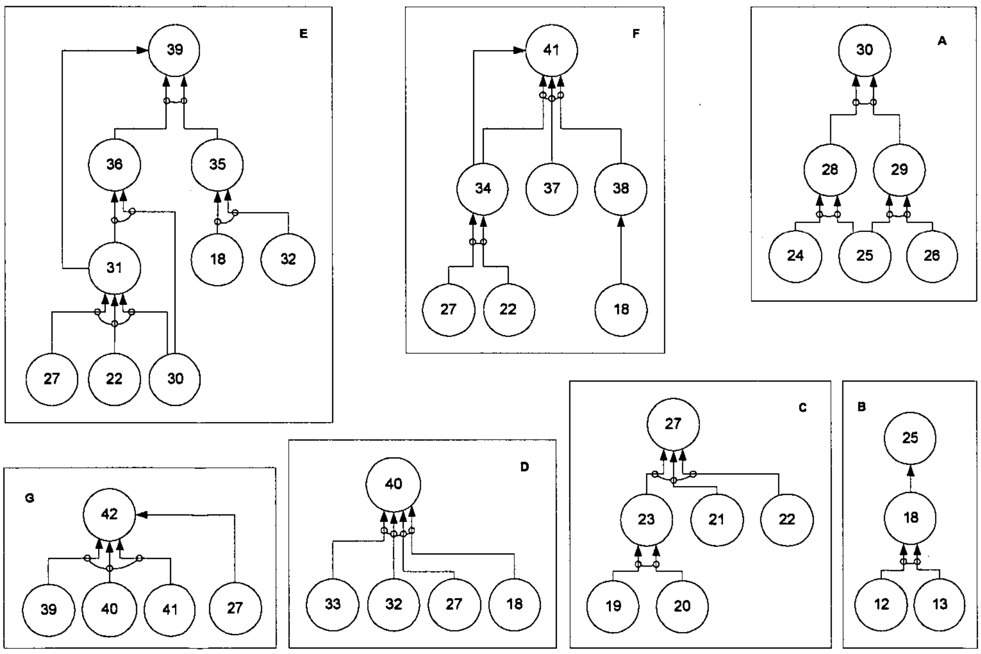


Figure 10: The partitioning of the information element structure in activities for the original "Request for student grant" process.

The original process design

The original process design is a division of the information element structure into 7 activities, as displayed in the process model of Figure 9. The accompanying information element structures of the activities are shown in Figure 10. The formalization of the structure is as follows (we assume there is only one resource ("1") that is allowed to perform all operations, because we preferred to abstract from resource information of this process):

$D_A = \{24, 25, 26, 28, 29, 30\}$
$O_A = \{ (28, 1, \{24, 25\}), (29, 1, \{25, 26\}), (30, 1, \{28, 29\}) \}$
$D_B = \{12, 13, 18, 25\}$
$O_B = \{ (18, 1, \{12, 13\}), (25, 1, \{18\}) \}$
$D_C = \{19, 20, 21, 22, 23, 27\}$

$O_C = \{ (23, 1, \{19, 20\}), (27, 1, \{21, 22, 23\}) \}$
$D_D = \{18, 27, 32, 33, 40\}$
$O_D = \{ (40, 1, \{18, 27, 32, 33\}) \}$
$D_E = \{18, 22, 27, 30, 31, 32, 35, 36, 39\}$
$O_E = \{ (31, 1, \{22, 27, 30\}), (35, 1, \{18, 32\}), (36, 1, \{30, 31\}), (39, 1, \{35, 36\}),$
$(39, 1, \{31\}) \}$
$D_F = \{18, 22, 27, 34, 37, 38, 41\}$
$O_F = \{ (34, 1, \{22, 27\}), (38, 1, \{18\}), (41, 1, \{34, 37, 38\}), (41, 1, \{34\}) \}$
$D_G = \{27, 39, 40, 41, 42\}$
$O_G = \{ (42, 1, \{39, 40, 41\}), (42, 1, \{27\}) \}$

The activity relation cohesion, the activity information cohesion and the activity cohesion are computed for all seven activities (see Table 3). Next the process cohesion is calculated. The process cohesion of the original process design is:

$$c = \frac{0.5 + 0.25 + 0.167 + 0 + 0.222 + 0.143 + 0}{7} \approx 0.183 \ . \qquad \textbf{(11)}$$

As an illustration of the computation of the coupling metric for this process a table (see Table 4) is given, containing all relations between activities (when two activities are connected the value is 1, when they are not connected the value is 0). The coupling value can now be calculated as follows:

$$k = \frac{2 + 4 + 4 + 5 + 6 + 5 + 4}{7 * 6} = \frac{30}{42} \approx 0.714 \ . \qquad \textbf{(12)}$$

Based on these values for cohesion and coupling for the process we can compute the coupling/cohesion ratio:

$$\rho = \frac{0.714}{0.183} \approx 3.9 \ . \qquad \textbf{(13)}$$

Activity	Activity relation cohesion	Activity information cohesion	Activity cohesion
A	1	0.5	0.5
B	1	0.25	0.25
C	1	0.167	0.167
D	0	0	0
E	0.5	0.444	0.222
F	0.5	0.286	0.143
G	0	0	0

Table 3: The activity relation cohesion, the activity information cohesion and the activity cohesion for all activities in the original process design.

	A	B	C	D	E	F	G	
A	0	1	0	0	1	0	0	**2**
B	1	0	0	1	1	1	0	**4**
C	0	0	0	1	1	1	1	**4**
D	0	1	1	0	1	1	1	**5**
E	1	1	1	1	0	1	1	**6**
F	0	1	1	1	1	0	1	**5**
G	0	0	1	1	1	1	0	**4**
	2	**4**	**4**	**5**	**6**	**5**	**4**	**30**

Table 4: The connections between activities from the original process design.

The first alternative process design: smaller activities

In the first alternative design the activities are considerably smaller (see Figure 15, in Appendix E). Activities A and E are split up into four new activities: A1, A2, A3, and A4. The total number of activities in this first alternative design is now nine. In Appendix E (Table 10), the values for the cohesion metrics for every activity can be found for this first alternative. The process cohesion for this first alternative process design is: 0.104. The value for coupling is: 0.611. And the process coupling/cohesion ratio is 5.8.

The second alternative process design: larger activities

In the second, alternative process design we merged activities A and E from the original design together (see Figure 16, Appendix E). This process design has six activities. In Appendix E (Table 11), the values of the cohesion metrics per activity can be found. The process cohesion is 0.123, the process coupling is 0.867 and the coupling/cohesion ratio is 7.0.

Now we know the value for the coupling/cohesion ratio of each process design, we can apply our heuristic. When we compare the coupling/cohesion ratio of the original design to that of the first alternative design we can conclude that the original design is best, because its coupling/cohesion ratio is lower. In the first alternative, various very small activities have been defined that seem to lead to a very fragmented workflow model, which is not something to aim for.

Then, comparing the original design to the second alternative, the heuristic indicates that the original design again is best (lower coupling/cohesion ratio). This appeals to our intuition that the very large task AE is not very attractive, because of its relatively high complexity.

In conclusion, the heuristic points out the original design as the most favourable one, which is confirmed by our intuition. Moreover, it seems to be in line with earlier findings where activities, which were not too large and not too small, were favoured by experienced workflow designers ([66]).

3.6.2 Construction permits

The second application we consider, is an actual (re)design project at a local authority in the Netherlands. For the support of the construction permit process a workflow management system is used, but still some problems exist. Therefore, the process is critically reviewed by workflow designers and a redesign is proposed to overcome the identified problems.

The main purpose of the application of our heuristic to an actual redesign project is to find out whether the same decisions for redesigning activities are made, or in other words, we would like to know if our heuristic supports the decisions workflow designers make in actual redesign projects.

This section elaborates on the construction permits process, its redesign and the application of our heuristic. First, a short explanation of the content of the process is given, the redesign elements that the workflow designers proposed are clarified, the applicability of our heuristic to this specific situation is discussed and our conclusions and research questions are presented.

The process

The process of construction permits deals with the request for permission to build a certain object at a certain place, situated within the area managed by the local authority. Before a construction permit can be issued, a number of things have to be checked. For instance, external parties have to give an advice, it has to be checked legally by a jurist and the building inspector employed to enforce the regulations regarding the external appearance of buildings has to approve the draft for the building. The main process consists of seven activities, respectively "incoming documents", "receipt of documents", "advices", "susceptibility", "judgement", and "check building". Each of these activities consists of a sub process. The process models can be found in Appendix F.

The workflow designers provided detailed information on the information processing in the construction permits process. For every activity on the sub process level they indicated the information elements that are going in and the information elements that come out. Based on this information, an information element structure is constructed. See Figure 17, in Appendix F, for a graphical representation of this structure made with our tool[7]. The meaning of the information element numbers can also be found in Appendix F. However, the workflow designers often did not provide us with detailed information about how to produce the output using the input of an activity. Therefore, we assumed that such an activity consists of only one operation. For example, activity "Beoordelen of aan indieningseisen is voldaan" ("Assessment on submission requirements") takes as input the file and produces a file containing the decision whether a file is susceptible or not. How this decision is made is not clarified, but we think this could be quite complex. Still we consider it as a single operation, without half-assembly products, because we do not have information about intermediate steps.

Redesign and assessment
The workflow designers proposed some redesign elements to solve existing problems in the process. They mainly came up with more time control in the process: the WfMS should better control and handle time-outs. By this they mean for example the WfMS should react when one of the advices is not received in time, etcetera. To overcome the problem of time control the workflow designers made a redesign of the process (the process model can be found in Appendix F). They introduced time control activities, but did not redesign the other activities of the process.
After a closer examination we have to conclude that they made no substantial redesign with respect to the size of activities (process granularity). We expected them to come up with some proposals to split or combine activities on the lowest level (based on for example task elimination, task addition and task composition, from [65]), but they did not. The redesign changes they proposed cannot be assessed by our heuristic, because the partition of the information element structure into activities will be the same (resulting in the same values for the metrics and an indecisive heuristic). However, by the examination of the applicability of our method to this actual project we have found some points of interest for further research, which will be presented in the next section.

Remarks
As explained in the previous sections, we would like to apply our heuristic to the original process design and the redesign of an actual workflow project. Unfortunately this turned out not to be possible within the construction permits project. However, we have found some interesting remarks by trying to apply our method of PBWD and the heuristic to this case.
In the following we will elaborate on these issues. First, the problems we experienced in the application this process will be explained and afterwards some general notes or research questions are formulated.

- The workflow designers have developed a process model for the main process that can be divided into several sub processes. These sub processes are modelled on a more detailed level. On the main level, the activities consist of operations that have to be executed by different roles, which means the activity is not 'resource correct'[8]. On the other hand, considering the detailed sub models (that can be linked together to form a more detailed process model), almost every activity contains just a single operation. This means that the cohesion for most activities will be 0, which, eventually, makes alternative designs difficult to compare.
  Based on this experience we feel there should be a level in between to examine the process model or the way to produce the output of an activity, based on the input, should be better described.

  In general we can say the choice for the <u>level of detail</u> in modelling a process is something, which can cause difficulties to our method. We cannot handle too highly modelled activities, because they are not resource correct, nor too small activities, because they consist of only one operation. Modelling tools often provide a lot of freedom in the level of modelling and in the level of detail of a model. Our method is not able to deal with this in a structured way. We need very detailed information about the process (on information element level), but we cannot handle a big amount of very small activities (i.e. activities that contain only a single operation).

[7] The description of the tool can be found in Appendix G.

[8] Resource correct means that the resource that has to perform the activity is allowed to perform all underlying operations of that activity. (i.e. the activity resource is equal to or on a higher level than the resources that should perform the operations.)

- Next we can remark that it seems to be very difficult to retrieve all the information elements and their mutual relationships that are needed to make a decision and to distinguish which elements are used in which part of making the decision. Companies often work with customer files, as can be seen in this case study. When all possible information to request a construction permit is received, a file ("dossier") is created. The remaining parts of the process all operate on this file, which means they use the information they need from the file. The content of the file is not described in detail. Thus it is hard to find out exactly which information elements are used to make a particular decision or to execute a particular operation or activity in the process. This makes it not very feasible to construct a good information element structure.
  In this particular case, we solved this problem by considering the file as one information element. But when adopting this view on a file as information element we created cycles in the information element structure for every time the file was updated in an operation or activity. We solved this problem by considering the updated output file as a new information element, because circular references are not allowed in the information element structure. This contrasts with people's intuition of a file as one information element.

  In general companies do not consider the information elements of the process on the detailed level we do. They often use customer files, to keep all the information concerning a customer together and they use the file to execute the process. Usually, the content of a file and the relationship between the parts of this file are not described in detail and it therefore is not possible to determine which specific information elements from the file are needed to an operation.

- The workflow designers have paid much attention to the exceptions in the process, i.e. what should happen when documents arrive too late, when advices are not in time, when not all required documents are present, when an activity is wrongly performed, etc. These exceptions cannot be considered in the information element structure and its partition in activities, because information element structures describe the structure of the product, not the flow of the process. In this case, time control information is not a part of the end product.
  Another exception can be the negative result of a check. This is neither considered in an information element structure. The negative result of a check will lead to the re-execution of one or more operations, which means an update of an information element. In this thesis, information elements are considered as static elements that cannot be updated.
  This means using the information processing view it is only possible to develop or design the main structure of the process, but it is not possible to handle exceptions and time control, based on this information structure.

  In general we can say that by the use of the information processing view we can only determine the main structure of the process. The focus is on the product and not on the flow of the process. The handling of exceptions and time control is not included.

In this section, we tried to assess the actual redesign of a process model using our information element structure and heuristic based on cohesion and coupling metrics. Unfortunately, this turned out to be not possible, because of the kind of redesign the workflow designers proposed. However, by the assessment we identified some issues for further research. Therefore, we feel this application nevertheless has a positive contribution to our research.

3.7 Conclusion / Future work

In this chapter, we discussed the applicability and development of cohesion and coupling metrics in workflow process design. We extended the work of Reijers ([66]) and defined cohesion and coupling metrics for the design of activities in a workflow design, based on an information processing perspective on workflow processes. Although existing software quality metrics inspired us, none of these seemed directly applicable to the workflow process domain. Based on our newly introduced coupling and cohesion metrics, we also defined a ratio that allows for a comparison between alternative designs to decide which one is best.

With the applications described in the previous section we have shown the advantages and possibilities of applying these metrics to practical situations and process (re)design. However, still some limitations and remarks for further research exist. This can be especially clear from the second example of the assessment of our heuristic to an actual redesign project.

Obviously, the *information processing perspective* we took is but one of the views on workflow design. As we stated before, these metrics can make a positive contribution to the quality of a workflow process design concerning the content of the process, but, they are not the only aspects that determine the success and quality of such a design. Clearly, an overall quality concept of a workflow design must encompass elements such as implementability, its effect on customer satisfaction, the acceptance of the users and the degree to which business goals of the process are met. However, we still believe that a well structured design (as can be obtained by using cohesion and coupling metrics) will decrease problems with respect to the subjects mentioned above. It therefore can be used as one of the ways to develop better designs.

Another important limitation of the metric in isolation is the *performance* aspect of the operations. We abstracted, for example, from timing information on the various operations, although this may be essential information to come up with a well-structured workflow design. When certain operations take much time compared to other activities, it is not sensible to put them together in one. We feel that this is one of the most pressing issues for further research in extending our – as of yet – purely logic view on the operations. Besides that, the timing aspect or duration of an operation provides another difference between software programs and workflow processes with respect to the runtime execution.

From working on the case studies we experienced that the *level of detail* in the design can be an issue too. For the PBWD method to be effective, resulting in an information element structure, a great level of detail is needed to describe the process. When these details are not stated in any rules, laws or procedures, it is very difficult to find out the exact operations and the level of detail of the process. We will clarify this with the example of the building permits process of section 3.6.2. At first the only information that was available, was that the fire department, the environmental department and the department of culture and civil engineering had to give an advice on granting the permit. A description of the steps of the process was available, but in order to apply PBWD we needed to know what exactly happened in activities with respect to the processing of information. In the next state of the redesign process, much more detailed information was available, but even then the workflow designer considered the customer file as an atomic information element. Besides the difficulties in achieving the proper information to produce an information element structure, this can also be a time consuming and difficult process when the content of activities is not explicitly formulated, by for example law or working instructions.

Moreover, the presented heuristic seems to be only applicable to workflow processes in which a decision is made based on a lot of information and rules to process this information. This can limit the number and nature of processes that can be designed using this method.

Finally, the developed method has not yet a strategy to handle *sub processes*. For the applications in the previous section, we assumed the process consisted of one level. But bigger processes can be split up into smaller parts with a "main track" to control the flow through the process. In this we think a high coupling between the parts or sub processes is worse than between activities of one sub process. But our metrics cannot handle this situation yet.

Many possibilities to extend this research exist. On a higher level, we aim for the *testing* of our hypothesis on the quality of workflow designs. Up till now, we have only obtained the experts' view on some small design dilemma's ([66]) and examined two processes (one having a medium or low complexity of information processing and the other one was redesigned in a different way than suitable to our approach). The results of this research seem to be very promising, but we think more practical validation is needed. This necessarily will require the further involvement of actual workflow models, the knowledge of workflow designers, and end-users.

Finally, the cohesion and coupling metrics and the heuristic can only support the workflow designer in making decisions with respect to activity definition. The heuristic does not suggest any *clustering or ordering of information elements* itself. An extension of the heuristic so that it efficiently generates (semi-)optimal activity definitions itself will be an ultimate and challenging next step in this domain.

4 A human oriented approach to the "tuning" of workflow systems

As the sub title of this Master's Thesis states, the second part of this research elaborates on a human oriented approach to workflow system (re)design. In this chapter, we will elaborate on our last research question, as introduced in Chapter 1:

"In what way can a workflow system be adapted to improve an employee's experience of the work he or she is doing within the system?"

We will take a human centric view on the "tuning" of a workflow system. This means we will not determine the content of a job, but we will consider the means and methods to perform a job.

As explained in the introduction, we will not focus on the granularity of an activity from a human perspective. Although this view would perfectly fit with the previous chapter, we think this is a too detailed approach, because the job an employee has to perform usually comprises of more than one activity. Instead, we will propose some ideas to adapt the workflow system in an easy way in order to improve an employee's perception of the work he or she is doing, without changing the content of the work (i.e. the input and output of an activity).

First we will give an introduction to this area and the reason of this research. Next, related literature is discussed and our ideas are presented. Finally, the ideas will be validated by some experts and the possibilities of supporting these ideas with current workflow technology will be examined.

4.1 Introduction

After discussing the theoretical and technical side of workflow process design in the previous chapter, it seems justified to explore the human side of designing workflow systems and business processes. After all, people have to work with and in the developed system. When they feel well, they probably will work harder, be more loyal and complain less, which can result in a better performance. According to literature on human needs and motivation (see for example [8] and [70]), an employee's well-being, satisfaction and motivation is dependent on many things[9], like promotional opportunities, opportunities for personal growth, recognition, responsibility, and achievement.

In general, employees experience their work based on their perception and their values. Such experiences lead to job attitudes (i.e. evaluative statements or judgements concerning the work), such as job satisfaction and well-being[10]. These attitudes affect an employee's motivation, which can be described as the processes that account for an individual's intensity, direction, and persistence of effort toward attaining a goal ([70])[11]. The behaviour and especially the performance of an employee are influenced by his or her motivation.

Personality, well-being, job satisfaction and motivation of an employee are correlated to job performance ([43], [44], [45]). Even though the correlation between job satisfaction and performance is low[12] ([39], [43], [72]), it is generally believed that a users' satisfaction and motivation is a very important part in making automation systems work ([72]). We also adopt this assumption in our view on jobs within workflow systems.

[9] Note that these things are not all influenced by the automation system

[10] Another component of a person's well-being is his or her health.

[11] Motivation is more than an attitude. It also encompasses a feeling of having energy to do something, rather than being just an evaluative statement.

[12] Based on [62], a more accurate conclusion is actually the reverse: productive workers are likely to be happy (satisfied) workers.

Nowadays, more and more companies automate their administrative business processes using workflow management systems. These systems can realize serious improvement in customer satisfaction and key performance indicators of the process (like throughput time, waiting time, occupation rate), but they can also affect the well-being and satisfaction of involved employees ([23], [42]). We are not sure about the positive effects of computer and information systems on an employee's experience of the work he or she is doing, and related to that, the impact on the motivation and performance of the employee.
Similar to the changes caused by mass production and working on the assembly line in the 19th and 20th century[13], a danger of (workflow) automation can be the erosion of tasks, resulting in monotonous and boring work ([49]). In this chapter, we will develop some directions for how to improve the work in a workflow system and to make working in a workflow system more enjoyable for people, without proposing changes that can greatly affect the organisation, business process, or content of the work. In some respects, this can be seen as the "tuning" of the workflow system to improve a user's experience of the work in the system. We will recommend small and easy adaptations to the total workflow system, based on a combination of two relevant models from job design theory and workflow management. Our extended view of a workflow system as introduced earlier, in section 2.4, will be used.

The structure of this chapter is as follows. First we will give an overview of job design theory and related literature and we will explain a well-known model from this field: the Job Characteristics Model (JCM) of Hackman and Oldham ([28], [29]). This model is based on psychology, and gives an idea on what people like in executing their work. It consists of five job characteristics that will be elaborated in the next section.
Next, we will present zur Muehlen's assignment and synchronisation policies. This model is based on the technical aspects of workflow management systems. As clarified in the introduction, a workflow management system operates based on a business process model that specifies the static structure of the process. Following this static structure, cases are led through the process and through the organisation. Using applications, interfaces and other parts of the system, employees execute their activities to produce the outcome of a case. The static structure of the process model does not determine the total design of the system, but still leaves some freedom in the system for the design of the interfaces and the assignment of work items to employees, for instance. The "tuning" of a workflow system can be obtained using this freedom. Some parts of the system that can be adapted are described by zur Muehlen ([54]) and will be discussed in more detail. We will mainly focus on the distribution and assignment of work items.
In the third section, the two previously described models are combined and we elaborate on the impact of the options for assignment and synchronisation policies to the job characteristics. After that, our own ideas to "tune" a workflow system that are inspired by this combination, are presented. They are linked to the workflow system model presented in the introduction. In section 5, the generated ideas will be validated by a qualitative expert validation. Next, we will identify to which degree present technology, in the form of three actual workflow management systems, is able to facilitate and support the best of the ideas. Finally, this chapter concludes with some general remarks and directions for future work.

4.2 Job design theory

This section will give a short overview[14] of literature on job design theory, to show the many views on job design. According to Holman et al ([37]), job design is an important element in any appreciation of how organisations and the people in them behave. By job design they mean "the content of the job that an individual or group undertakes (for example, the tasks and roles they fulfil) and the methods they use to undertake their work.[15]

[13] Some well-known concepts in the area of mass production and assembly line are Taylorism and Fordism.

[14] Mainly based on [37].

[15] Note that in the research of this chapter we will not focus on the content of the work, but on the methods, as Holman et al call it.

In job design, a lot of research is done to discover the problems in introducing new technical systems and to make the implementation of an information system successful, resulting in user-centred design programs, user design participation and general design principles[16]. Most of these approaches focus on the process of designing a new system and have a core goal of creating employee support for the new system and the accompanying changes to organisational structure, work methods, etcetera, within every part of the organization.

We think the focus should not only be on the process of designing but also on the resulting design itself. Clearly, we can not guarantee that a well-designed technical system will work perfectly, because the social system, health, character traits and perceiving of a person among other factors will influence a person's well-being, satisfaction, motivation and performance too. We believe a user-friendly design of the technical system, taking into account the social system it is connected with, indeed can contribute to the success of information systems, particularly, by improving an employee's experience of the work (s)he performs.

Before elaborating on our own ideas, we will first give a small overview of literature concerning the psychological and socio-technical aspects of job design in technical systems.

Holman, Clegg and Waterson ([37]) state that at present the field of job design in a psychological context is rather fragmented and therefore they present a general overview of the different perspectives on job design. They divide the research done into three different approaches, or paradigms:

- **Functionalism** – Functionalism is characterized by an objective approach to social science and a sociology of regulation[17] (i.e. an emphasis on the status quo, social order, consensus, integration and cohesion). As such, functionalism is concerned with rational explanations of the prevailing social order and its stability and maintenance. Burell and Morgan ([16]) state that functionalism seeks to provide essentially rational explanations of social affairs. It is a perspective, which is highly pragmatic in orientation, concerned to understand society in a way, which generates knowledge, which can be put to use.
- **Interpretivism** – Like functionalism, interpretivism is characterized by an emphasis on the "sociology of regulation", although the focus is on subjective experience and how individuals create and maintain the world though their own actions and those of others. Interpretivism seeks explanation within the realm of individual consciousness and subjectivity, within the frame of reference of the participant as opposed to the observer of action ([16]).
- **The critical paradigm** – The critical paradigm emphasizes change, conflict, domination and emancipation, and includes radical structuralism[18] and radical humanism[19]. In particular labour process theory has contributed to job design by its attention to the nature and use of management practices that attempt to exert control over the labour process, whether or not such practices are oppressive and alienating, and the manner in which such practices have been consented to, or resisted by, non-managerial employees.

Here, we will only focus on the functionalist approach to job design, because it is an objective approach and we want to conform to the functional approach of the design metrics and heuristic we presented earlier.

Historically, the main focus of the functionalist job design paradigm has been on the psychological consequences of work simplification brought about through the pervasive adoption of Tayloristic

[16] See for example Chern's 'principles for socio-technical design' ([18]), which are famous in this area. Other references to relevant literature are: [9], [19], [20], [27], [32], [33], [34], [56], [78], and [84].

[17] Burrell and Morgan ([16]) introduce the term '*sociology of regulation*' to refer to the writings of theorists who are primarily concerned to provide explanations of society in terms which emphasise its underlying unity and cohesiveness. It is a sociology, which is essentially concerned with the need for regulation in human affairs. The opposite of '*sociology of regulation*' is '*sociology of radical change*'.

[18] Radical structuralism is defined by its concern to develop a '*sociology of radical change*' from an objectivist standpoint.

[19] Radical humanism is defined by its concern to develop a '*sociology of radical change*' from an subjectivist standpoint.

approaches to work organization and job design. (Holman, Clegg and Waterson, [37]). Three approaches of functionalism can be distinguished in literature:

- **Socio-technical theory** – Socio-technical theory is concerned with the design of work systems and posits that these comprise a technical system and a social system. These two sub-systems are seen as interdependent and should, therefore, be jointly designed in such a way that the overall system is optimal. Socio-technical theory has largely focused on redesigning social systems around a "given" technology. (Holman, Clegg and Waterson [37]).
  Note that we would like to examine the possibilities of adapting the technical system to the requirements of the social system.
- **Job characteristics theory** – The second main theoretical approach also focuses on the characteristics of jobs and has been strongly influenced by the ideas of Hackman an Oldham ([28], [29]). They proposed five core job dimensions as ways of describing and understanding job design. Collectively, these job characteristics are seen to predict important work outcomes such as motivation, and as a consequence, performance, job satisfaction, labour turnover and absence. (Holman, Clegg and Waterson [37]).
- **Social information processing model** –The social information processing model states employees adopt attitudes and behaviours in response to the social cues provided by others with whom they have contact. The fact that people respond to their jobs as they perceive them rather than to the objective jobs themselves is the central thesis in this third functionalist approach. (Robbins [70]).

According to Holman et al ([37]), the first two approaches, socio-technical and job characteristic theories have been the most influential. One of these approaches will be used for further research on the human oriented "tuning" of workflow systems. We have chosen for the Job Characteristics theory, because it seems to be the most concrete and applicable theory. Below, the Job Characteristics Model of Hackman and Oldham is further explained.

4.2.1 The Job Characteristics Model (JCM)

Based on the theory of human needs (see for example [8]) and Turner and Lawrence's Requisite Task Attributes Theory ([83]), Hackman and Oldham developed the Job Characteristics Model (JCM), ([28], [29]). Today this model is known as the dominant framework for defining task characteristics and understanding their relationship to employee motivation, performance and satisfaction. According to this theory a job can be characterised in terms of five core job dimensions ([30], [31], [70]):

- **Skill variety** – the degree to which the job requires a variety of different activities so the worker can use a number of different skills and talent.
- **Task identity** – the degree to which the job requires completion of a whole and identifiable piece of work.
- **Task significance** – the degree to which the job has a substantial impact on the lives or work of other people.
- **Autonomy** – the degree to which the job provides substantial freedom, independence, and discretion to the individual in scheduling the work and in determining the procedures to be used in carrying it out.
- **Feedback** – the degree to which carrying out the work activities required by the job results in the individual obtaining direct and clear information about the effectiveness of his or her performance.

The higher a job scores on each of these characteristics, the better the job is, and the higher the motivation, performance and satisfaction of the person executing this job will be. ([70]).
This model seems to have proven its validity, because it is used in many kinds of research on the quality of jobs (see for example [58]). Moreover, [26] shows that there is quite a strong correlation between the job characteristics and job satisfaction and a small correlation between the job characteristics and performance, which justifies the use of the Job Characteristics Model to improve an employee's work experience.

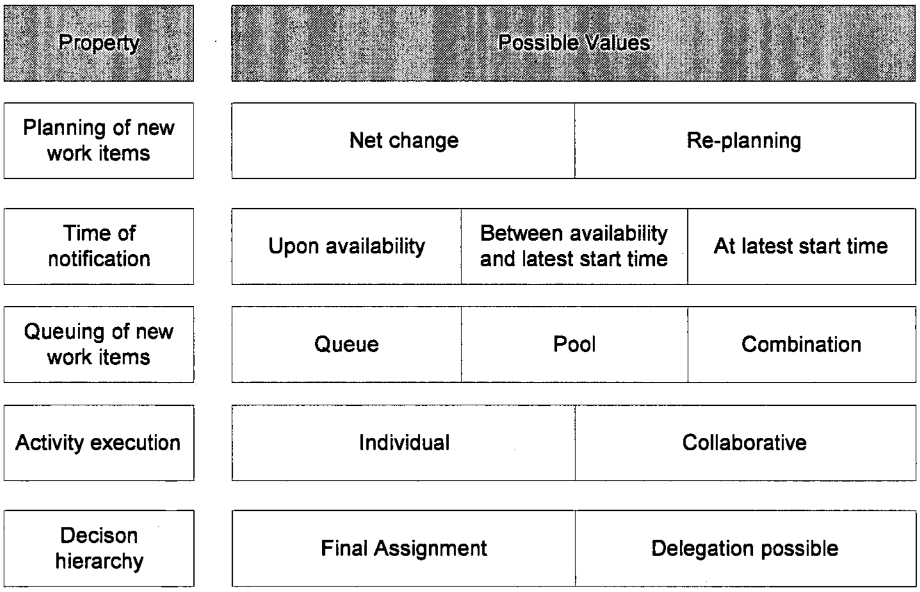
In this context, we would like to apply the Job Characteristics Model to explore the human side of working within automation systems. We will work out some possibilities to make small adaptations to the process automation layer of a workflow system (as will be explained in a later section) in order to reach a situation with a more employee-friendly working environment and system. As a guideline, we will use the five job characteristics as some kind of dimensions to the human oriented adaptation of a workflow system. We will try to "tune" the workflow system on these dimensions by changing the (technical) characteristics of the system and creating better conditions under which the same content of the work can be performed more pleasantly. But before elaborating on this, the parts of a workflow system that can be "tuned" are clarified in section 4.3.

4.3 Assignment and synchronisation policies in WfM-systems

In [54], zur Muehlen describes the "tuning" of the process automation part of the workflow system by elaborating on assignment and synchronisation policies. "Assignment and synchronisation policies determine in which way the work items from a case are distributed and assigned to employees." In this section, the most important parts of zur Muehlen's model will be discussed. The below explanations are straight from his paper ([54]).

4.3.1 Assignment policies

The first part, on assignment policies, covers the distribution of work among qualified employees. "For the assignment of pending activities, different strategies can be implemented. These strategies have an impact on how the workflow enactment service prioritises activities and notifies candidate performers" (see Figure 11).



Property	Possible Values		
Planning of new work items	Net change	Re-planning	
Time of notification	Upon availability	Between availability and latest start time	At latest start time
Queuing of new work items	Queue	Pool	Combination
Activity execution	Individual	Collaborative	
Decison hierarchy	Final Assignment	Delegation possible	

Figure 11: Assignment policies (from: [54])

Below, each row of Figure 11 is explained. To be more clear, an assignment property can be seen as an axis on which the value can be varied over the possible values as indicated in the second column of the figure (see for a more detailed explanation Appendix H).

"The planning of new work items describes the behaviour of the workflow enactment service, when new activities become executable. A net change strategy would only determine the assignment for the new work item, while a re-planning strategy would re-allocate all work items that have not yet been started, possibly removing work items from some performer's worklists and placing them on other worklists.

The workflow enactment service can notify performers about pending activities either upon the availability of these activities, at the latest start time of the activity, or at an arbitrary time between these two points. While human performers are typically informed upon availability of an activity, it may be desirable to delay the assignment of activities to technical resources, if the workflow management system is capable of optimising the activity schedule based on the economic value or the priority of the activities. [...] "

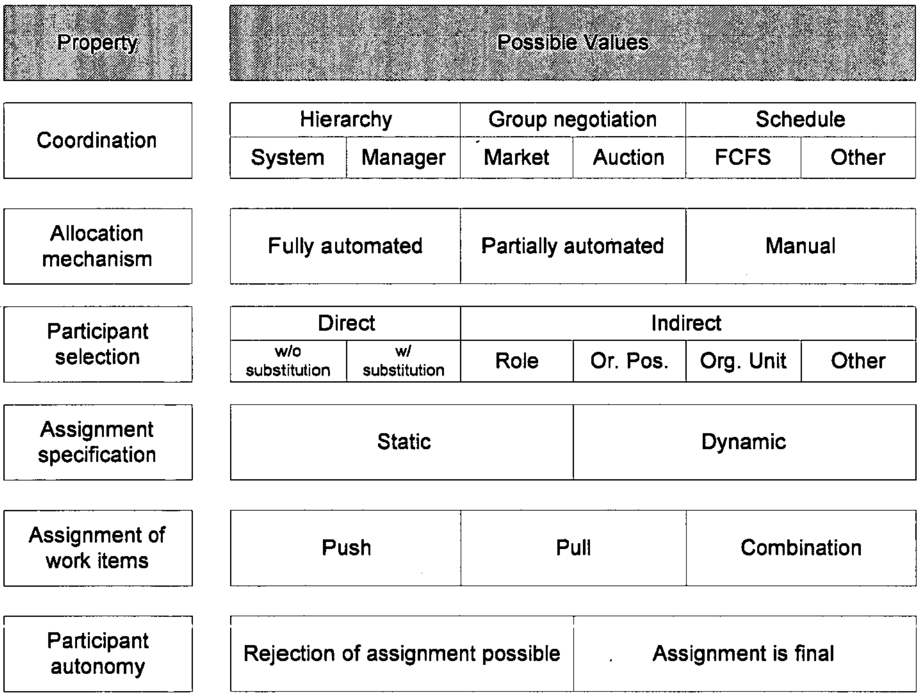
"The queuing of work items can be performed either using a queue, ensuring that work items are selected in the order in which they become available; a pool, where resources can choose freely between available work items; or a combination of the two, where resources select a collection of work items at a time. [...] "

"The form of activity execution (individual or collaborative) prescribes how many resources may select a work item for execution. While many workflow systems support only activities that are assigned to individual users, some research has been done on assigning work to teams. [...] "

"The decision hierarchy describes whether an activity can be passed on from a workflow performer to another delegate performer. This act of substitution is common during the absence of a resource (when a deputy takes over some or all of the functions of the assignee). [...] "

4.3.2 Synchronisation policies

The synchronisation policies are the second part of the model. Zur Muehlen's explanation of synchronisation policies is as follows: "Once a work item has been placed on a shared worklist, synchronisation policies determine how it can be accessed by individual workflow participants."
Note that, in contrast with this explanation and zur Muehlen's role resolution procedure model ([54]), we do not assume available work items are always placed on a shared worklist. According to [54], the allocation of a work item to an individual worklist is possible through a shared worklist, but we assume it can be directly allocated to the individual worklist by the workflow enactment service, without intervention of a shared worklist.



Property	Possible Values					
Coordination	Hierarchy		Group negotiation		Schedule	
	System	Manager	Market	Auction	FCFS	Other
Allocation mechanism	Fully automated		Partially automated		Manual	
Participant selection	Direct		Indirect			
	w/o substitution	w/ substitution	Role	Or. Pos.	Org. Unit	Other
Assignment specification	Static			Dynamic		
Assignment of work items	Push		Pull		Combination	
Participant autonomy	Rejection of assignment possible			Assignment is final		

Figure 12: Synchronisation policies (from [54])

Each row of Figure 12 contains a synchronisation policy that will be explained in the following.

"The coordination of pending work items can either be hierarchical or market-based. In a hierarchical scenario, either the workflow enactment service assigns work items to individual performers [...], or in an intermediate step, a manager receives all pending work items and manually assigns them to the individual workflow participants. While the first strategy is useful if the workflow management system is capable of performing schedule optimisation, the latter strategy reduces the "mechanistic" assignment of activities by a technical system, which may increase user acceptance. If a market-based allocation policy is implemented, workflow participants compete for the right to perform a particular activity, either through auctions or other market mechanisms. [...] Finally, work items can be assigned according to a fixed schedule. In the most common case, they are assigned to a shared worklist, from where they can be selected by individual performers on a first-come-first-served (FCFS) basis. However, other scheduling principles are theoretically possible."

"The allocation of work items can be fully automated by the workflow management system, i.e. the system automatically determines the qualified performer(s) for an activity and assigns pending work items to these resources. In a partially automated scenario, the workflow management system may suggest qualified performers, but the ultimate assignment is performed by another decision instance, e.g. a team leader. In a manual scenario, the workflow management system would only show pending activities, but not perform any task allocation. It is up to the workflow-enabled organization to find organizational measures to ensure the actual execution of pending activities, or an external resource management application could perform the assignment."

"The participant selection can be either direct (at the level of the individual performer), or indirect (using a proxy construct like role or organizational unit as the ultimate recipient of a work item). During direct assignment, the workflow system may check the availability of the selected resource, e.g., if the performer is currently logged into the system. If the resource is absent, the workflow system can determine a substitute and assign the work item to this resource, deliver the work item only to the initially chosen resource (e.g. if the activity is marked for direct assignment only) or raise an exception and escalate the process. Indirect selection is based on those elements of the resource meta model, which allow the grouping of individual resources. [...] "

"The specification of the resource requirements can be either static or dynamic. Using the static assignment, an activity is associated with one or more instances of one or more resource meta model entities (for instance role, organizational position, or organizational unit). For example, the activity "review customer order" could be assigned to the instance "sales manager" of the entity type "organizational position". Dynamic assignments take data from the current workflow instance into account for the selection of qualified performers. This data can either relate to the current workflow instance (e.g. the ID of the process initiator) or to the business objects processed in the current workflow instance (e.g. the customer number contained in an order document). Dynamic assignment allows a very detailed specification of assignment policies (e.g. all orders from a particular customer are handled by the same customer service agent), but require the tight integration of the workflow management system with external data sources. In practice, this type of assignment is found in embedded workflow solutions, where the workflow component has immediate access to all relevant system data."

"The assignment of pending activities can either be performed in a push or a pull manner. Using the push mechanism, resource signal their availability to the workflow system, which determines the next activity this performer should work on. This may be desirable in environments, when the workflow enactment service can optimise the scheduling of activities, the activities themselves are uniform, and the resources are mainly technical (without user intervention). [...] Using a pull mechanism, a resource requests the next work item at a convenient time, but not necessarily upon availability. A combination of push and pull is possible, if a resource has assigned a number of activities, but the sequence of execution of the individual activities is left to the performer."

"Participant autonomy determines whether the workflow enactment service determines the final recipient of a work item, or whether a workflow participant may refuse the assignment of a particular work item and send it back to the shared worklist or the workflow enactment service for reallocation."

Now we know on what basis adaptations to the technical system can be made, we can explore the effects of these adaptations in job characteristic terms.

4.4 Impact of assignment and synchronisation policies in JCM-terms

The first step in combining the two models is finding out which of the JCM-dimensions is affected by the policies of zur Muehlen's models. In Table 5, the influenced dimensions by the assignment policies are shown. Table 6 shows what effects the synchronisation policies can have on the job dimensions. Below, an elaboration and explanation of this comparison can be found. In Appendix H, a more detailed examination of the impact of the policies on the job dimensions is included.

Note that the order in which the job characteristics occur in the table has a meaning. The first characteristic that is mentioned is, in our opinion, more influenced by the assignment or synchronisation policy than the second one and so on. The explanations below will clarify these relations between job dimensions (e.g. through the impact on autonomy often skill variety is influenced too).

Assignment policies	**Impact on employee in JCM-terms**	
planning of new work items	autonomy	
time of notification	autonomy	
queuing of new work items	autonomy skill variety	
activity execution	autonomy skill variety	task significance feedback
decision hierarchy	autonomy skill variety	

Table 5: Assignment policies (from [54])

The *planning of new work items* affects the autonomy of an employee. When a net change strategy is implemented an employee knows (s)he once has to perform the work items that are assigned to him or her. On the contrary, when a re-planning strategy is followed, the employee is never sure whether he or she will keep the work items that are assigned to him/her at a certain moment. Therefore, it is not possible for employees to make their own plan for the order of execution of the work items. For example, when an employee usually performs the tasks he or she does not like at first, followed by tasks he or she likes more, there is a substantial risk that the nicest tasks will be re-planned and he or she will get boring tasks instead. Of course, the employee will not like this strategy and it will have a negative influence on the autonomy (the possibility to choose work items yourself) the employee experiences.

Time of notification also influences autonomy. If a work item is released upon availability there still is a lot of flexibility in the start time of performing the work item. An employee can choose to perform other work items before or can start immediately working on this work item. In contrast to this situation, when a work item is released at the latest start time, it has to be performed immediately and there is nothing to choose for the employee.

The *queuing of work items* is also related to autonomy (the possibility to choose work items yourself) and to skill variety (variation in the type of work items and the work to be performed). When a queue is used to present the available work items, an employee has to perform the first work item on the list. There is nothing to choose and he or she has no influence on the kind of work items that follow. When a pool is used, the employees can make their own selections of the available work items and they are able to select different kinds of work items. In this way (s)he can improve the skill variety of the job by selecting different kind of work items. Actually, this skill variety is caused by autonomy. When an employee likes to do the same kind of work he or she can also choose to work in this way and just select similar work items only, which will not improve skill variety.

We think the influence of *activity execution* can be very broad. Individual or collaborative execution of a work item can affect the autonomy, skill variety, task significance and feedback. When a work item is executed collaboratively the skill variety can be influenced negatively because every employee will do the part of the execution he or she is good at. In the same way the autonomy of an employee will be less because the employee is not alone responsible for the execution and can not make decisions on his/her own. Therefore, to improve autonomy and skill variety one may prefer to execute work items individually. On the contrary, the collaborative execution can improve task significance and feedback, by cooperating on a task and finding out what colleagues are doing and how they appreciate the part the employee executed. Moreover, feedback from team members is possible in case of mistakes made. In the organisational psychology much research is done on self-managing teams (see for example: [14], [53], [77]).

The last property of the assignment policies, *decision hierarchy*, also influences the autonomy of an employee. When there is a possibility to reject or delegate an assigned work item, the employee has more freedom in deciding which work items he or she wants to perform. Employees can reject the work items they do not like or they do not want to perform. This can have a positive impact on skill variety, because through this autonomy the employee can refuse to execute the same kind of work item over and over again. On the other hand, (s)he has also the option of exclusively performing the same kind of work if (s)he prefers to do that, by rejecting all other work items. This will influence skill variety negatively.

Synchronisation mechanisms	**Impact on employee in JCM-terms**
coordination	autonomy skill variety task significance
allocation mechanism	autonomy task significance
participant selection	autonomy
assignment specification	task identity task significance
assignment of work items	autonomy
participant autonomy	autonomy skill variety

Table 6: Synchronisation policies (from [54])

The first synchronisation mechanism, *coordination*, affects the autonomy, skill variety and task significance of an employee by the amount of freedom that is left in the assignment and schedule of work items. When a work item is directly assigned to an employee in a FCFS order, the employee has minimal influence on the (kind of) tasks he/she has to perform, but when the work items are for example assigned through an auction-mechanism, the employee can have much more influence on the work that is assigned to him/her and the kind of work he/she has to perform. Moreover, when a market or auction mechanism is used or when an employee has to negotiate with a manager in a hierarchical structure, an employee has to make judgements of the content of each work item and assign some kind of 'value' of importance or interest to a work item. This will provide him or her with some insight in the importance of the work items he or she finally gets to perform. Hereby, task significance is influenced in a positive way.

In our opinion the *allocation mechanism* can influence autonomy and task significance positively by using a manual allocation mechanism. An employee has the possibility to negotiate with for instance a manager if he or she wants to perform certain tasks. The manager can also try to convince an employee why he or she should perform other tasks and why those tasks are more important. In this way, the autonomy and task significance of an employee is affected. When the work is allocated automatically by the system, there is no possibility for negotiation or consultation.

The*participant selection* mechanism can have an impact on the autonomy of an employee. When this is done indirectly (for example via a role or organizational unit construct), it is easier to facilitate a shared worklist to give the employee a choice in selecting the next work item or to delegate or reject a work item. The work item has to be executed by someone performing that role, but it can be done by several people who are authorised to perform that role. It does not necessarily have to be a certain employee. When a work item is assigned directly to an employee (without intervention of a role construct), the employee has to perform the work. It is assigned to him/her, and to him/her only, and he or she has to perform it.

The *assignment specification* influences task identity and task significance. When this specification is dynamic, the workflow enactment service can control that a case will be performed as much as possible by the same employee. This decreases the 'start-up' time for every task, but it also contributes positively to the identity and the significance of the tasks that the employee has to perform. He or she will have more insight in and knowledge of the specific case.

The *assignment of work items* again affects autonomy. When this assignment is done in a push manner, the employee has to do what the system says: the work items have to be executed in the order and at the time the workflow enactment service offers them. While, in the pull construction, the employee has more freedom. They can signal their availability themselves and can pick up and start work items when they like.

Finally, *participant autonomy*, indicates the amount of freedom to reject or delegate a work item. When the assignment of a work item is final, the employee will experience less autonomy and when rejection of a work item is possible the employee will have more autonomy. As mentioned in the explanation of decision hierarchy in the previous paragraph, through this improvement in autonomy an improvement in skill variety can be achieved.

As can be read in these explanations, the policies and properties are strongly interdependent. Therefore, it is difficult to sharply differentiate between the influences of the policies on the job characteristics. Overall, we can see that mainly an employee's autonomy stands out as an often influenced job characteristic. But most of the technical adaptations also influence other job characteristics. In work and organisational studies, many researchers consider the extent in which an employee has the possibility to control (or in other words has autonomy) as one of the most important dimensions of work quality (see for example [12], [22], and [58]). The possibility to control is also a very important factor in the well-being of a person. An uncontrollable situation is more stressful.
In many cases the influence on skill variety, next to autonomy, could be justified too. This improvement in skill variety is reached by an improvement in autonomy. When an employee has more freedom in selecting the work items he or she wants to perform, he or she automatically has the opportunity to improve skill variety by selecting different kinds of work items. Therefore, we only mentioned skill variety in the above comparison if we thought it was a direct result of the policy. In all other cases we did not mention the impact on skill variety, because it can be seen as some kind of side effect to the improvement of autonomy. We tried to explain this side effect in several elaborations.

The comparison of the two models and the parts that influence each other inspired us to propose some new ideas to adapt a workflow system in such a way that the perception of the employees working in the system will be improved.

4.5 How to improve work perception in workflow systems?: ideas

For the generation of the ideas to improve work perception by "tuning" the workflow system in a human centred way, we have used the extended view on the workflow system as introduced in section 2.4. This section first elaborates on the creation of the ideas by giving some examples. Then, the ideas we have generated will be summed up followed by a selection of the ideas on the ease of implementation. The result is a list of 21 ideas for making small and easy adaptations to the workflow system.

4.5.1 Generating ideas: examples

First we will describe in what way the ideas for improving work in workflow management systems are created, for example by using the models and explanations of zur Muehlen. We will do this by giving some example lines of thought in generating an idea.

- In the explanation of the re-planning strategy in planning new work items, zur Muehlen states: "... while a re-planning strategy would re-allocate all work items that have not yet been started, possibly removing work items from some performer's worklists and placing them on other worklists". We think an employee will not like the fact that work that is allocated to him/her suddenly is removed or changed. Especially when there is freedom in the order of execution of work items. Some people first perform the most boring or less interesting jobs to have nice jobs afterwards.[20] When the re-planning strategy is chosen these people have a risk that the nice jobs get re-planned to some other performers before they can execute them. Of course, in this way the re-planning strategy affects the autonomy of an employee negatively. The effect is that an employee will learn fast and will turn to always performing the nicest tasks first. If all employees will do this then the most boring tasks will stay in the system, because no one wants to execute them. Therefore, one of the ideas we generated is: "Do not re-plan work items by workflow enactment service" (idea nr. 29).

- When people perform activities according to a certain role, they have to perform the same kind of activities over and over again, so probably they will get bored with it. If it is possible to teach them how to perform several different roles, it would also be possible to change roles from time to time (idea nr. 1) or to give employees a choice which role they want to perform (idea nr. 2).
- In [36], Hoffmann and Loser state that the autonomy of an employee is influenced positively when the decisions that have to be made in a process are left to the employees, even though they could be automated (idea nr. 23).[21]
- Zur Muehlen's explanation of queuing of work items ("The queuing of work items can be performed either using a queue, ensuring that work items are selected in the order in which they become available; a pool, where resources can choose freely between available work items; or a combination of the two, where resources select a collection of work items at a time.") inspired us to a number of ideas. For instance, the combination of a queue and a pool led to a design in which 'batches of work items' are allocated to a person (idea nr. 21). Within a batch the employee has a lot a freedom in for example the order of execution, but at the same time the workflow enactment service can easily control the work items to be executed. This idea will increase the autonomy of an employee.

4.5.2 The ideas

In this section, we present 32 ideas, which originated from the considerations of the previous section. Where appropriate we have indicated related and/or inspiring research for each idea.

1. Versatile employees: one employee can operate in different roles. (Control of role distribution is at manager or system.) This idea is similar to the concept of job rotation. (see for example [70]).
2. Versatile employees can choose the role they want to execute themselves.
3. Automate boring tasks, do not automate challenging tasks. ([36])
4. Keep up with the kind of activities an employee likes and make sure he or she will get more of this kind of activities (and less of activities he or she does not like). ([33])
5. Do not "over-specify" the content of an activity. When it is possible to have an amount of freedom in executing the activity, this freedom should be used. For example: when there are several ways to produce the output of an activity, let the employee choose in which way he wants to perform the activity. ([34])
6. Do not specify in what order parallel activities should be executed. ([34])
7. If possible, do not specify in what order the operations of an activity (the parts of the execution of the activity) should be executed. ([34])
8. Offer a variety in work items to an employee. Remember the kind of work items an employee has executed and decide, based on this history, what kind of new work items will be offered to him or her.
9. Use coupling/cohesion ratio to design activities in the workflow process model. (see also [69])
10. Make a uniform interface concerning other applications or systems that have to be used in executing an activity.

[20] Remark that most people do not use this work strategy, but it is a possible situation. Mostly, people avoid boring activities and perform nice activities first.

[21] This should only be done if performing the decision has an added value to the person (for example when a human has a better insight in making the decision).

11. Case management: let an employee work on the same case as much as possible. (See also [65], page 171, the ASSIGN allocation rule, and [54].)
12. Design a possibility for the employee to examine the static process model in a comprehensible way (static aspect).
13. Give an employee the possibility to check the progress and route of a case during the process (dynamic aspect).
14. Give each employee authorisation to view the final decision or result of a case in the process.
15. Return feedback of a customer to the organisation as detailed as possible and as far as possible to the workers on the shop floor.
16. Do not only show the parts of information to be completed during the execution of an activity, but also show the information that is already filled in and the information to be filled in after the activity.
17. Let an employee choose work items from the private worklist himself/herself: pull mechanism. ([33] and [54])
18. Use a shared work list, from which an employee can choose himself/herself: pull-manner. ([33] and [54])
19. If possible, show more than one work item on the worklist of an employee, even if a push mechanism is used. ([54])
20. The queuing of work items in the work list should be random. ([54])
21. Offer an employee "batches" of work items. In this way the batch is pushed, but the employee can choose the order of execution of work items within this batch. (Here we assume the work list is private.) ([54])
22. Give employees the opportunity to adjust the appearance of work items in their worklists to their own preferences: FCFS, earliest due date, random, etc. (Here we assume the assignment of work items is in a pull manner and the worklist is private.) ([54])
23. Decisions in a process should be executed by an employee, and should not be automated. (see: [36])
24. Create 'team batches' of work items. A team of employees (having the same competences/role) can divide the work according to their own preferences. (Here we assume the allocation mechanism is manual, but is not necessarily controlled by a team leader or manager.) This idea is quite similar to the concept of a self managing team, which is one of the hot items in organisational psychology (see for example: [53], [77]). ([33] and [54])
25. Create 'team work items'. Employees (with different competences) have to cooperate to execute an activity. ([54])
26. Give employees the possibility to send a work item to another employee, who is better in performing the job, who has more knowledge about the case, who is not busy, etc. ([33] and [54])
27. Give employees the possibility to reject a work item (with a valid reason) and return it to the workflow enactment service. ([33] and [54])
28. Release a new work item directly. (Time of notification is upon availability.) ([54])
29. Do not 're-plan' work items by workflow enactment service. ([54])
30. Make available an employee's place within the ranking of good employees (for instance 'hard working', 'producing high quality work').
31. Show an employee if he or she works hard enough, if he or she is satisfying the targets.
32. When a work item has to be performed again after a (negative result of a) check, return it to the same employee to execute it again.

The ideas presented above affect different parts of the total workflow system within an organisation. In the introduction on workflow management, we explained the meaning of the four levels of our model of the workflow system as a whole. Figure 13, contains the same visual model. For each idea a circle with the number of the idea is added to the model. This circle shows in which part of the model an idea is situated or which part of the model is affected by the idea.

As you can see, all levels are taking part in possible improvement, but most of the generated ideas are concentrated on the process automation layer and its interfaces to the user. This means the ideas mainly concern the distribution of work items, the assignment of work items to employees, and the interaction of an employee with the system.

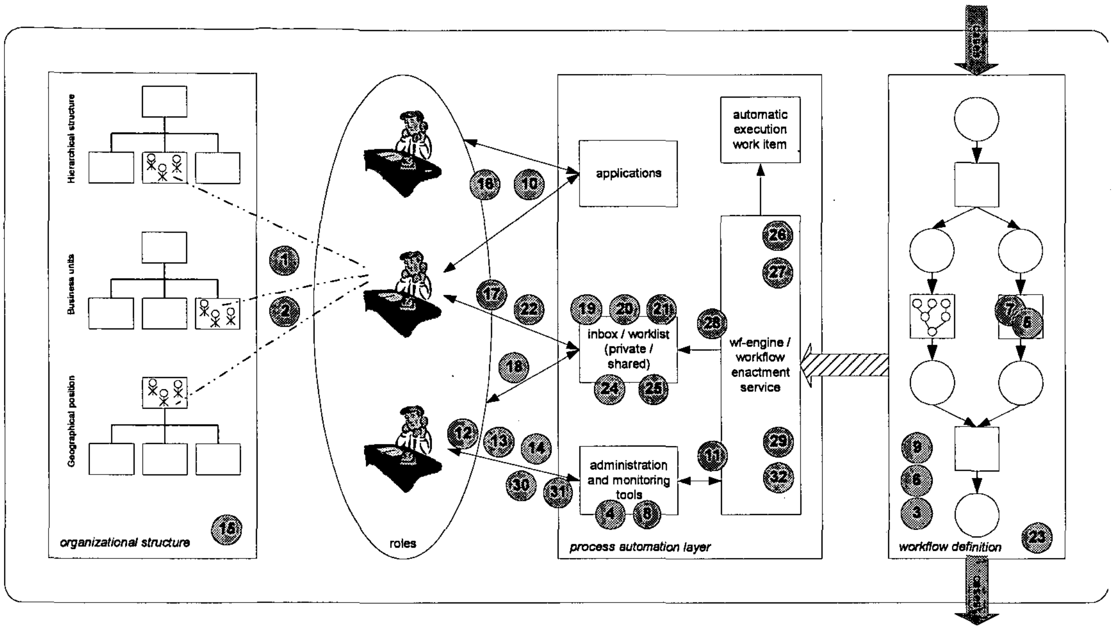


Figure 13: A visual representation of the parts of the whole workflow system that are affected by the ideas.

4.5.3 Selection of easy to implement ideas

For further research we will focus on improvement ideas that are easy to implement in an existing workflow system. This means, we want to explore the adaptations that can be made to an existing system in order to improve a user's perception of the work he or she is doing. By 'easy to implement' we mean those improvements that:

- are not very expensive,
- will not take a lot of time, and
- can be realised without making changes to the process or organisational structure. (In case of any changes to the process or organisational structure the involvement of major change programs is unavoidable. These change programs consist of for instance acceptance programs, training and education of employees, and clearly, they will take a lot of time and cost a lot of money.

Therefore, any idea which causes (re)design steps will automatically be eliminated, because (re)designing always takes time and money and should be managed from a change perspective.

When we take a look at the generated ideas and the model of Figure 13, this means in the first place that the content of a process can not be changed because the (re)design of a business process and the development of a new process model will take a lot of time. Many people are involved with designing the new process layout and therefore it will be an expensive improvement too. Therefore, ideas number three, five, six, seven, nine and twenty-three are eliminated.

Next, a change in the organisational structure of a company will also be a huge investment in terms of time and money. Changes of organisational structure can be of several kinds. For instance, the geographical position can be re-organised, but there can also be a change in the role-division among employees, or the required competences to a role or function. The re-training of employees for instance is expensive. Therefore, ideas one, two and fifteen are eliminated.

In this way the left hand side and the right hand side of the model of Figure 13 can be eliminated. The remaining ideas for improvement are concentrated on the process automation layer and its interfaces to the worker.

Finally, for the same reasons of ease of implementation, ideas ten and sixteen have to be eliminated. The (re)design of applications and their interfaces to the user will take a lot of time and money too.

The ideas that remain, are all linked to the actual process automation layer and its accompanying interfaces for distribution of work items and administration and monitoring of the process. Most of these remaining ideas are linked to the assignment and synchronisation policies zur Muehlen described ([54]).

Below you will find the list of remaining ideas. For ease of use in further validating these ideas we have clustered them into groups of ideas that affect the same part of the system. We also gave them an acronym that makes referring to the idea easier further on. This list is included in Appendix I again, indicating the corresponding ideas form the original list and the corresponding policies from zur Muehlen's model.

Release of work item / participant selection by workflow engine

1. [CASEMAN] Case management: let an employee work on the same case as much as possible.
2. [REDIRECT] Give employees the possibility to send a work item to another employee, who is better in performing the job, who has more knowledge about the case, who is not busy, etc.
3. [REJECT] Give employees the possibility to reject a work item (with a valid reason) and return it to the workflow enactment service.
4. [RELEASE] Release a new work item directly. (Time of notification is upon availability.)
5. [REPLAN] Do not 're-plan' work items by workflow enactment service.
6. [RESUBMIT] When a work item has to be performed again after a (negative result of a) check, return it to the same employee to execute it again.

Worklist

7. [PRIV PULL] Let an employee choose work items from the private worklist himself/herself: pull mechanism.
8. [SH PULL] Use a shared work list, from which an employee can choose himself/herself: pull-manner.
9. [#ITEMS] If possible, show more than one work item on the worklist of an employee, even if a push mechanism is used.
10. [RANDOM] The queuing of work items in the work list should be random.
11. [BATCH] Offer an employee "batches" of work items. In this way the batch is pushed, but the employee can choose the order of execution of work items within this batch. (Here we assume the work list is private.)
12. [APPEAR] Give employees the opportunity to adjust the appearance of work items in their worklists to their own preferences: FCFS, earliest due date, random, etc. (Here we assume the assignment of work items is in a pull manner and the worklist is private.)
13. [TEAM BAT] Create 'team batches' of work items. A team of employees (having the same competences/role) can divide the work according to their own preferences. (Here we assume the allocation mechanism is manual, but is not necessarily controlled by a team leader or manager.)
14. [TEAM WI] Create 'team work items'. Employees (with different competences) have to cooperate to execute an activity.

Administration and monitoring

15. [PREFS] Keep up with the kind of activities an employee likes and make sure he or she will get more of this kind of activities (and less of activities (s)he does not like).
16. [HISTORY] Offer a variety in work items to an employee. Remember the kind of work items an employee has executed and decide, based on this history, what kind of new work items will be offered to him or her.
17. [STAT MOD] Design a possibility for the employee to examine the static process model in a comprehensible way (static aspect).
18. [ROUTE] Give an employee the possibility to check the progress and route of a case during the process (dynamic aspect).
19. [RESULT] Give each employee authorisation to view the final decision or result of a case in the process.
20. [RANKING] Make available an employees place within the ranking of good employees (for instance 'hard working', 'producing high quality work').
21. [TARGET] Show an employee if he or she works hard enough, if he or she is satisfying the targets.

Now we have selected the 21 ideas that are easy to implement we would like to validate these ideas by asking (workflow) experts on their opinion. This expert validation will be described in the next section. After that, we will try to identify to which degree current workflow technology facilitates or supports these ideas.

4.6 Validation of the ideas

Of course we need to validate these newly generated ideas to "tune" a workflow system to justify our research on the combination of work psychology, organisational behaviour and workflow management. To carry out this validation, we have interviewed six experts on workflow management and business processes, with different backgrounds, to gather their opinion on the ideas. Based on their views we have selected the best six ideas. This section elaborates on this subject.

4.6.1 Qualitative expert validation

The expert validation is done by asking six experts to give their view on the 21 ideas. All interviews were taken face-to-face, providing the possibility to give more explanation where needed, except for one (of respondent R5). He answered the questions by e-mail.

The six experts can be divided into two categories: three of them are researchers and the other three are people with practical experience in the area of workflow management. Respondent 1 (R1) and respondent 4 (R4) are workflow designers with a Dutch consultancy firm and a Dutch bank, respectively. R6 is a workflow project manager with a Dutch insurance and banking company. R2, R3 and R5 are researchers at respectively Delft University of Technology (area of information systems/business processes), Eindhoven University of Technology (organisational behaviour) and Stevens Institute of Technology (process automation and workflow management). We have to note that R2 also has a lot of practical experience in workflow projects and that R3 is an expert on work psychology and organisational behaviour. Her expertise in technical systems is less, which made it not very feasible to answer the question of ease of implementation for each idea.

We asked the respondents to indicate for each idea whether they think the idea has a positive impact on the worker and whether it is easy to implement (see questionnaires in Appendix J). There was also left some space for their own comments. Finally, we requested to list the top five of ideas they thought were best. The results of the questionnaire can be found in appendix K.

4.6.2 Analysis of expert validation

To analyse the experts' opinion, we focus on the experts' top fives. To process this information we transformed their rankings into a ranking with stars. The higher an idea is situated in the top five of a respondent the more stars it gets. For each respondent, the number one idea gets five stars, number two gets four, number three gets three, etc. For example, R1 thinks idea number 14 is the best and takes the first place. Therefore, it has five stars. (See also Table 7). Based on the total number of stars we can select the overall best ideas, which will be explained below.

From this overview of top five ideas according to the respondents, we composed a list of the six (overall) best ideas. This composition is based on two different approaches to select the top ideas.

The first approach considers the height of the ranking of the ideas. The ideas with the highest ranking (or the greatest number of stars in Table 7) are selected. We can see idea number 8 is the best (16 stars), followed by idea 21 (12 stars) and idea 6 (8 stars). Next, three different ideas have 7 stars: idea number 1, 12, and 13. Their mutual ranking can be formed by considering the number of selections and the places in the individual rankings. Idea number 13 will then be first because it was selected three times. After that, idea number 12 follows with two selections and a number one place and finally idea number one is selected with two individual selections and a second place as the highest ranking.

The second approach to the selection of the six best ideas considers the number of individual selections as most important. In this case, idea number 8 and idea number 6 are the best ideas, with four individual selections, followed by idea number 13 and 21 (three individual selections). Then seven ideas are selected twice. We will make a difference between these ideas by considering the total number of stars as a second important criterion. In that case idea number 1, and number 12 are selected. (See also Table 7.)

	R1	R2	R3	R4	R5	R6	Total number of *-s		Number of selections
1	****			***			*******	***(7)***	**2**
2			* [1]	****			***** [1]	(5)	2
3		*	* [1]				** [1]	(2)	2
4								(0)	0
5			****				****	(4)	1
6		**	**		*	***	********	***(8)***	**4**
7	**						**	(2)	1
8	***	***	*****			*****	****************	***(16)***	**4**
9								(0)	0
10								(0)	0
11								(0)	0
12				*****		**	*******	***(7)***	**2**
13				**	****	*	*******	***(7)***	**3**
14	*****			*			******	(6)	2
15					*** [1]		*** [1]	(3)	1
16					*** [1]		*** [1]	(3)	1
17		****					****	(4)	2
18	*				**		***	(3)	2
19					*****		*****	(5)	1
20								(0)	0
21		*****	***			****	************	***(12)***	**3**

[1] Note that two respondents selected two (similar) ideas at the same rank in the top five. They liked the general idea behind it, but could not distinguish between the two rather similar ideas. Therefore, we both recorded them in this summary, because they could have some contribution to the list of best ideas extracted from these experts' views. However, in the end they did not show up as one of the top ideas.

Table 7: Interview results of the top five of best ideas per respondent.

When comparing these two approaches it appears that the same six ideas are selected, although their mutual ranking is different[22]. This gives us some confidence that these six ideas are really the best ideas:

1. **Idea no. 8 [SH PULL]**: "Use a shared worklist, from which an employee can choose himself: pull-manner."
2. **Idea no. 21 [TARGET]**: "Show an employee if he or she works hard enough, if he or she is satisfying the targets."
3. **Idea no. 6 [RESUB]**: "When a work item has to be performed again after a (negative result of a) check, return it to the same employee to execute it again.
4. **Idea no. 13 [TEAM BAT]:** "Create 'team batches' of work items. A team of employees (having the same competences/role) can divide the work according to their own preferences. (Here we assume the allocation mechanism is manual, but is not necessarily controlled by a team leader or manager.)"

[22] The difference in mutual ranking will not cause any problems, because we will not use this ranking in further research.

5. **Idea no. 12 [APPEAR]:** "Give employees the opportunity to adjust the appearance of work items in their worklists to their own preferences: FCFS, earliest due date, random, etc. (Here we assume the assignment of work items is in a pull manner and the worklist is private.)"
6. **Idea no. 1 [CASEMAN]:** "Case management: let an employee work on the same case as much as possible."

As it turns out from the tables with interview results, the opinions of the experts were not unanimous. In the following, some specific notes from the questionnaires are extracted to illustrate the different viewpoints on the overall best ideas:

- For *idea number 8* [SH PULL], the interviewees all agreed on the positive impact on the worker and the ease of implementation. Some of them remarked that a shared worklist is usually implemented in a workflow system. They even question whether it is practically possible to implement a private worklist in some situations, with respect to the durability of the process and the quick processing of cases. From a flexibility viewpoint of the system they think it is best to use shared worklists.
- R5 and R6 doubt if *idea number 21* [TARGET] has a positive impact on the worker. R6 thinks the feedback on satisfying the targets should not be on an individual level, but an employee should have the possibility to check the average results of himself/herself and of the department he or she is a member of. On the contrary R1 thinks it is a good idea, if the results remain private (i.e. they are only shown to the person to whom they apply). Concerning the implementability of idea number 21, the opinions are differing too. R5 thinks it is not easy to implement, while R6 thinks the requesting of management information from the system is easy (it is already implemented in the systems of his company).
- Except for one, the respondents are positive about the impact of *idea number 6* [RESUB] on the worker. R3 remarks that the returning of a work item should only happen once. After that, when it is checked negatively again, probably there is something else wrong and it has to be found out what. The respondents do not expect any problems with the implementation of this idea.
- Although the respondents are quite positive about the impact on the worker with respect to *idea number 13* [TEAM BAT], R3 thinks it is not a good idea. This is because people need an individual part in their work, otherwise there is a risk that the most powerful employees play a too important role and get to be the "boss". Furthermore, team work is only useful if dependency between employees requires it. About implementability, R1, R2 and R6 remark that the success of this idea depends on the kind of work that is supported. Self-managing teams will only work when the team members are professionals. Moreover, this idea seems to require a certain structure of the organisation, which has to be adapted in certain situations.
- According to the respondents, *idea number 12* [APPEAR] is a good idea except for the random part. R6 indicates that within his organisation it is already possible for employees to sort or filter their (shared) worklists. R2 thinks it is possible to implement this idea but from a control point of view on the process, it is not desirable to give employees freedom in choosing what way the work items are represented (and in what order they are executed). R1 also notes that it can be too much freedom for the worker. Finally, R3 thinks it is a good idea, but it is not possible to have a random worklist, because there is always an order in the work that has to be performed (i.e. earliest due date, urgent orders, etc.).
- With respect to the last idea, *idea number 1* [CASEMAN], the respondents are unanimous about the impact on the worker. They all think it is positive. It creates a feeling of ownership and responsibility (R1). R2, R3, R6 all remark that it depends on the process if it is easy to implement: some processes do not have activities that can be executed by the same people, or by the one role, because the activities are too different.

Although the experts do not agree on all aspects and particularly not on the importance of each aspect, the two different approaches to the selection of the best ideas, based on the experts' opinions, led to the same top six of ideas. Clearly, these six ideas are the best and will be used for a further identification with actual workflow management systems in the next section.

The comments made by the experts, make it also clear that some ideas present implementation problems we did not foresee. For instance, idea number 13 provides more autonomy, but it suggests the organisation is used to work in teams. Even if the technical implementation poses no problems, the idea implies a certain organisational structure and the fact that people must learn to negotiate and cooperate might present difficulties.

Moreover, the implementation of certain combinations of ideas is not possible, if these ideas are contradicting. For example, when team work (e.g. idea number 13 [TEAM BAT]) is introduced, it is not possible to assess employees individually (e.g. idea number 21 [TARGET]).
Surprisingly, when we take a look at the workflow system model again in section 4.5.2, we can conclude that three of the selected top ideas affect the interfaces between the users and the system and the other three cover the workflow management system itself (the kind of worklist and the way of assigning work items of a case to an employee based on the administration of the employees that have performed other activities of the case). Therefore, we can conclude that the six selected ideas are well spread among the parts of the workflow system that are easy to adapt.

4.7 System identification

In this section, the use of the six best ideas in present workflow technology is tested with three actual workflow management systems to find out if the ideas can be implemented in the existing systems. If not, we will try to give some directions to extend the workflow management systems in order to support the idea. Where appropriate we will link the idea with existing literature and research. This is only to give a notion of related research, without aiming to be complete.

First of all, we will identify whether three actual workflow management systems (Staffware, FLOWer and COSA) are able to support the ideas we developed to improve a worker's experience of the work he or she is doing. Because of time limitations to this final assignment we will not examine the possibilities to support all ideas that are generated before. But we will use the six best ideas as obtained from the previous section, to check whether they can be implemented in actual workflow management systems.
Moreover, this system identification is done based on documentation only. We have examined the user's guides and developer's guides of the three workflow management systems ([40], [41], [51], [52], [59], [60], [61]) and gathered a lot of information. Still, it is possible that this information is not complete. Therefore, it would be nice to extend this research with system experts' views and an experiment to implement the ideas in the actual workflow management systems, in order to get a better-grounded judgement on the implementability of the ideas in present workflow technology.

Staffware, FLOWer and COSA are different workflow management systems. While Staffware and COSA are good at production workflow (i.e. handling a large number of cases, that all have to be processed in a similar way), FLOWer is a more flexible system, based on the case handling paradigm ([7], [68]).

	Staffware 9.0	FLOWer 3.0	COSA 4.2
Idea no. 8 [SH PULL]	+	-	+
Idea no. 21 [TARGET]	-	-	+/-
Idea no. 6 [RESUB]	-	+	+
Idea no. 13 [TEAM BAT]	-	-	-
Idea no. 12 [APPEAR]	+	-	+
Idea no. 1 [CASEMAN]	+/-	+/-	+/-

Table 8: Summary of the result of implementability of the six best ideas.

In Table 8, you will find a summary of this system identification. Before going deeply into the technical aspects of the system we have to nuance this validation a little. The degree to which the actual workflow management system is able to support an idea can be split up into three levels:

1. The idea can be directly supported by the workflow management system itself (indicated with a "+")

2. The idea can be partly supported by the workflow management system, some small adaptations to the system have to be made or some "add-on's" have to be installed (indicated with a "+/-").
3. The idea can not be supported by the workflow management system, or the underlying concept of the workflow management system makes the facilitation of the idea not possible (indicated with a "-").

As turns out from Table 8, not all the ideas can be implemented or supported by the three present workflow management systems we considered. Below, a more detailed explanation can be found.

4.7.1 System identification for idea no. 8 [SH PULL]

In Staffware it is possible to create "group queues", implementing the idea of a shared worklist from which all members of a certain group (role) can select the work items they want to perform. (see also [79], page 34). Likewise, in COSA exists the "group work items" worklist (see [52], page 3-4). Thus, idea number 8 can be implemented using Staffware and COSA.

In FLOWer there is no inbox or worklist of an employee or group as we consider it to be. Through the so-called "Case Query" interface, an employee can search for cases, satisfying certain user-defined selection criteria. From the obtained list he or she can select a case on which he or she wants to work and check it out. This means that the whole case is checked out and any activities that can be executed in parallel for this case are checked out too. Thus, the "Case Query" is some kind of shared worklist, only, it does not contain work items, but cases. To select a work item from a specific case you have to go to the "Case Guide" of that particular case and ask for the next performable work item. But this does not support the idea of having a worklist. It is only possible to request the next activity to be executed for a certain case.

In summary, FLOWer supports the concept of having a shared worklist, which means that several people can select the work from the same list. But FLOWer does not support a worklist of work items. This makes us conclude that the idea is not supported by FLOWer, due to the concept of case handling on which the FLOWer system is based.

4.7.2 System identification for idea no. 21 [TARGET]

Staffware itself does not support a possibility to view management information on cases and workers. However, IDS Scheer developed the ARIS Process Performance Manager (PPM) tool that can be added to Staffware (SPM). It turns out (from [40] and [41]) that SPM only is meant to support the controlling of the efficiency of running business processes. This means it provides management information about the cases and not about (individual) employees. Employees can use the functionality of SPM to get (aggregate) information about the process and the cases, but they can not ask for information about their own performance or targets. However, in some way SPM implements this idea on a higher level that is an employee can check the performance of his department. But the original idea aimed for a very specific feedback to the employee on his results, performance and targets. Therefore we conclude this idea is not yet implementable with the existing extensions of Staffware.

With the "Case Monitor" in FLOWer you can find some management information on the cases (like number of cases, percentage of certain kind of cases, throughput time, etc.). See also [61]. But this is on a very high level and it is not possible to get information about individual workers, like in Staffware. We conclude that FLOWer does not support this idea.

In COSA an employee can ask for specific (statistical) information on cases and processes, like in Staffware and FLOWer, but (s)he can also use the "activity journal" to check the work items (s)he already performed (see [52], page 4-24). In this way, the employee can get some kind of information on his or her performance. But it seems that he/he should draw a conclusion about the performance himself/herself. Therefore, we think COSA only partially supports this idea.

On this part some research has been done by other researchers. Hur et al. ([38]) developed a "customizable workflow monitoring"-system on top of workflow management systems, that allows employees, based on their authorisation profile, to see all kind of information about the process: it includes a process tracer, process statistics, bottleneck analysis and workload. Within the workload part, statistics are calculated about the number of activities the employee performed (possibly in comparison to other employees), the average activity duration time and the overdue activity ratio.

4.7.3 System identification for idea no. 6 [RESUB]

When defining activities in FLOWer it is possible to indicate which person or which role is allowed to "redo" an activity. By using the additional constraint "same user" you can make sure the activity is carried out by the same user that performed the activity before (see [59]). Therefore, FLOWer supports idea number 6.

In Staffware, it is not possible to implement this construction. Although, it is possible to fix who has to perform an activity by the person's name, it is not possible to give some kind of dependence relationship. Or in other words, in the properties of an activity can be stated that 'Charles' has to perform the activity, but not that the same user has to performed the activity that also did it the first time.

In COSA, a work item can be sent back to the sender of the work item (by "SEND_USER", [51], page 8-5). In this way the work item can be sent back after a check to the user who performed the activity before, which implements idea number 6.

4.7.4 System identification for idea no. 13 [TEAM BAT]

The Staffware documentation does not state whether and in which way working in teams can be supported. We conclude there is no possibility to support for instance negotiating or manually distributing the work items among the members of a team with the same skills and competences. For FLOWer and COSA our conclusion is the same.

A workaround can be the use of one worklist by the whole team. But to implement idea 13, we would like the workflow management system to support the negotiating and manual distribution of work in a team. Probably this can be done by having the inbox for a team and supporting the sub distribution of the work items to individual worklists after negotiations. This means there has to be some kind of intermediate 'team worklist'. This is not (yet) supported by Staffware, nor by FLOWer or COSA.

Probably, its (future) realisation can be done using idea number 2: the redirection of work items to another user. At this point in time, COSA is already able to support the redirection of work items by the "reroute"-option of a work item for a user ([52], page 4-21). The user himself/herself can select a work item and indicate to which user it should be rerouted.

In Staffware, the construction of redirecting work items is partially implemented. It is only possible to redirect a whole worklist to another user (see [80], page 79). Probably this is done to overcome the problem of employees that are absent for a certain period, due to illness, or holidays, for example.

Finally, FLOWer does not support the redirecting of work items at all, because it does not work with worklists of work items and individual worklists. As is said previously, this is due to the different concept of case handling FLOWer is based on.

4.7.5 System identification for idea no. 12 [APPEAR]

Staffware provides an option to organise your own 'worklist items'. The sort, filter and display options can be set in a specific menu to adjust the appearance of the work items in the worklist to your own preferences (see [80], page 55).

In FLOWer it is possible to sort and filter in the "Case Query" mode. Thus, it is possible to adapt the appearance of cases to your own preferences. But the sorting and filtering does not concern the work items, but just the cases. Because of FLOWer does not have a worklist with work items we consider this idea as not implementable by FLOWer.

COSA provides a possibility to filter on specific search criteria and to sort the listbox of work items, based on the attributes ([52], page 4-48).

4.7.6 System identification for idea no. 1 [CASEMAN]

In Staffware, COSA and FLOWer it is possible to create something like a very strict version of case management. It is possible to set down for each activity that it has to be performed by a certain user, like the one who started the case or the one who performed previous work items from the case.

In Staffware this can be implemented using 'sw_starter' as a reference to the user who started the case. When 'sw_starter' is indicated as the addressee of an activity, the work item corresponding to a particular case will be routed to the user who started that particular case. (See also [79], page 34.)
FLOWer supports this by linking all activities together (as explained in the description of the implementability of idea 6). (See also , page 203.) For each activity you can indicate that it has to be performed by the user who performed another (previously performed) activity from the process.
The ways to implement a strict version of case management in COSA are similar to those of Staffware and FLOWer. For an activity you can indicate that it has to be sent to the "CREATOR" (i.e. the user who started the process instance), to the "OWNER" (i.e. the current process instance owner) or to "SEND_USER" (i.e. the user who initiated the entry of a work item in a worklist and/or who ran the previous activity instance (sender of the work item).

When idea number 1 is implemented this way, the work items of a particular case are always sent to the same person, but you will lose some flexibility that, in our view, is contained in real case management. The "as much as possible" in the description of this idea refers to the possibility that when the employee, who is assigned to the case, is not able to perform an activity (due to for instance illness, holiday, etc.), the work item can be routed or passed on to another employee.
Due to this 'inflexibility' this idea can only be partly supported by Staffware and FLOWer.
However, in COSA this strict case management can be a little stretched. COSA provides a possibility to define a logical expression for the addressee of a work item. In that way, it is possible to define for example that a work item has to be performed by "the creator of the case or by an account manager of the same department, or by a manager, but not the one from the same department". This provides a useful start to the facilitation of more flexible case management.

4.7.7 Analysis of system identification

From this comparison of the implementability of the six best ideas in three present workflow management systems, we can conclude that there are some striking differences between the systems. But there are some similarities too. The most important remarks that can be made are the following.

- By taking a focus on case handling (i.e. a focus on the case and not on the process), FLOWer does not provide a worklist to the employee containing the work items that he or she is allowed to execute. Instead, it contains a list of cases that are "in the pipeline". An employee can select a case from this list and ask for the next activity to be performed for this case, but when doing so he or she is not aware of the kind of work that has to be done in this case. The "Case Query" mode looks something like a worklist, but it does not contain concrete pieces of work. It is more like an overview of the work in hand, but the pieces of work are not assigned to an employee, or group of employees. Therefore we do not consider this concept of a case list as similar to a worklist. The ideas concerning a worklist by definition are not supported by FLOWer. This results in a somewhat more negative judgement for FLOWer with respect to the support of the ideas. FLOWer may appear worse than actually is the case.
  Because of case handling, FLOWer already creates more flexibility in the workflow management system than traditional workflow management systems do. In FLOWer the structure of the process model is not as rigid as in traditional systems. If this flexibility is used correctly it can even be translated into more autonomy to the worker, which would be a positive argument for FLOWer.
- Furthermore, we think it is surprising that in all three systems a lot of monitoring can be done with respect to the cases, work items and the process, but not on an employee's performance. To us, it seems to be a natural extension to a workflow management system to provide the employees with information on their own performance, besides the information they can request on the process itself.
- Finally, the most striking difference between the systems is that Staffware does not support idea number 6, while FLOWer and COSA do. Even though Staffware provides a possibility to not hard encode the name of an addressee of a work item, when it has to be assigned to a specific person (by "sw_starter"), this facility is on its own and its range is not wide enough.

4.8 Conclusion / future work

Based on theory in organisational behaviour, psychology and workflow management, we developed a number of ideas in this chapter to adapt ("tune") a workflow system in such a way that the working within the system will become more pleasant to the employee. The reason for this research is the risk of getting boring and monotonous work by the introduction of a workflow management system in an organisation, due to the rigid and less flexible structure of the designed and supported workflow process, identified by Kueng in [49]. And by this, causing unmotivated employees and less performance of the process.

As turns out from a literature study in the area of socio-technical theory and psychology, a lot of research has been done on the human centred introduction of automation systems. Less research did focus on the actual user-friendly design of the system itself. Particularly, we did not find any research on the human oriented design of workflow systems. Therefore, we made a combination of models and theory from the area of psychology and workflow management to initiate the research on this part. In our opinion, the change to a workflow management system can have more success if it is properly introduced *and* properly designed to work with. We have not aimed for development of a complete spectrum of guidelines for human oriented workflow system design, but for a new view in the (re)design of workflow systems, resulting in a number of ideas to adapt the workflow system in an easy and human centred way.

To generate these ideas to "tune" a workflow system we used our own extended view of the workflow system as a whole, as introduced in chapter 2, because employees do not only have to perform their work using the workflow management system. They also have to deal with the division of roles, the organisational structure, etc., which we all consider as part of the workflow system.

The ideas are validated with an expert validation of six experts with different backgrounds in research and/or practical experience in workflow projects. They all gave their opinion on the ideas and selected the five ideas they thought were the best ideas. Based on these top fives we selected six overall best ideas and identified to what degree current workflow technology is able to support or facilitate these ideas.

Although most of the generated ideas seem promising, according to the expert validation, we have to remark that work experience still is a very subjective thing. Job characteristics cannot be judged in figures or statistics. They are about one's experience. One person may like challenging, alternating work, while the other thinks performing the same tasks over and over again is nice for him. Therefore, it is not possible to quantitatively express the quality of a job, which makes it hard to discuss proposals for improvement. Moreover, we adopted only one of the functional views on job design and have not considered other job design theories in the generation of our ideas.

But even though these differences in experience exist, an employee should experience a certain amount of autonomy in the work he or she performs, which makes him or her feel better, perform better and decrease the risk of getting ill. As explained before, there is scientific evidence that employee satisfaction and motivation are related to performance, although the correlation between satisfaction and performance is low. It is generally believed that a well-designed job, with respect to the job dimensions, is better to the employee. The ideas presented in this chapter mainly provide the employee with more autonomy, but some of them also affect other job characteristics.

The research on a human oriented approach to workflow system design has not yet received much attention, and it can be extended in several ways. First of all we would like to further validate the generated ideas by asking the opinion of end-users of workflow systems. In the qualitative validation we performed, end users were not involved, although we think they will have a different and very important view on the ideas than the workflow experts.

Secondly, the system identification we did is only based on documentation of the workflow management systems, which means it is quite brief. To be more complete we would like to extend this identification by asking experts or developers of the systems about the possibilities to implement the ideas and finally, we would like to try to implement them in the actual workflow management systems.

Besides that, we only used the six overall best ideas to identify the possibilities of facilitating these ideas with present workflow management technology. This identification can be extended with the

other ideas we generated, or at least with the ideas that once or more appeared in the top five lists of the experts (that would be ideas number 2, 3, 5, 7, 14, 15, 16, 17, 18, and 19, see Appendix H).
Also, we only examined three present workflow management systems, representing the current state of workflow technology. In future research other systems, like MQ Workflow, FileNet, InConcert, and even less conventional workflow management (support) systems, like SAP, should be examined too.

Finally, the ideas should be tested in practice, by implementing them in a workflow runtime environment and examining their effects. When they seem to have a positive impact, possibly some general workflow system design guidelines can be formulated to design user-friendly workflow systems.
Two recommendable designs to execute a field study in this case are (see also [22]):

- A measurement of the variables (e.g. autonomy, skill variety, task significance, task identity, feedback, satisfaction with the system, task motivation, and output) before and after implementing the ideas using a control group comparison. This field study design is also called the *untreated control group design with pretest and posttest* ([22]). During pretest the control group and the test group are measured under the same circumstances. After that, the situation is changed for the test group and during the posttest the control group works in the old, unchanged situation, while the test group members do their work in the changed system.
  It is possible to test the impact of a single idea or of a combination of ideas, dependent on the expectations on the impact of the ideas (i.e. the test group is working in a situation with only one idea implemented, or a combination of ideas). Moreover, it is possible to use several test groups, that all are concerned with a different combination of improvement ideas.
- A *nonequivalent dependent variables design* can be used ([22]). This design requires a single test group that is pretested on for example two scales. One of these scales is expected to be affected by the proposed change and the other scale is not. This means that before the research is conducted a hypothesis on the controlled and affected variables has to be stated (i.e. researchers have to determine beforehand which variable should be affected by the change and which should not).

It would be best to design an experiment, based on theoretical expectations of the results. For example, when a positive effect is expected from each idea, it is possible to compare different groups with the control group. For each group with more ideas, the impact of the change should be higher than the control group or a group with less implemented ideas.
Besides this, there is also a choice in the variables that are measured during the pretest and posttest of the experiment and how many times they are measured. Because results can fluctuate it would be best to measure them several times. In this way it is possible to show that, although they are fluctuating, the fluctuation is on a higher level after implementation of one or more of the ideas.

5 Conclusion

With the research described in this thesis we developed two methods for (re)designing workflow systems. To conclude this thesis we will reflect on the research questions we introduced in chapter 1 and summarise the research and results we conducted.
Before elaborating on the research, we explained some backgrounds on workflow management and introduced our own specific view on the workflow system, which has been especially important in chapter 4. Based on this workflow system model we clarified the parts that are covered by our research questions.

The first research question covered a very specific part of the workflow system: the process model within the workflow definition level (and even in more detail, the information element structure of the process). The goal of this question was to develop a method to design well-structured activities within a workflow process, based on the processing of information in the process. The detailed research question has been formulated as follows:

"How can operations be clustered best in to activities?

The method for redesign we developed, provides a way to design well-structured activities within the process model, by considering the underlying operations on information in the process. From the field of software engineering, we adopted the rule of thumb "*strong cohesion, loose coupling*", which means that the operations within one activity must be cohesive, while the activities mutually are loosely coupled. To measure these characteristics we developed cohesion and coupling metrics and the coupling/cohesion ratio, based on the information element structure. This ratio can be used to determine the strongly cohesive and loosely coupled design among several alternatives (i.e. the one with the lowest value for the ratio). This heuristic is applied to a toy example and to an actual workflow redesign, which revealed some difficulties and dissimilarities in the information processing view on the process and the view on process redesign actual workflow designers adopt.
However, we think this method can be very valuable in supporting workflow designers to design better structured process models and the possibilities for future research are many.

The research question for the second part of this final assignment has been formulated as:

"In what way can a workflow system be adapted to improve an employee's experience of the work he or she is doing within the system?"

At first, the focus of this research question was on the whole workflow system as described in the introduction. However, we limited our field of research later on to the process automation layer and its interfaces to the employees. Based on Job Characteristics Theory and workflow management research, we developed a number of ideas to adapt the workflow system in order to make the work more enjoyable for the workers, without changing the content of the work. By "tuning" the technical assignment and synchronisation policies, described by zur Muehlen, in a right way, we were able to generate some ideas to achieve a positive effect on job characteristics and to improve an employee's work perception in a workflow system.
The ideas were validated by experts, resulting in a list of the six best ideas. For these best ideas we identified the possibilities to support or facilitate them with current workflow technology in the form of three actual workflow management systems. It turned out that there are some striking differences and resemblances between the systems. Overall we can remark, that not all the ideas can be implemented already using present workflow management systems. But we think this view can be important to the future development and design of workflow systems. Of course more research is needed, but probably these ideas can lead to some general guidelines for human oriented workflow system design.

Glossary

Activity – An activity is an "atomic" process: one that is not further subdivided into component processes. It thus is a logical unit of work; in other words, an activity is either carried out in full or not at all. An activity is not itself linked to a specific case. When an activity is carried out for a specific case, we refer to it as an activity instance. We also differentiate between manual, automatic, and semi-automatic activities. A manual activity is performed by a person, without any intervention by an application (for example, the signing of a document). An automatic activity is one performed by an application without any human intervention. A semi-automatic activity involves the use of an interactive application (for example, a word processor). (*modified from [5] and [85]*)

Activity instance – An activity instance is the carrying out of an assigned activity. In contrast to an activity an activity instance is related to a specific case. (*modified from [5] and [85]*)

AND-join – An AND-join is an activity that may only be carried out once certain conditions have been met. We can compare an AND-join with a stage in assembly that can only take place once all the necessary components are available. An AND-join is applied at the moment when several parallel workflows need to be synchronised. Using the AND-join, it is possible to coordinate various parallel workflows for a particular case. (*from [5]*)

AND-split – An AND-split activity is the logical opposite of an AND-join activity. Carrying out an AND-split results in more than one parallel workflow being created for the same case. We can also say that an AND-split divides a case into various parts, which can be worked upon simultaneously. (*from [5]*)

Attitudes – Attitudes are evaluative statements – either favourable of unfavourable – concerning, objects, people or events. They reflect how one feels about something. Attitudes consist of three components: a cognitive component (the opinion or belief segment of an attitude), an affective component (the emotional or feeling segment of an attitude) and a behavioural component (an intention to behave in a certain way toward someone or something). (*from [70]*)

BPR – Business process re-engineering or business process (re)design is the process of (re)assessment, analysis, modelling, definition and subsequent operational implementation of the business processes of an organisation, or other business entity. (*from [85]*)

(Business) process – A business process is one focused upon the production of particular products. These may be either physical products, such as an aircraft or bridge or less tangible ones such as a design, a consultation paper, or an assessment. In other words, the "product" can also be a service. (*from [5]*)

Case – A case is what a workflow management system is designed to control. We can also regard it as a "product in progress". Examples of a case could include an insurance claim, a mortgage application, a tax return, an order, or a course of treatment in a hospital. Each case has a unique identity. Moreover a case is always at a particular stage of development at any given moment. (*from [5]*) It can also be referred to as a workflow process instance.

Case manager – A case manager is a person who is responsible for the handling of a whole case or a set of several activities for the case. (*from [5]*)

Conditional routing – Because most processes need to be able to handle various types of cases, not all cases proceed through a given process in the same way. In other words, there may be various routes through a process. In order to ensure that - dependent upon a case's characteristics – a particular route is chosen, we can make use of the OR-split or the OR-join. For each case, an OR-split selects form a number of alternative activities for each case. These different routes can be reconverged using an OR-join. *(from [5])*

Inbox – The way in which the worklist is presented to the employee. Usually this looks like the inbox of an e-mail program.

Iteration – Iteration is possible within a workflow if its structure permits one or more activities to be performed repeatedly. An iteration may, for example, result from a quality control: as long as the result of the task is unsatisfactory, it must be repeated. *(from [5])*

Job – An employee has to perform his job. With job we refer to the total range of activities and tasks he has to perform. These activities can be part of one workflow process but it is also possible the employee has to perform several activities from several processes.

Job satisfaction – A general attitude toward one's job; the difference between the amount of rewards workers receive and the amount they believe they should receive. *(from [70])*

Motivation – Motivation is defined as the processes that account for an individual's intensity, direction, and persistence of effort toward attaining a goal. *(from [70])*

OR-join – An OR-join is an activity in which a number of alternative workflows reconverge. Unlike an AND-join, however, no synchronisation occurs. In other words, the activity can be performed as soon as just one single condition has been met. *(from [5])*

OR-split – An OR-split is an activity in which a choice is made. During the performance of an OR-split, one workflow is selected from a number of available options. Only the selected flow is initiated by the OR-split. the choice is often based upon the particular attributes of the case in hand. However it may also be a random one. The OR-split is the logical opposite of the OR-join: an OR-split can divide a workflow into a number of alternative streams that later reconverge at an OR-join. There are two types of OR-split tasks: implicit and explicit. the difference between the two is based upon the moment at which the choice is made. *(from [5])*

Parallel routing – Two or more activities related to a specific case may be carried out in parallel if, by definition, the process contains an AND-split and an AND-join. The AND-split allows more than one task to be initiated at the same time. Upon completion, the parallel workflows are resynchronised using the AND-join. *(from [5])*

Perception – A process by which individuals organize and interpret their sensory impressions in order to give meaning to their environment. *(from [70])*

Performance indicator – A performance indicator is a (definition of a) quantity that is used to measure a critical success factor of a process or system. Examples of performance indicators are average flow time, utilisation, and service level. *(from [5])*

Primary process – A process for dealing with customer-oriented cases. The process concentrated upon the delivery of products and/or services to the company's customers. *(from [5])*

Process automation layer – The process automation layer in the workflow system model takes care of the distribution of work items and the automatic and computer supported execution of work items. It consists of the workflow enactment service (workflow engine(s)), client applications (worklists, inboxes), administration and monitoring tools, and invoked applications (interactive and/or fully

automatic). The process automation layer can be divided into two parts: the distribution of work items and the execution of work items.
In comparison with the workflow reference model, the process automation layer consists of all parts and interfaces that are described in the reference model, except for the process definition tool).

Process granularity – The process granularity indicates the size of the individual activities in a process.

Process model – The definition of a process indicates which tasks must be performed – and in what order - to successfully complete a case. In other words, all possible routes are mapped out. A process consists of tasks conditions, and possible further sub processes. By using AND-splits, AND-joins, OR-splits, and OR-j oins, parallel and alternative flows, can be defined. Sub processes also consist of tasks, conditions, and possible further sub processes. The use of sub processes can enable the hierarchical structuring of complex processes. *(from [5])*
It can also be referred to as process definition.

Resource – A resource is a means of production or a group of such means. It may include such actors as people, machines, means of transport, applications, departments, and business units. Resources can only perform certain tasks, and so are grouped into one or more resource classes. The inclusion of a resource in a particular category provides information about the place that a resource has in the organisation or about a particular quality that it has. *(from [5])*

Resource class – Resources can only perform a limited number of activities. In order to make it easy to indicate – when defining a process – which resources can carry out a certain activity, they are grouped into so-called resource classes. One resource may belong to several resource classes. The grouping of resources is in general structured in two ways. First, resources are divided up on the basis of their place within the organisation. This results in resource classes, which are also known as organisational units: for example, "Purchasing Department", "Team A", or "Atlanta Branch". Second, they may be divided up according to functional characteristics – also known as roles. Examples of roles are "Executive C", "Information Analyst", and "Cobol Programmer". Each of these roles corresponds with a resource class. Those categories not based upon a role or an organisational unit are called free resource classes. *(from [5])*

Resource classification – Resources – both staff and automated devices – can only perform a limited number of activities. What these are depends upon such factors as which roles a resource can fulfil and the location where this must be done. A resource classification divides the resources into subsets, also known as resource classes. Examples of resource classification include separation into roles or into organisational units. Resources with the same characteristics under a particular system of classification form a resource class. Some workflow management systems enable the relationships between the resource classes to be illustrated schematically. *(from [5])*

Role – In order to perform activities, skills, are required. Each resource – for example, a person – has certain skills. a role is a collection of complementary skills. It thus becomes possible to identify which role is necessary to perform which activity. Which roles each resource can perform is also indicated. By using roles, it is possible to ensure that activities are assigned to the correct people. In fact, a role is the same as a resource class based upon functional characteristics. *(from [5])*

Routing – The definition of a process determines how cases are routed through the various tasks. Four types of routing are often distinguished: sequencing, selection, parallelization, and iteration. *(from [5])*

Secondary process – A process that supports the primary processes, in particular by providing resources. *(from [5])*

Sequential routing – We refer to sequential activity execution when a number of activities are performed one after the other. When two successive activities are linked by a condition, then they must be performed sequentially. *(from [5])*

Tertiary process – Tertiary processes are those managerial processes that control the primary and secondary processes. *(from [5])*

Values –Values are basic convictions that a specific mode of conduct or end-state of existence is personally or socially preferable to an opposite or converse mode of conduct or end-state of existence. (psychology) *(from [70])*

Work item – A work item is the combination of a case and an activity, which is about to be carried out. A work item is linked to a specific case. The work item disappears at the moment that it begins to be acted upon – the moment that performance of the activity itself starts. It then becomes an activity instance. Note that it is possible, base upon the case's state, to determine which work items are waiting to be handled. *(from [5])*

Workflow –The computerised facilitation or automation of a business process, in a whole or part. *(from [85])*
A workflow comprises cases, resources, and triggers that relate to a particular process. *(from [5])*

Workflow application – A software program(s) that will either completely or partially support the processing of a work item in order to accomplish the objective of an activity instance. (*modified from [85]*)

Workflow definition – A workflow definition consists of the definition of a process (process model), a summary of the resources required and the classification of those resources into classes. *(from [5])*

Workflow definition tool – The tool used to define processes and resource classifications. *(from [5])*

Workflow enactment service – The workflow enactment service is the heart of a workflow management system. It provides the runtime execution environment for supporting workflow processes. It consists of one or more workflow engines. *(from [85])*

Workflow engine – the workflow engine takes care of the actual management of the workflow. Amongst other things, it is concerned with task-assignment generation, resource allocation, activity performance, case preparation and modification, the launching of applications, and the recording of logistical information. *(from [5])*

Workflow management – The term workflow management refers to the ideas, methods, techniques, and software used to support structured business processes. The objective of workflow management is to achieve streamlined and easy-to-maintain work processes. *(from [5])*

Workflow management system – A workflow management system is a software package for the implementation of workflow management in an organisation. It completely defines, manages and executes business processes through the execution of software whose order of execution is driven by a computer representation of the process logic (process model). Such a system maps to the Workflow Management Coalition's Reference model (*modifiedfrom [5] and [85]*)
Note that this term can both refer to a generic software package and to the actual implementation of a package in a specific situation.

Workflow process – A workflow process is a business process that is supported (or can be supported) by a workflow management system. It has the following features: it is a *case-driven* and *make-to-order* process.

Workflow net – A workflow net is a Petri net which represents a workflow process. Such a workflow net has one source place and one sink place. Every node is on a path from the source place to the sink place. A workflow net is sound if, for any case, the procedure will terminate eventually and the moment the procedure terminates there is a token in sink place and all the other places are empty. Moreover there should be no dead transitions; it should be possible to execute an arbitrary task by following the appropriate route through the workflow net. *(from [5])*

Workflow system – A workflow system is the total system including the workflow management system in a specific situation. It includes among others the organisational structure (hierarchical, geographical, business units), the employees, the roles, the process automation layer (workflow enactment service, automatic execution of work items, and invoked applications), and the workflow definition of the specific situation (process model and resource classification). It differs from a workflow management system because it covers much more than the 'technical' workflow management system.

Worklist – A list of work items. *(from [85])*
A worklist can be private or shared. When the worklist is private the workflow engine directly assigns the work items to a resource. When the worklist is shared, a work item is assigned to a pool of resources that can all execute the work item by the workflow engine. Usually this is a group of resources that all perform the same role in the workflow system.

List of variables

In the list below you can find an explanation of the variables and abbreviations that are used in chapter 3.

Variable	**Description**
ρ	The process coupling/cohesion ratio
$\lambda(t)$	The activity relation cohesion
$\mu(t)$	The activity information cohesion
(D, W, O)	An operations structure, with D, W and O as defined in this list
(p, w, cs)	An operation, with p, w, cs as defined in this list
(q, x, ds)	Another operation with, q, x, ds as defined in this list
(t, e)	A valid activity, with t and e as defined in this list
(u, f)	Another valid activity, with u and f as defined in this list
c	The process cohesion
$c(t)$	The activity cohesion
cs	The input information elements of an operation
D	The set of information elements
ds	The input information elements of another operation
e	A resource class
f	Another resource class
F	A partial ordering on T
k	The process coupling
O	The set of operations
p	The output information element of an operation
q	The output information element of another operation
R	The set of relations between information elements
s	An input information element of an operation, $s \in cs$
t	A set of operations
T	A set of valid activities
u	Another set of operations
w	The resource class that is allowed to execute the operation
W	The set of resource classes
x	The resource class that is allowed to execute another operation

List of acronyms

In the list below, you will find the acronyms and corresponding ideas from chapter 4.

[#ITEMS] If possible, show more than one work item on the worklist of an employee, even if a push mechanism is used.

[APPEAR] Give employees the opportunity to adjust the appearance of work items in their worklists to their own preferences: FCFS, earliest due date, random, etc. (Here we assume the assignment of work items is in a pull manner and the worklist is private.)

[BATCH] Offer an employee "batches" of work items. In this way the batch is pushed, but the employee can choose the order of execution of work items within this batch. (Here we assume the work list is private.)

[CASEMAN] Case management: let an employee work on the same case as much as possible.

[HISTORY] Offer a variety in work items to an employee. Remember the kind of work items an employee has executed and decide, based on this history, what kind of new work items will be offered to him or her.

[PREFS] Keep up with the kind of activities an employee likes and make sure he or she will get more of this kind of activities (and less of activities (s)he does not like).

[PRIV PULL] Let an employee choose work items from the private worklist himself/herself: pull mechanism.

[RANDOM] The queuing of work items in the work list should be random.

[RANKING] Make available an employees place within the ranking of good employees (for instance 'hard working', 'producing high quality work').

[REDIRECT] Give employees the possibility to send a work item to another employee, who is better in performing the job, who has more knowledge about the case, who is not busy, etc.

[REJECT] Give employees the possibility to reject a work item (with a valid reason) and return it to the workflow enactment service.

[RELEASE] Release a new work item directly. (Time of notification is upon availability.)

[REPLAN] Do not 're-plan' work items by workflow enactment service.

[RESUBMIT] When a work item has to be performed again after a (negative result of a) check, return it to the same employee to execute it again.

[RESULT] Give each employee authorisation to view the final decision or result of a case in the process.

[ROUTE] Give an employee the possibility to check the progress and route of a case during the process (dynamic aspect).

[SH PULL] Use a shared work list, from which an employee can choose himself/herself: pull-manner.

[STAT MOD] Design a possibility for the employee to examine the static process model in a comprehensible way (static aspect).

[TARGET] Show an employee if he or she works hard enough, if he or she is satisfying the targets.

[TEAM BAT] Create 'team batches' of work items. A team of employees (having the same competences/role) can divide the work according to their own preferences. (Here we assume the allocation mechanism is manual, but is not necessarily controlled by a team leader or manager.) This idea is quite similar to the concept of a self managing team, which is one of the hot items in organisational psychology.

[TEAM WI] Create 'team work items'. Employees (with different competences) have to cooperate to execute an activity.

Bibliography

[1] W.M.P. van der Aalst. The application of Petri nets to workflow management. The Journal of Circuits, Systems an Computers, 8(1): 21-66, 1998.

[2] W.M.P. van der Aalst. On the Automatic Generation of Workflow Processes Based on Product Structures. Computers in Industry, 39(2):97-111, 1999.

[3] W.M.P. van der Aalst. Reengineering Knock-out Processes. Decision Support Systems, 30(4):451-468, 2001.

[4] W.M.P. van der Aalst and P.J.S. Berens. Beyond Workflow Management: Product-Driven Case Handling. In S. Ellis, T. Rodden, and I. Zigurs, editors, International ACM SIGGROUP Conference on Supporting Group Work (GROUP 2001), pages 42-51. ACM Press, New York, 2001.

[5] W.M.P. van der Aalst and K.M. van Hee. Workflow Management: Models, Methods, and Systems. MIT press, Cambridge, MA, 2002.

[6] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns. Distributed and Parallel Databases, 14(1):5-51, 2003.

[7] W.M.P. van der Aalst, M. Stoffele, and J.W.F. Wamelink. Case Handling in Construction. Automation in Construction, 12(3): 303-320, 2003.

[8] R.J. Aldag, A.P. Brief. Task Design and Employee Motivation. Scott, Foresman and company, 1979.

[9] C.M. Axtell, P.E. Waterson, C.W. Clegg. Problems integrating user participation into software development. International Journal of Human-Computer Studies, 47, 323-345, 1997.

[10] L. Baresi, F. Casati, S. Castano, M. Fugini, I. Mirbel, and B. Pernici. WIDE Workflow Development Methodology. Proceedings of International Joint Conference on Work Activities Coordination and Collaboration., pp. 19-28, 1999.

[11] L. Baresi, F. Casati, S. Castano, M. Fugini, P. Grefen, I. Mirbel, B. Pernici, and G. Pozzi. Workflow Design Methodology. In: P. Grefen, B. Pernici, and G. Sanchez (eds.) Database Support for Workflow Management: the WIDE Project, 1999, pages 47-94. Kluwer Academic Publishers.

[12] R. Batenburg, et al. Work and ICT in research. Lemma, Utrecht, 2002. (in Dutch).

[13] A. van den Berg, P. Pottjewijd. Workflow: Continuous Improvement by Integral Process Management. Academic Service, 1997 (page 77). (In Dutch)

[14] M.M. Beyerlein, D. A. Johnson. Advances in interdisciplinary studies of work teams, volume 1, theories of sef-managing work teams. JAI Press inc, 1994.

[15] J.M. Bieman, B-K Kang. Measuring Design-level Cohesion. IEEE Transactions on Software Engineering, 24(2): 111-124, 1998.

[16] G. Burrell, G. Morgan. Sociological Paradigms and Organisatinal Analysis. Heinemann, London 1979.

[17] D.N. Card, V.E. Church, W.W. Agresti. An Empirical Study of Software Design Practices. IEEE Transactions on Software Engineering 12(2): 264-271, 1986.

[18] Cherns. Principles of Sociotechnical Design Revisited. Human Relations, 40(3), pp. 153-162, 1987.

[19] W. Clegg. Sociotechnical principles for system design. Applied Ergonomics. 31, pp. 463-477, 2000.

[20] C.W. Clegg, M. Older Gray, P. E. Waterson. The "Charge of the Byte Brigade" and a socio-technical response. International Journal of Human Computer Studies, 52, pp. 235-251, 2000.

[21] S.D. Conte, H.E. Dunsmore, V.Y. Shen. Software Engineering Metrics and Models. Benjamin/Cummings Publishing Company, Inc., 1986.

[22] S. Dhondt, K. Kraan. Work in information society. Lemma, Utrecht, 2001. (in Dutch).

[23] A. Dvash, B. Mannheim. Technological coupling, job characteristics and operators'well-being as moderated by desirability and control. Behaviour and information technology, 20(3), pp. 225-236, 2001.

[24] T.J. Emerson. A Discriminant Metric for Module Cohesion. Proceedings of the 7th International Conference on Software Engineering (ICSE-7), pages 294-303, 1984.

[25] N. Fenton, A. Melton. Deriving Structurally Based Software Measures. Journal of Systems and Software, 12: 177-187, 1990.

[26] Y. Fried. Meta-Analytical Comparison of the Job Diagnostic Survey and Job Chracteristcs Inventory as Correlates of Work Satisfaction and Performance. Journal of Applied Psychology, 76 (5), pages 690-697, 1991.

[27] J. Gulliksen, B. Göransson, I. Boivie, S. Blomkvist, J. Persson, Å. Cajander. Key principles for user-centered systems design. Behaviour & Information Technology, 22(6), pp. 397-409, 2003.

[28] J.R. Hackman, G.R. Oldham. Development of the Job Diagnostic Survey. Journal of Applied Psychology, 60, pp. 159-170, 1975.

[29] J.R. Hackman, G.R. Oldham. Motivation through the design of work: test of a theory. Organizational Behavior and Human Performance, 15, pp. 250-279, 1976.

[30] J.R. Hackman, J.L. Suttle. Improving Life at Work: Behavioral Science Approaches to Organizational Change. Goodyear Publishing, 1977.

[31] J.R. Hackman, M.D. Lee. Redesigning Work: a Strategy for change. Studies in Productivity, Work in America Institute, 1979.

[32] T. Herrmann, M. Hoffmann, K.-U. Loser. Sozio-orientierte und semi-strukturierte Modellierung mit SeeMe. In: Proceedings der Fachtagung MobIS'98. Informationssystem Architekturen. Rundbrief des GI-Fachausschusses 5.2. 5. Jg., Heft 2. pp. 15-22. (In German).

[33] T. Herrmann, M. Hoffmann. Augmenting Self-Controlled Work Allocation in Workflow-Management-Applications. Proceedings of HCI '99, pp. 288-292, 1999.

[34] T. Herrmann. Evolving Workflows by User-driven Coordination. In: R. Reichwald, J. Schlichter (Eds.): Verteiltes Arbeiten – Arbeiten der Zukunft. Tagungsband D-CSCW. pp. 103-114, 2000.

[35] F. Herzberg. The managerial choice: to be efficient and to be human. Jones-Irwin, 1976.

[36] M. Hoffmann, K.-U. Loser. Mitarbeiter-orientierte Modellierung und Planung von Geschäftsprozessen bie der Einführung von Workflow-Management. Proceedings of EMISA-Fachgruppentreffens 1997, pp. 39-57. (In German)

[37] D. Holman, C. Clegg, P. Waterson. Navigating the territory of job design. Applied Ergonomics, 33. pp. 197-205, 2002.

[38] W. Hur, H. Bae, S.-H. Kang. Customizable Workflow Monitoring. Concurrent Engineering: Research and Applications. 11 (4), pp. 313-325, 2003.

[39] M.T. Iaffaldano, P.M. Muchinsky. Job Satisfaction and Job Performance: A Meta-Analysis. Psychological Bulletin, 97(2), pp. 251-273, 1985.

[40] IDS Scheer. ARIS Process Performance Manager. Presentation, http://www.ids-scheer.de/PPM/ , 2004.

[41] IDS Scheer. ARIS Process Performance Manager (ARIS PPM). Whitepaper, http://www.ids-scheer.de/PPM/ , 2004.

[42] K. Joshi, A. Rai. Impact of the quality of information products on information system users' job satisfaction: an empirical investigation. Information Systems Journal, 10, pp. 323-345, 2000.

[43] T.A. Judge, C.J. Thoresen, J.E. Bono, G.K. Patton. The Job Satisfaction – Job Performance Relationship: A Qualitative and Quantitative Review. Psychological Bulletin, 127 (3), pages 376-407, 2001.

[44] T.A. Judge, D. Heller, M.K. Mount. Five-Factor Model of Personality and Job Satisfaction: A Meta-Analysis. Journal of Applied Psychology, 87 (3), pages 530-541, 2002.

[45] T.A. Judge, R. Ilies. Relationship of Personality to Performance Motivation.: A Meta-Analytical Review. Journal of Applied Psychology, 87 (4), pages 797-807, 2002.

[46] B-K. Kang, J.M. Bieman. Using Design Cohesion to Visualize, Quantify, and Restructure Software. 8th International Conference on Software Engineering and Knowledge Engineering, Knowledge Systems Institute, Skokie IL, pages 222-229, 1996.

[47] B-K. Kang, J.M. Bieman. A Quantitative Framework for Software Restructuring. Journal of Software Maintenance, 11: 245-284, 1999.

[48] W.J. Kettinger, J.T.C. Teng, S. Guha. Business Process Change: A Study of Methodologies, Techniques, and Tools. MIS Quarterly, 21(1), pp. 55-80, 1997.

[49] P. Kueng. The effects of workflow systems on organizations: a qualitative study. In: W. van der Aalst et al (eds.), Business Process Management, LNCS 1806, pp. 301-316, 2000.

[50] A. Lakhotia. Rule-based Approach to Computing Module Cohesion. Proceedings of the 15th International Conference on Software Engineering, pages 35-44, 1993.

[51] Ley GmbH. COSA 4, Business-Process Designer's Guide, 2003.

[52] Ley GmbH. COSA 4, User's Guide, 2002.

[53] H. van Mierlo. Self-managing teamwork and psychological well-being. PhD-dissertation, Technische Universiteit Eindhoven, 2003.

[54] M. zur Muehlen. Organizational Management in Workflow Applications – Issues and Perspectives. Information Technology and Management Journal 5(3), pp. 271-291, 2004.

[55] G.J. Myers. Composite/Structured Design. Van Nostrand Reinhold, New York NY, 1978.

[56] W.P. Neumann, S. Kihlberg, P. Medbo, S.E. Mathiassen, J. Winkel. A case study evaluating the ergonomic and productivity impacts of partial automation strategies in the electronics industry. International Journal of Production Research, 40(16), pp. 4059-4075, 2002.

[57] A. Orlicky. Structuring the Bill of Materials for MRP. Production and Inventory Management, December, pp.19-42, 1972.

[58] R.J. van Ouwerkerk, T.F. Meijman, G. Mulder. Industrial Psychological Task Analysis. Lemma, Utrecht, 1994. (In Dutch).

[59] Pallas Athena. Administrator Guide FLOWer 3.0. Pallas Athena, the Netherlands, 2004

[60] Pallas Athena. Designer's Buide FLOWer 3.0. Pallas Athena, the Netherlands, 2004.

[61] Pallas Athena. User Guide FLOWer 3.0. Pallas Athena, the Netherlands, 2004.

[62] M.M. Petty, G.W. Mcgee, J.W. Cavender. A Meta-Analysis of the Relationships Between Individual Job Satisfacton and Individual Performance. Academy of Management Review, 9 (4), pages 712-721, 1984.

[63] E.A.H. Platier. A Logistical View on Business Processes: Concepts for Business Process Redesign and Workflow Management. PhD thesis, Eindhoven University of Technology, Eindhoven, 1996.

[64] H.A. Reijers, S. Limam, W.M.P. van der Aalst. Product-Based Workflow Design. Journal of Management. Information Systems, 20(1): 229-262, 2003.

[65] H.A. Reijers. Design and Control of Workflow Processes: Business Process Management for the Service Industry. Lecture Notes in Computer Science 2617. Springer-Verlag, Berlin, 2003.

[66] H.A. Reijers. A Cohesion Metric for the Definition of Activities in a Workflow Process. Proceedings of the Eighth CAiSE/IFIP8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design 2003, pages 116-125, 2003.

[67] H.A. Reijers and W.M.P. van der Aalst. Formal methods in Business Process Management. Informatie, 45(5): 50-53, 2003. (In Dutch)

[68] H.A. Reijers, J. Rigter, and W.M.P. van der Aalst. The Case Handling Case. International Journal of Cooperative Information Systems, 12(3):365-391, 2003.

[69] H.A. Reijers, I.T.P. Vanderfeesten. Cohesion and Coupling Metrics for Workflow Process Design. Proceedings of the 2nd International Conference on Business Process Management, pages 290-305, 2004.

[70] S.P. Robbins. Organizational behavior. Prentice Hall, New Jersey, 2001.

[71] R.O. Rupp, J.R. Russell. The Golden Rules of Process Redesign. Quality Progress, 27(12): 85-92, 1994.

[72] F.E. Saal, P.A. Knight. Industrial/Organizational Psychology. Brooks/Cole Publishing Company, California, 1995.

[73] A. Seidmann and A. Sundararajan. The Effects of Task and Information Asymmetry on Business Process Redesign. International Journal of Production Economics, 50(2-3): 117-128, 1997.

[74] R.W. Selby, V.R. Basili. Analyzing Error-Prone System Structure. IEEE Transactions on Software Engineering, 17(2):141-152, 1991.

[75] M. Shepperd. Software Engineering Metrics Volume I: Metrics and Validations. McGraw-Hill, 1993.
[76] F. Simon, S. Löffler, C. Lewerentz. Distance Based Cohesion Measuring. Proceedings of the 2nd European Software Measurement Conference (FESMA) 1999, pages 69-83, 1999.
[77] L.U. de Sitter. Synergetic production. Van Gorcum, Assen, 1998. (In Dutch).
[78] B. Smith, P. Caputi, N. Crittenden, R. Jayasuriya, P. Rawstorne. A review of the construct of computer experience. Computers in Human Behavior, 15, pp. 227-242, 1999.
[79] Staffware. Staffware Process Suite, Defining Staffware Procedures. Issue 2, 2002.
[80] Staffware. Staffware Process Suite, Using the Staffware Process Client. Issue 2, 2002
[81] W.P. Stevens, G.J. Myers, L.L. Constantine. Structured design. IBM Systems Journal, 13(2): 115-139, 1974.
[82] D.A. Troy, S.H. Zweben. Measuring the Quality of Structured Designs. Journal of Systems and Software, 2: 113-120, 1981.
[83] A.N. Turner, P.R. Lawrence. Industrial Jobs and the Worker. Harvard University Press, 1965.
[84] P.E. Waterson, M.T. Older Gray, C.W. Clegg. A sociotechnical method for Designing Work Systems. Human Factors. 44(3), pp. 376-391, 2002.
[85] Workflow Management Coalition. The Workflow Reference Model (WFMC-TC-1003). 1995.
[86] Workflow Management Coalition. Terminology and Glossary (WFMC-TC-1011). 1999.
[87] Workflow Management Coalition. WFMC Home Page. http://www.wfmc.org .
[88] T.A. Wright, R. Cropanzano. Psychological well-being and job satisfaction as predictors of job performance. Journal of Occupational Health Psychology, 5 (1), pages 84-94, 2000.
[89] M. Xenos, D. Stavrinoudis, K. Zikouli, D. Christodoulakis. Object-oriented Metrics – A Survey. Proceedings of the Federation of European Software Measurement Association (FESMA) 2000, pages 1-10, 2000.

Appendix

A: The difference between BPR and WfM

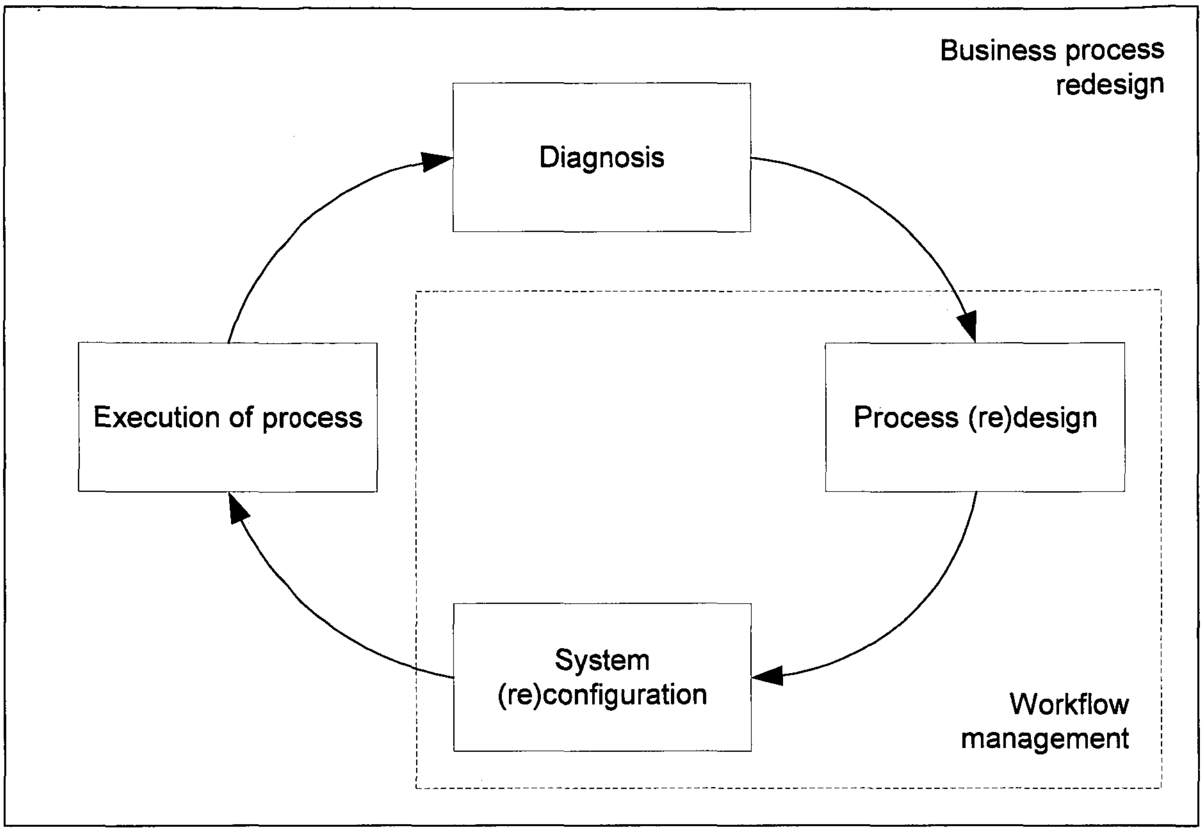


Figure 14: The BPR-lifecycle (from [5] and [67]).

The "Business Process Redesign" process is an ever ongoing process to improve the business processes of a company. This process can be represented by a lifecycle containing four phases ([5]):

- **Diagnosis**
  This phase begins with an analysis of the current situation, and in particular of the problem caused by the existing way of working.
- **Process (re)design**
  In this phase a new design of the process is developed, to overcome the problems determined in the first phase.
- **System (re)configuration**
  During this phase, a new system is created to support the processes previously identified.
- **Execution of the process**
  In this phase the process is operational. Again the performance of the process is measured and assessed, possibly leading to another run of the lifecycle.

Typically with this lifecycle the difference between BPR and workflow management can be explained. While BPR focuses on the whole change and improvement process, workflow management only supports the bottommost part of the cycle.

B: Categories of processes

The three kinds of business process that can be distinguished in literature are ([5]):

- **Primary processes** – primary processes are those that produce the company's products or services. They therefore are known also as production processes. They deal with cases for the customer. As a rule, they are the processes that generate income for the company, and are clearly customer-oriented.
- **Secondary processes** – secondary processes are those that support the primary ones. They therefore are also known as support processes.
- **Tertiary processes** – tertiary processes are the managerial processes that direct and coordinate the primary and secondary processes. During these, the objectives and preconditions within which the managers of the other processes must operate are formulated, and the resources required to carry out the other processes are allocated.

C: Description of the information elements of Figure 7

Information element number	**Description**
1	Last name applicant
2	First name(s) applicant
3	Address applicant
4	Telephone number(s) applicant
5	Sex applicant
6	Social Security Number applicant
7	Country where applicant lives
8	Birth place of applicant
9	Account number applicant
10	Identification number of applicant (student grant institute)
11	Signature applicant
12	First day of study
13	Date from which applicant wants to receive a scholarship
14	Signature father of applicant
15	Signature mother of applicant
16	Identification number of father of applicant (student grant institute)
17	Identification number of mother of applicant (student grant institute)
18	Period or year for which the applicant requests a student grant
19	Date of request
20	Birth date of applicant
21	Nationality of applicant
22	The kind of student grant the applicant requests (There are four possibilities for its value: (1) a basic scholarship (2) basic scholarship and supplementary scholarship (3) basic scholarship, supplementary scholarship and loan (4) basic scholarship and loan)
23	Age of applicant
24	Social Security Number father of applicant
25	Reference year for tax authorities
26	Social Security Number mother of applicant
27	Applicant has a right to receive a student grant
28	Income of father of applicant
29	Income of mother of applicant
30	Income of applicant's parents
31	Applicant has a right to receive a supplementary scholarship
32	Kind of health insurance of applicant
33	Living situation of applicant
34	Applicant has a right to receive a loan from the student grant institute
35	Maximum amount that can be received for supplementary scholarship
36	Parental contribution
37	The amount of loan the applicant requests
38	Maximum amount of loan
39	The amount of supplementary scholarship that is assigned to applicant
40	The amount of basic scholarship that is assigned to applicant
41	The amount of loan that is assigned to applicant
42	The total amount of student grant that is assigned to applicant

Table 9: Description of information elements of the information element structure in Figure 7.

D: Transitivity, reflexivity and symmetry of relations

A relationship R on a set X can have to following characteristics:

Reflexive	$\forall s \in X\left(R(s,s)\right)$
Symmetric	$\forall s,t \in X\left(R(s,t) \Rightarrow R(t,s)\right)$
Transitive	$\forall_{s,t,u \in X}\left(R(s,t) \wedge R(t,u) \Rightarrow R(s,u)\right)$

E: Alternative designs for the "request for student grant" process

E.1 The first alternative process design: smaller activities

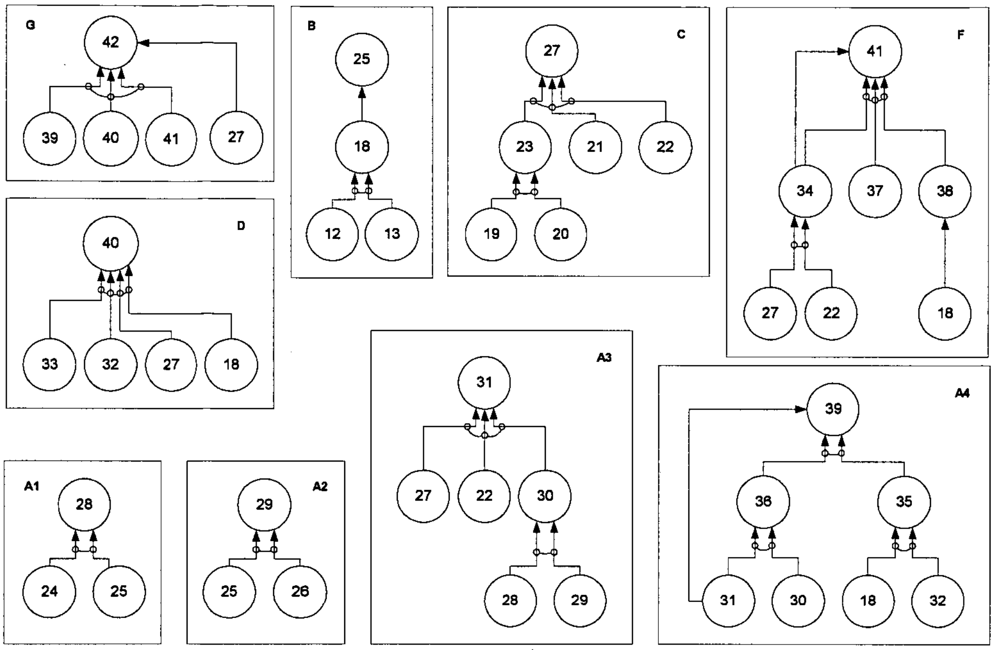


Figure 15: The partitioning of the information element structure in smaller activities. Compared to the original structure activities A and E are split up.

Activity	**Activity relation cohesion**	**Activity information cohesion**	**Activity cohesion**
A1	0	0	0
A2	0	0	0
A3	1	0.167	0.167
A4	0.5	0.429	0.214
B	1	0.25	0.25
C	1	0.167	0.167
D	0	0	0
F	0.5	0.286	0.143
G	0	0	0

Table 10: The activity relation cohesion, the activity information cohesion and the activity cohesion for all activities in the first alternative process design.

E.2 The second alternative design: larger activities

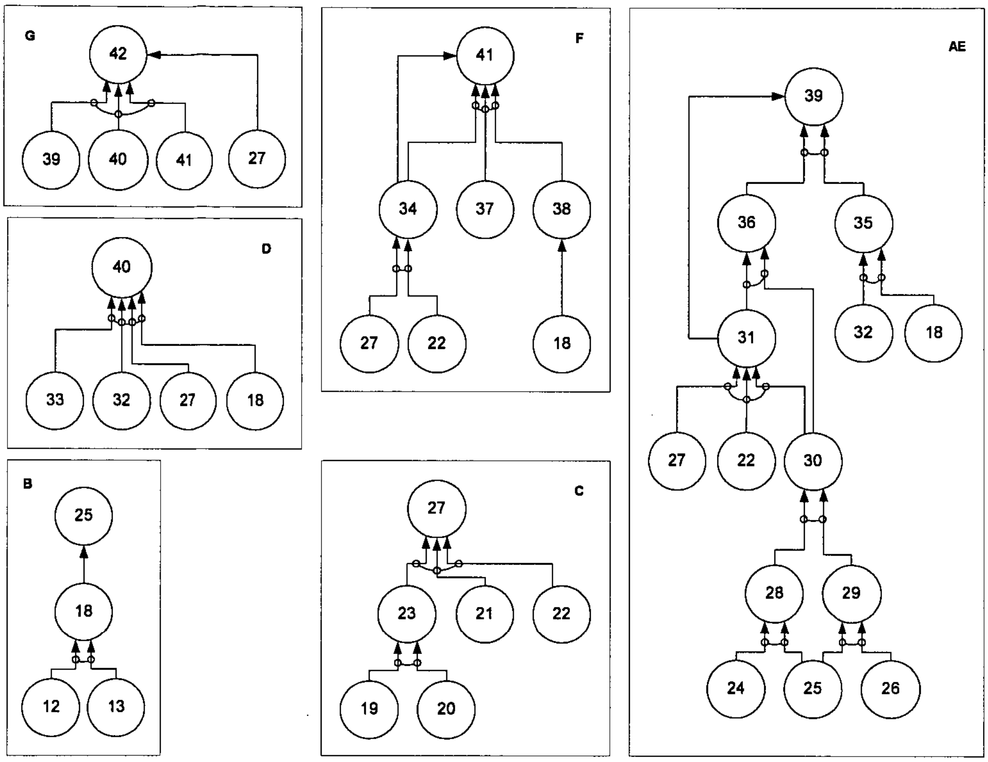


Figure 16: The partitioning of the information element structure in larger activities. Compared to the original structure activities A and E are merged.

Activity	Activity relation cohesion	Activity information cohesion	Activity cohesion
AE	0.356	0.5	0.179
B	1	0.25	0.25
C	1	0.167	0.167
D	0	0	0
F	0.5	0.286	0.143
G	0	0	0

Table 11: The activity relation cohesion, the activity information cohesion and the activity cohesion for all activities in the second alternative process design.

F: Construction permits process

F.1 Description of the information elements

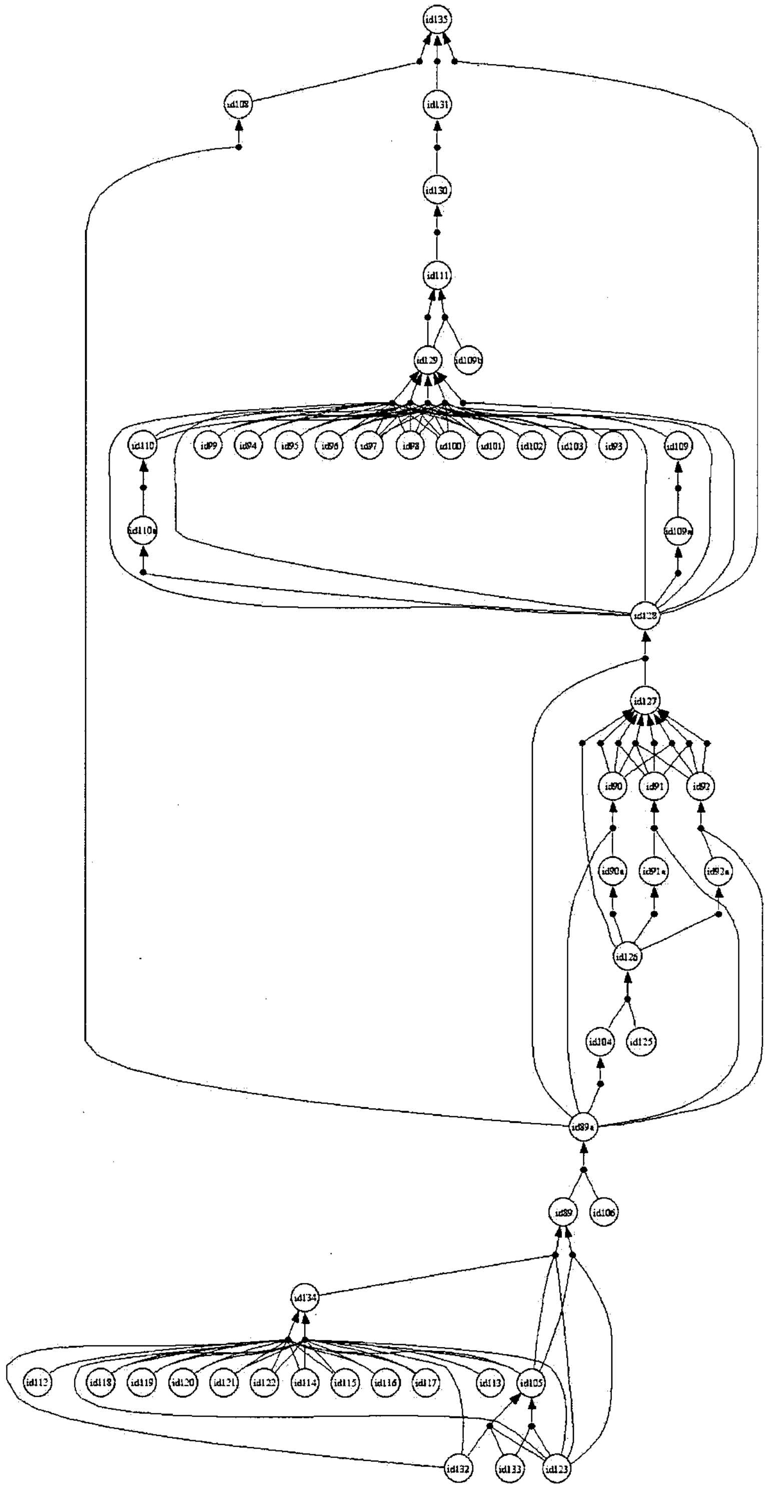


Figure 17: Information element structure for the construction permits process

Below you will find a list with the description of the information elements belonging to the construction permits process of section 3.6.2.. The list is in Dutch because building industry jargon is used, which is difficult to translate correctly, without having knowledge on this jargon.

1. Naam en voorletters aanvrager
2. Geslacht aanvrager
3. Vertegenwoordiger rechtspersoon (indien bedrijf)
4. Correspondentieadres in Nederland
5. Postcode en plaats
6. Telefoon overdag
7. Faxnummer
8. E-mailadres
9. U bent eigenaar / huurder / anders, nl
10. Welke bouwvergunning wordt aangevraagd? Keuze uit:
    a. Lichte bouwvergunning
    b. Reguliere bouwvergunning
    c. Reguliere bouwvergunning fase 1
    d. Reguliere bouwvergunning fase 2
11. Indien 10d: Datum afgifte bouwvergunning fase 1
12. indien 10d: (registratie)nummer bouwvergunning fase 1
13. Al eerder een bouwvergunning aangevraagd? (ja/nee)
14. Datum verlening/weigering eerdere bouwvergunning
15. Registratienummer eerdere aanvraag om bouwvergunning
16. Straat en huisnummer bouwwerk/perceel
17. Postcode en plaats bouwwerk/perceel
18. Kadastrale aanduiding: gemeente
19. Kadastrale aanduiding: Sectie en nummer
20. Eigendomssituatie perceel/kavel. Keuze uit:
    a. Eigen grond
    b. erfpacht
    c. huur
21. Omschrijving van het bouwplan a. Geheel. b. Gedeeltelijk:
    i. Plaatsen
    ii. Vernieuwen
    iii. Veranderen
    iv. Oprichten
    v. vergroten

    van....?
22. Toelichting op 21
23. Gaat het om een tijdelijk bouwwerk? Ja/Nee
24. indien 23 ja: wat is de beoogde instandhoudingtermijn van het bouwwerk?
25. huidige gebruik van het bouwwerk en de bijbehorende terreinen
26. gebruik van het bouwwerk en de bijbehorende terreinen na uitvoering van de werkzaamheden
27. Gebruiksfunctie, bezettingsgraadklasse, de gebruiksoppervlakte en de vloeroppervlakte, na uitvoering van de werkzaamheden
28. Voor hoeveel woningen/wooneenheden vraagt u de bouwvergunning aan?
    a. Aantal huurwoningen
    b. Aantal koopwoningen
    c. Aantal huurwooneenheden
    d. Aantal koopwooneenheden
29. Bent u na voltooiing van de werkzaamheden bewoner van het bouwwerk? (ja/nee)
30. Verandert de onbebouwde oppervlakte van het terrein door de bouwwerkzaamheden?
    a. Ja . Bebouwde oppervlakte voor uitvoering van de werkzaamheden
        1. Bebouwde oppervlakte na uitvoering van de werkzaamheden
    b. Nee

31. Verandert de bruto vloeroppervlakte van het bouwwerk door de bouwwerkzaamheden?
    a. Ja . Bebouwde oppervlakte voor uitvoering van de werkzaamheden
        1. Bebouwde oppervlakte na uitvoering van de werkzaamheden
    b. Nee
32. Verandert de bruto inhoud van het bouwwerk door de bouwwerkzaamheden?
    a. Ja . Bebouwde oppervlakte voor uitvoering van de werkzaamheden
        1. Bebouwde oppervlakte na uitvoering van de werkzaamheden
    b. Nee
33. Materiaal en kleurgebruik
34. Wilt u het bouwplan mondeling toelichten voor de welstandscommissie/stadsbouwmeester?
35. Aanneemsom of raming van de kosten (exclusief BTW)
36. Hoe worden de bouwwerkzaamheden uitgevoerd?
    a. U huurt een hoofdaannemer in die eindverantwoordelijke is voor de bouwwerkzaamheden
    b. U huurt zelf verschillende aannemers in voor de uitvoering van de verschillende bouwwerkzaamheden
    c. U voert de bouwwerkzaamheden geheel zelf uit
    d. Nog niet bekend,
37. Gegevens hoofdaannemer: (naam en adres)
38. Gegevens onderaannemer: (naam, anders, Voor welk deel van de werkzaamheden wordt de onderaannemer ingeschakeld?)
39. Heeft u voor de bouwwerkzaamheden ook de volgende vergunningen nodig?
    a. Nee
    b. ja, nl:
        i. Monumentenvergunning
        ii. Vergunning Kernenergiewet
        iii. Milieuvergunning
        iv. Vergunning Wet ziekenhuisvoorzieningen
        v. Sloopvergunning
40. Laat u de aanvraag voor de bouwvergunning door een gemachtigde verzorgen?
    a. Nee
    b. ja,
        i. Naam en voorletters
        ii. Functie
        iii. Correspondentieadres in Nederland
        iv. Postcode en plaats
        v. Telefoon overdag
        vi. Faxnummer
        vii. E-mailadres
41. Aanvrager: datum en handtekening
42. Gemachtigde: datum en handtekening
43. Kwaliteitsverklaringen, gelijkwaardigheidsverklaringen e.d.
44. Overzichtslijst / verklaring digitaal ingediende gegevens en bescheiden
45. Plattegronden en doorsnedetekeningen
46. Aanduiding bestemmingen
47. Afmetingen perceel en situering
48. Hoogte bouwwerk t.o.v. straatpeil en aantal bouwlagen
49. Inrichting parkeervoorzieningen
50. Indieningsvereisten Agrarische Adviescommissie
51. Indieningsvereisten toetsing leefmilieuverordening
52. Geveltekeningen en belendende bebouwing
53. Detailtekeningen gezichtsbepalende bouwdelen
54. Foto's bestaande situatie en omliggende bebouwing
55. Indieningsvereisten vrijstelling bestemmingsplan
56. Belastingen en belastingcombinaties constructie

57. De uiterste grenstoestand van de bouwconstructie
58. Gegevens en bescheiden Wet Bibob
59. EPC-berekening thermische eigenschappen en luchtdoorlatendheid
60. Geluidwering uitwendige scheidingsconstructie en geluidabsorptie
61. Daglichttoetreding
62. Ventilatievoorzieningen, verbrandingsgassen en verbrandingslucht
63. Brandveiligheid en rookproductie
64. Brand- en rookcompartimentering
65. Vluchtroutes en brandveiligheidsvoorzieningen
66. Noodstroomvoorziening en –verlichting
67. Wateropname materialen vloer, wand en plafonds in sanitaire ruimten
68. Lucht- en waterdichtheid en vochtwerende voorzieningen
69. Riolering en hemelwaterafvoeren
70. Gas-, elektra- en waterleiding, inclusief aansluitpunten
71. Drinkwater- en warmwatervoorzieningen
72. Inbraakwerendheid
73. Weren van ratten en muizen
74. Gebruiksfunctie en afmetingen van ruimten
75. Opslagplaats afvalstoffen
76. Opslagplaats gevaarlijke stoffen
77. Stallingsruimte voor fietsen
78. Integrale toegankelijkheid en toegankelijkheid van ruimten
79. Trappen, hellingbanen en vloerafscheidingen
80. Opstelplaats aanrecht, kook-/stooktoestel en warmwatervoorziening
81. Aanduiding bad- en toiletruimte, meterruimte, liften en liftschachten
82. Hoogteligging vloeren t.o.v. het aansluitende terrein
83. Draairichting draaiende delen
84. Bouwveiligheidsplan en toegankelijkheid bouwplaats
85. Brandveiligheidinstallaties
86. Bluswatervoorzieningen en opstelplaatsen van brandweervoertuigen
87. Rapportage bodemgesteldheid
88. Indieningsvereisten experimentele bouw
89. Dossier
90. Advies CCT
91. Advies Brandweer
92. Advies Milieu
93. Archiefstukken
94. Bestemmingsplan
95. Bouwbesluit
96. Bouwverordening
97. Woningwet
98. Kadastrale gegevens
99. Gegevens burgerzaken
100. Gegevens educatie en welzijn
101. Gegevens monumentencommissie
102. Gegevens rijksdienst monumenten
103. Gegevens provincie m.b.t. vrijstellingen
104. Categorisering bouwaanvraag. Keuze uit

Aangebouwd bijgebouw
agrarische bedrijfsruimte
bedrijfsruimte
bouwwerk geen gebouw zijnde
dakkapel
erfafscheiding

kantoor
uitbreiding woonhuis
veranderen woonhuis
vrijstaand bijgebouw
woonhuis

105. Digitaal document
106. Checklist bij het aanvraagformulier
107. (Ontvangstbevestiging)
108. Verzoek aanvullende stukken
109. Advies jurist
110. Advies Welstand
111. Beschikking
112. sofi-nummer
113. KvK-nummer
114. domein
115. dossiertype
116. Ontvangstdatum
117. scandatum
118. documentdatum
119. Uw kenmerk
120. omschrijving
121. trefwoorden
122. prioriteit
123. Is aanvraagformulier aanwezig?
124. ()
125. adviestabel

Categorie	Advies Milieu?	Advies Brandweer?	Advies CCT
Aangebouwd bijgebouw	x		x
agrarische bedrijfsruimte	x	x	x
bedrijfsruimte	x	x	x
bouwwerk geen gebouw zijnde	x		x
dakkapel			
erfafscheiding			x
kantoor	x	x	x
uitbreiding woonhuis	x		x
veranderen woonhuis	x		x
vrijstaand bijgebouw	x		x
woonhuis	x		x

126. advies instanties (combinatie van CCT, Milieu en Brandweer)
127. adviezen
128. ontvankelijk dossier
129. beoordeling en dossier
130. ondertekende beschikking
131. bouwvergunning (ja/nee)
132. aanvraagformulier (bestaande uit informatie elementen 1 t/m 42)
133. aanvullende stukken (bestaande uit informatie elementen 432 t/m 88)
134. kenmerken dossier

F.2 Process models for the original process and the redesigned process

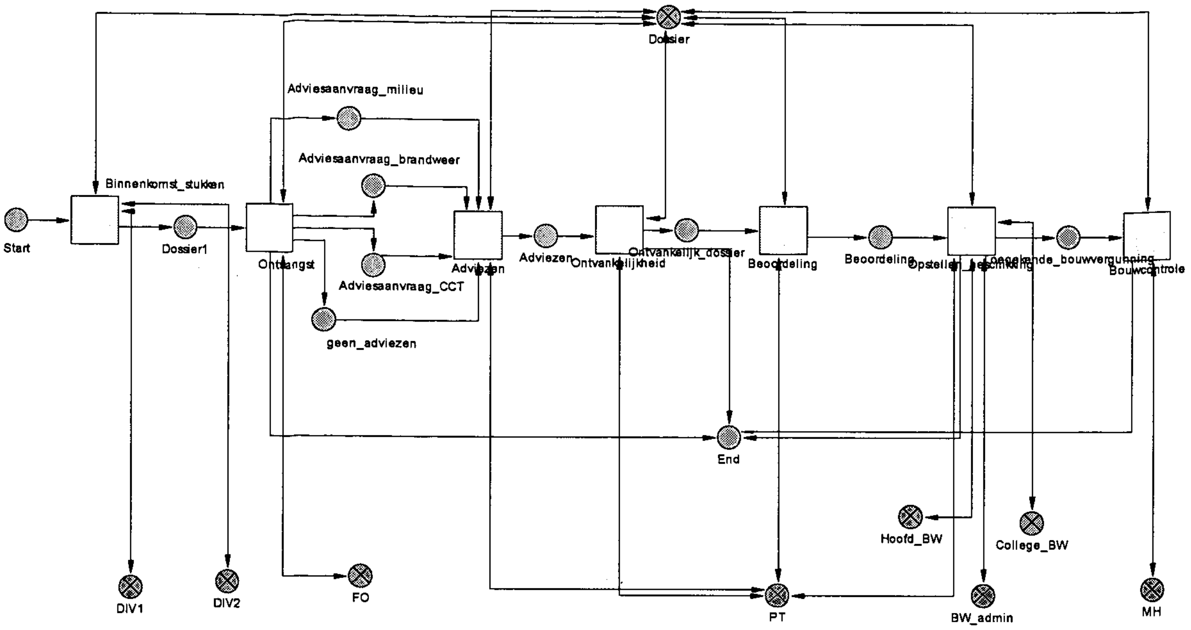


Figure 18: ExSpect model of the original main process

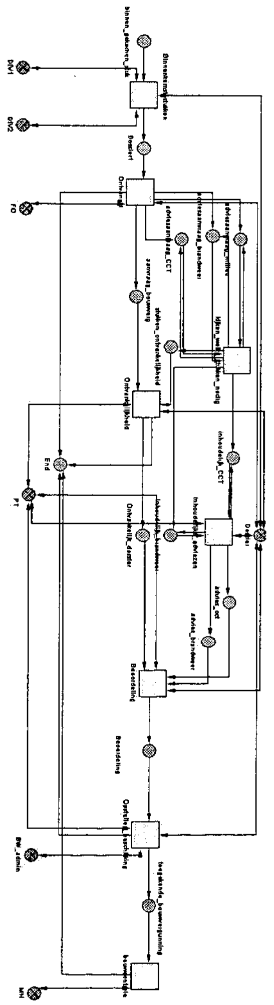


Figure 19: The ExSpect model of the redesigned process

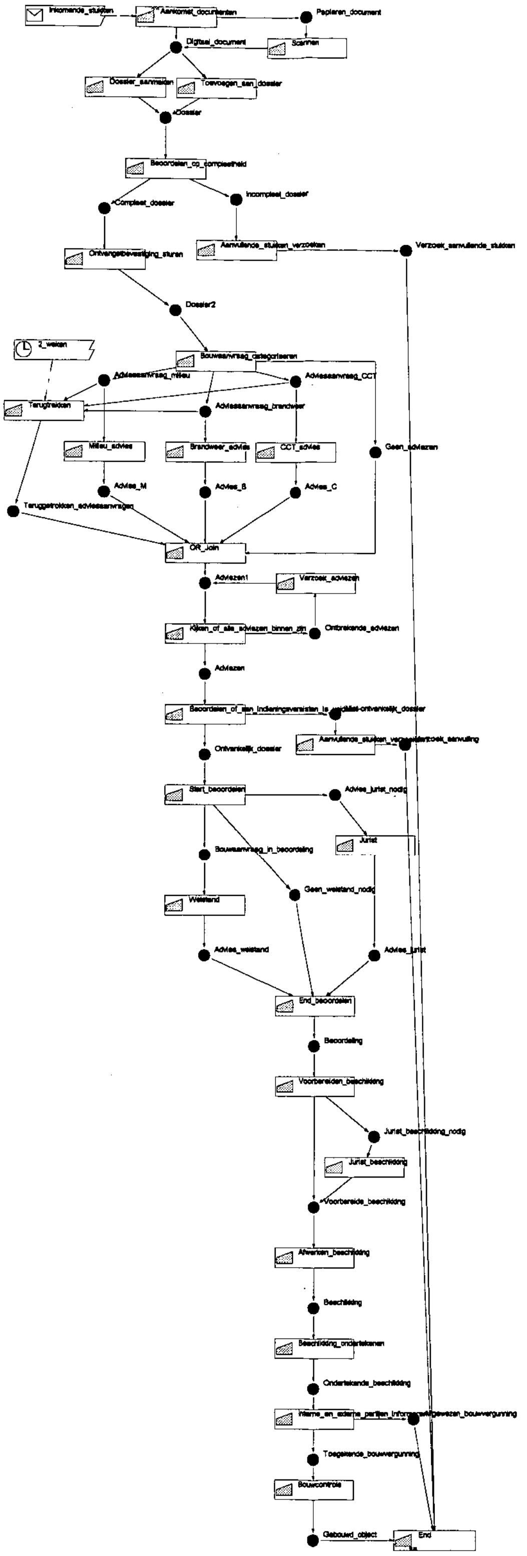


Figure 20: The Protos process model for the original process

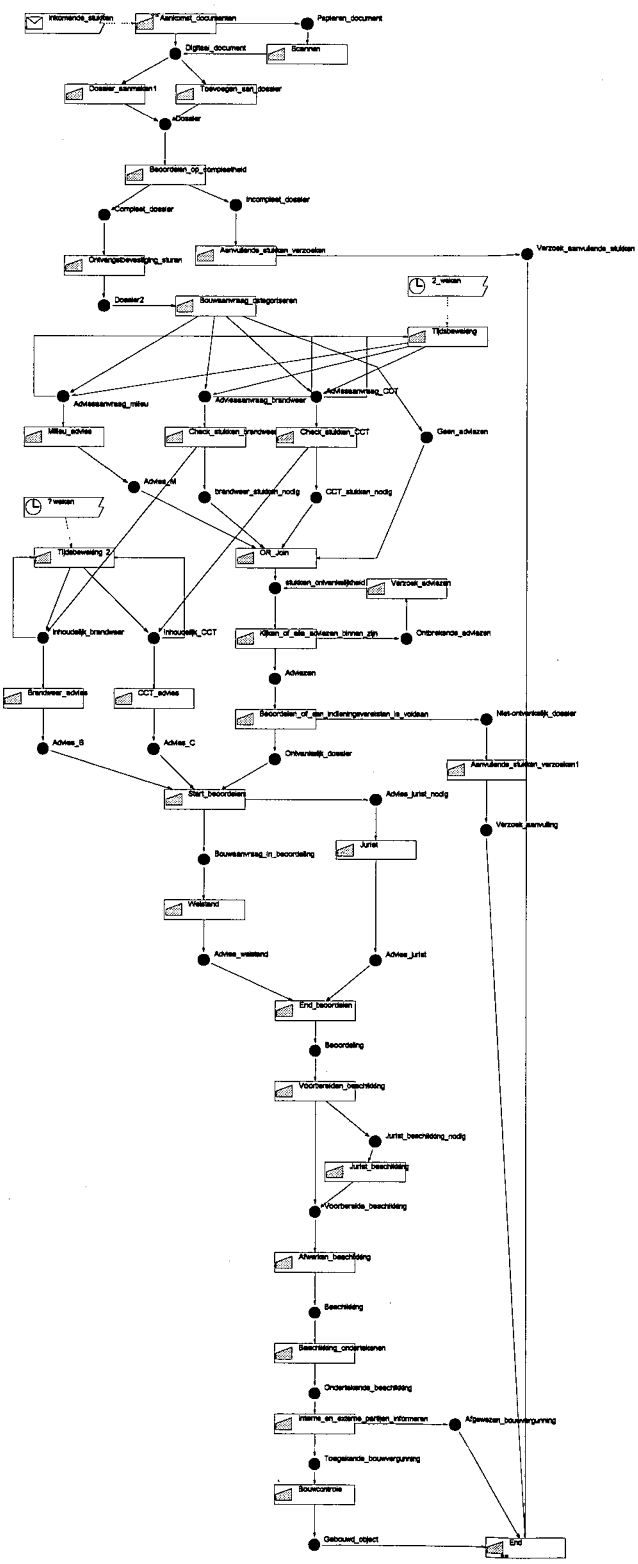


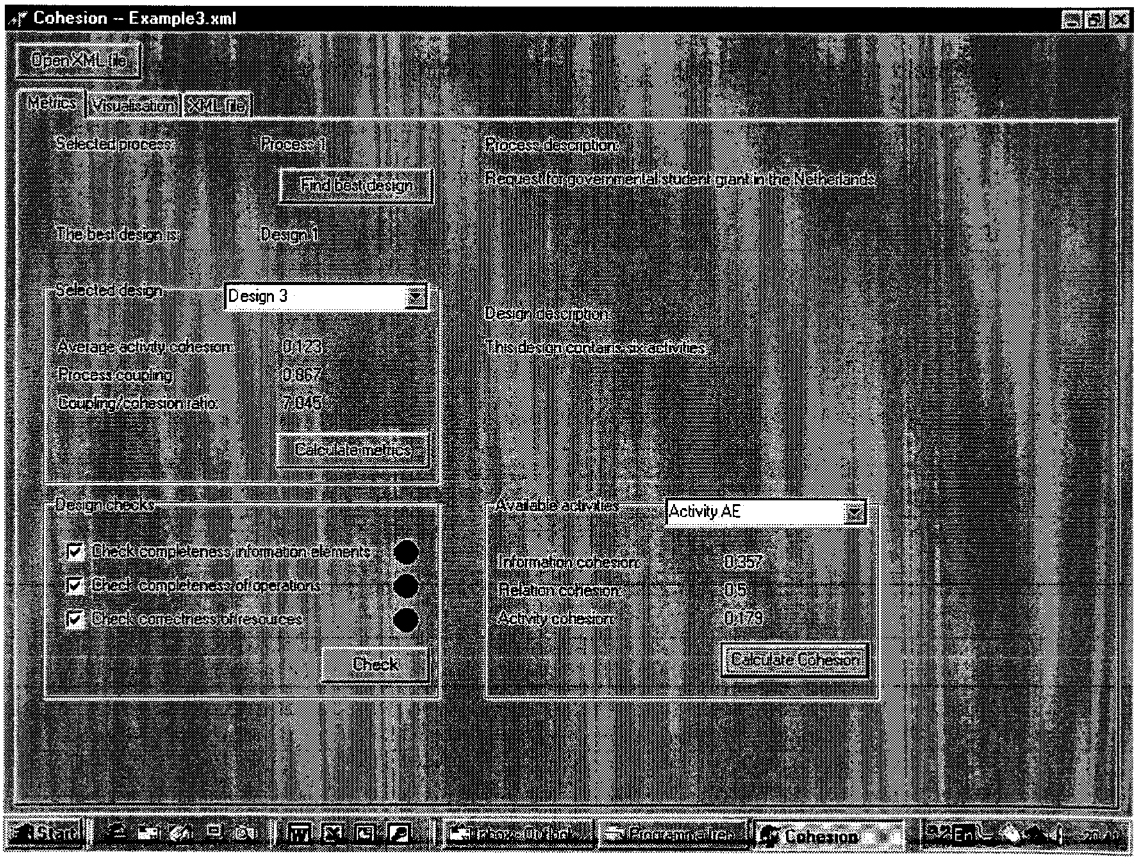
Figure 21: The Protos process model for the redesigned process

G: Tool description

To support the calculation of the metrics and the execution of the heuristic for a certain process design, we developed a software tool. In this part of the appendix, the tool will be described. First the user interface of the program will be explained. Next the predefined format of the input files will be presented and finally we will give some examples of input files, by showing the files used to elaborate on the case studies.

G.1 User interface

In this section we will explain the use of the software program that supports the calculation of the coupling and cohesion metrics for workflow process design.
Besides the button to open an XML-file, the program consists of three sheets: the metric-sheet, the visualisation sheet and the XML-file-sheet, which will be explained below.



Cohesion -- Example3.xml

Selected process: Process 1

Process description:
Request for governmental student grant in the Netherlands

The best design is: Design 1

Selected design: Design 3

Average activity cohesion: 0.123
Process coupling: 0.867
Coupling/cohesion ratio: 7.045

Design description:
This design contains six activities

☑ Check completeness information elements ●
☑ Check completeness of operations ●
☑ Check correctness of resources ●

Available activities: Activity AE

Information cohesion: 0.357
Relation cohesion: 0.5
Activity cohesion: 0.179

Figure 22: The metric sheet of the tool

The metric sheet

In the metric sheet it is possible to calculate metrics of the process and the designs. In the upper part you can find some information (name and description) about the process (from the XML-file). It is also possible to directly find the best design of the process, by clicking the "Find best design"-button. The metric-sheet then contains several group boxes with different topics:

- Selected design – The first group box is the "selected design"-box. In this box you can select one of the designs that is defined in the XML-file, and calculate all metrics that refer to designs (average activity cohesion, process coupling, and coupling/cohesion ratio of a design). On the right you can see a description of the design, if the XML-file is provided with this description.

- Design checks – The group box on the bottom left, is the "design checks"-box. This box contains some checks on the design that can be performed by selecting a check and clicking on the "check"-button. (When a check is OK, the circle will next to the checkbox will be coloured green, otherwise it will be coloured red.)
  - The first check verifies whether all information elements that are summed up in the XML-file appear at least once in an operation.
  - The second check verifies whether all defined operations from the XML-file appear at least in one activity of the selected design.
  - The last check verifies whether the resources are correct. This means that de resource to execute an activity from the design must be equal to or on a higher level than the resources to perform the operations of that activity.
- Available activities – The last group box contains all available activities of the selected design. For each activity the activity metrics (information cohesion, relation cohesion and activity cohesion) can be calculated by clicking on the "calculate cohesion"-button.

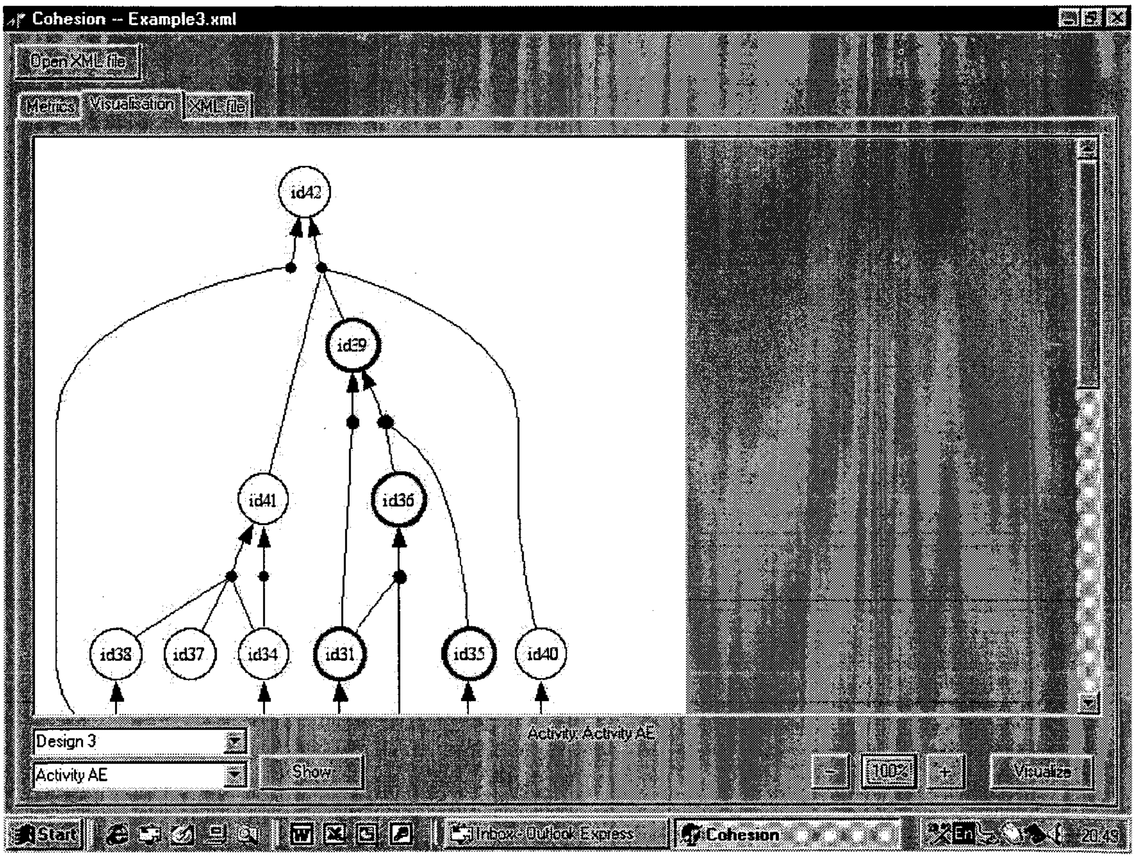
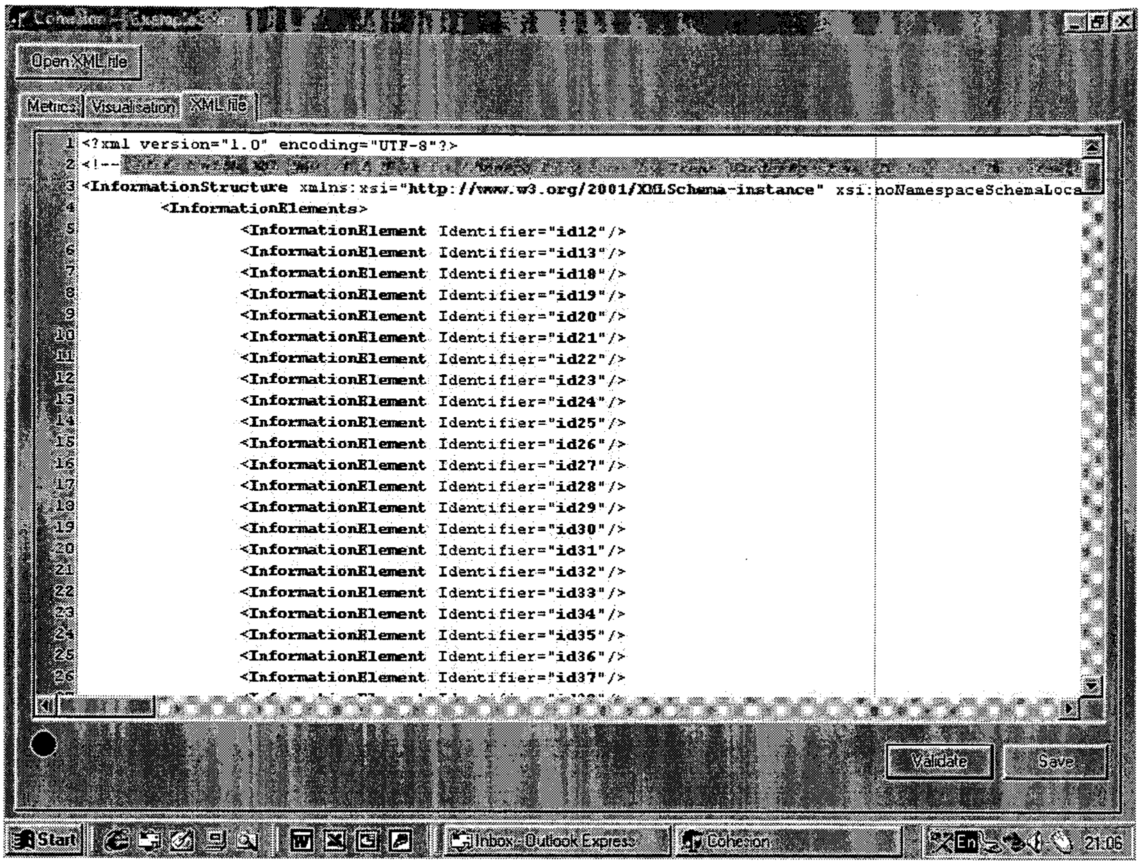


Figure 23: The visualisation sheet of the tool

The visualisation sheet

In the visualisation sheet it is possible to generate a drawing of the information element structure with "Dot". This can be done by clicking the "visualise"-button. Because most of the time the pictures can get very large, it is possible to zoom in and out.

By selecting a design and an activity and clicking the button "Show", the corresponding information elements and operations are coloured in red.



```
<?xml version="1.0" encoding="UTF-8"?>
<!--
<InformationStructure xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLoca
        <InformationElements>
                <InformationElement Identifier="id12"/>
                <InformationElement Identifier="id13"/>
                <InformationElement Identifier="id18"/>
                <InformationElement Identifier="id19"/>
                <InformationElement Identifier="id20"/>
                <InformationElement Identifier="id21"/>
                <InformationElement Identifier="id22"/>
                <InformationElement Identifier="id23"/>
                <InformationElement Identifier="id24"/>
                <InformationElement Identifier="id25"/>
                <InformationElement Identifier="id26"/>
                <InformationElement Identifier="id27"/>
                <InformationElement Identifier="id28"/>
                <InformationElement Identifier="id29"/>
                <InformationElement Identifier="id30"/>
                <InformationElement Identifier="id31"/>
                <InformationElement Identifier="id32"/>
                <InformationElement Identifier="id33"/>
                <InformationElement Identifier="id34"/>
                <InformationElement Identifier="id35"/>
                <InformationElement Identifier="id36"/>
                <InformationElement Identifier="id37"/>
```

Figure 24: The XML-file sheet of the tool

The XML-file sheet

In the XML-file sheet it is possible to change, validate and save the XML-file. In this way it is easy to make small adaptations to the designs, like the moving of an operation from one activity to another. The circle on the left hand side will indicate if the XML-file is valid or not (i.e. if it is well-structures according to the XSD-file). When it is not valid, the program will provide some hints to where to look for the problem.

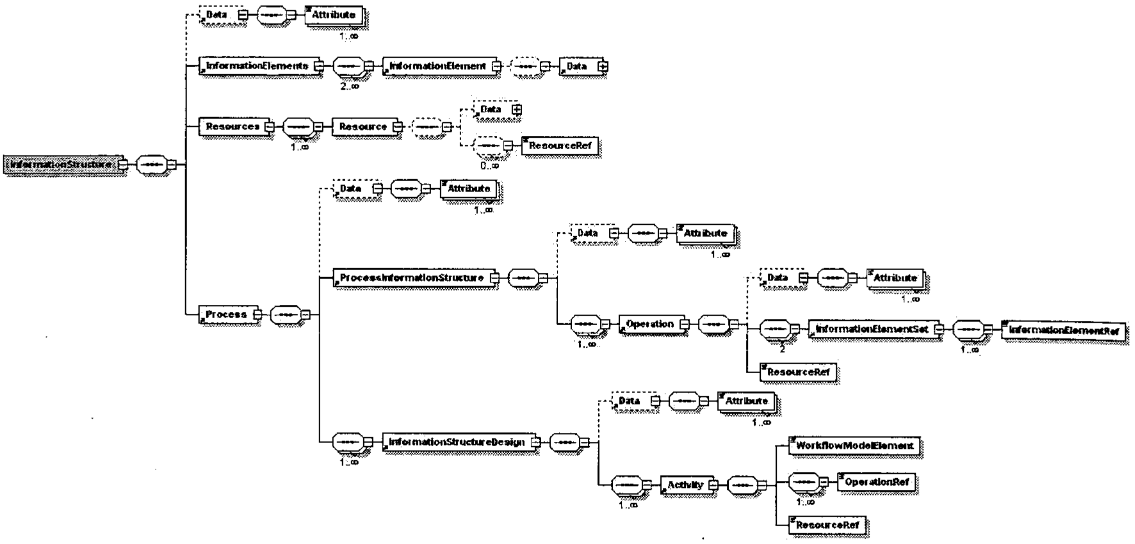


Figure 25: Graphical representation of the DTD-scheme.

G.2 XML Scheme Definition of information element structures

On the next pages you will find the DTD scheme for an information element structure. A graphical representation of this DTD scheme can be found in Figure 25.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.1 U (http://www.xmlspy.com) by Irene Vanderfeesten (Technische Universiteit Eindhoven) -
->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="InformationStructure">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Data" minOccurs="0"/>
                <xs:element ref="InformationElements"/>
                <xs:element name="Resources">
                    <xs:complexType>
                        <xs:sequence maxOccurs="unbounded">
                            <xs:element name="Resource">
                                <xs:complexType>
                                    <xs:sequence minOccurs="0">
                                        <xs:element ref="Data" minOccurs="0"/>
                                        <xs:sequence minOccurs="0" maxOccurs="unbounded">
                                            <xs:element name="ResourceRef" type="xs:IDREF"/>
                                        </xs:sequence>
                                    </xs:sequence>
                                    <xs:attribute name="Identifier" type="xs:ID" use="required"/>
                                    <xs:attribute name="Description" type="xs:string" use="optional"/>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element ref="Process"/>
            </xs:sequence>
            <xs:attribute name="description" type="xs:string" use="optional"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="Process">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Data" minOccurs="0"/>
                <xs:element ref="ProcessInformationStructure"/>
                <xs:sequence maxOccurs="unbounded">
                    <xs:element ref="InformationStructureDesign"/>
                </xs:sequence>
            </xs:sequence>
            <xs:attribute name="id" type="xs:string" use="required"/>
            <xs:attribute name="description" type="xs:string" use="optional"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="Data">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Attribute" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:simpleContent>
                            <xs:extension base="xs:string">
                                <xs:attribute name="name" type="xs:string" use="required"/>
                            </xs:extension>
                        </xs:simpleContent>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="InformationElement">
        <xs:complexType>
            <xs:sequence minOccurs="0">
```

```
                <xs:element ref="Data"/>
            </xs:sequence>
            <xs:attribute name="Identifier" type="xs:ID" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="ProcessInformationStructure">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Data" minOccurs="0"/>
                <xs:sequence maxOccurs="unbounded">
                    <xs:element ref="Operation"/>
                </xs:sequence>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Operation">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Data" minOccurs="0"/>
                <xs:sequence minOccurs="2" maxOccurs="2">
                    <xs:element ref="InformationElementSet"/>
                </xs:sequence>
                <xs:element name="ResourceRef" type="xs:IDREF"/>
            </xs:sequence>
            <xs:attribute name="label" type="xs:ID" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="InformationElementSet">
        <xs:complexType>
            <xs:sequence maxOccurs="unbounded">
                <xs:element name="InformationElementRef" type="xs:IDREF"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="InformationStructureDesign">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Data" minOccurs="0"/>
                <xs:sequence maxOccurs="unbounded">
                    <xs:element ref="Activity"/>
                </xs:sequence>
            </xs:sequence>
            <xs:attribute name="id" type="xs:string" use="required"/>
            <xs:attribute name="description" type="xs:string" use="optional"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="Activity">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="WorkflowModelElement" type="xs:string"/>
                <xs:sequence maxOccurs="unbounded">
                    <xs:element name="OperationRef" type="xs:IDREF"/>
                </xs:sequence>
                <xs:element name="ResourceRef" type="xs:IDREF"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="InformationElements">
        <xs:complexType>
            <xs:sequence minOccurs="2" maxOccurs="unbounded">
                <xs:element ref="InformationElement"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

In short we can say that an input file must first contain a list of information elements. After that, a list of resources follows. If a resource has subordinates, i.e. the resource can do the work of other resources, then this partial ordering is set down too. (A certain resource has a "ResourceRef" element, if the resource is on a higher level in the resource classification than the resource mentioned in

"ResourceRef". This means the resource is allowed to do all the work the resource in "ResourceRef" is allowed to do and more.)

Next, the process information structure is defined. After giving a description and an ID for the process, the operations are defined on the information elements. Every operation has an operation ID, followed by two sets of information elements and a resource. The first set of information elements contains the input elements of the operation. The second set contains the output elements.

Finally, several designs can be defined by defining the activities of the design on the operations. This means that for every design a number of activities are defined. These activities contain the operations of the process information structure.

G.3 XSD instance: request for student grant

In this section, you will find the XML Scheme Definition instance for the request for student grant process as described in chapter 3. Because there was no resource information we assumed all operations and activities are performed by the same resource, "resource 1".

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.1 U (http://www.xmlspy.com) by Irene Vanderfeesten (Technische Universiteit Eindhoven) -
->
<InformationStructure                              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="InformationStructure.xsd">
    <InformationElements>
        <InformationElement Identifier="id12"/>
        <InformationElement Identifier="id13"/>
        <InformationElement Identifier="id18"/>
        <InformationElement Identifier="id19"/>
        <InformationElement Identifier="id20"/>
        <InformationElement Identifier="id21"/>
        <InformationElement Identifier="id22"/>
        <InformationElement Identifier="id23"/>
        <InformationElement Identifier="id24"/>
        <InformationElement Identifier="id25"/>
        <InformationElement Identifier="id26"/>
        <InformationElement Identifier="id27"/>
        <InformationElement Identifier="id28"/>
        <InformationElement Identifier="id29"/>
        <InformationElement Identifier="id30"/>
        <InformationElement Identifier="id31"/>
        <InformationElement Identifier="id32"/>
        <InformationElement Identifier="id33"/>
        <InformationElement Identifier="id34"/>
        <InformationElement Identifier="id35"/>
        <InformationElement Identifier="id36"/>
        <InformationElement Identifier="id37"/>
        <InformationElement Identifier="id38"/>
        <InformationElement Identifier="id39"/>
        <InformationElement Identifier="id40"/>
        <InformationElement Identifier="id41"/>
        <InformationElement Identifier="id42"/>
    </InformationElements>
    <Resources>
        <Resource Description="Resource 0001" Identifier="Resource0001"/>
        <Resource Description="Resource 0002" Identifier="Resource0002">
            <ResourceRef>Resource0001</ResourceRef>
        </Resource>
    </Resources>
    <Process id="Process 1" description="Request for governmental student grant in the Netherlands.">
        <ProcessInformationStructure>
            <Operation label="OpID01">
                <InformationElementSet>
                    <InformationElementRef>id28</InformationElementRef>
                    <InformationElementRef>id29</InformationElementRef>
                </InformationElementSet>
                <InformationElementSet>
                    <InformationElementRef>id30</InformationElementRef>
                </InformationElementSet>
    <ResourceRef>Resource0001</ResourceRef>
</Operation>
<Operation label="OpID02">
    <InformationElementSet>
        <InformationElementRef>id24</InformationElementRef>
        <InformationElementRef>id25</InformationElementRef>
    </InformationElementSet>
    <InformationElementSet>
        <InformationElementRef>id28</InformationElementRef>
    </InformationElementSet>
    <ResourceRef>Resource0001</ResourceRef>
</Operation>
<Operation label="OpID03">
    <InformationElementSet>
        <InformationElementRef>id25</InformationElementRef>
        <InformationElementRef>id26</InformationElementRef>
    </InformationElementSet>
    <InformationElementSet>
        <InformationElementRef>id29</InformationElementRef>
    </InformationElementSet>
    <ResourceRef>Resource0001</ResourceRef>
</Operation>
<Operation label="OpID04">
    <InformationElementSet>
        <InformationElementRef>id12</InformationElementRef>
        <InformationElementRef>id13</InformationElementRef>
    </InformationElementSet>
    <InformationElementSet>
        <InformationElementRef>id18</InformationElementRef>
    </InformationElementSet>
    <ResourceRef>Resource0001</ResourceRef>
</Operation>
<Operation label="OpID05">
    <InformationElementSet>
        <InformationElementRef>id18</InformationElementRef>
    </InformationElementSet>
    <InformationElementSet>
        <InformationElementRef>id25</InformationElementRef>
    </InformationElementSet>
    <ResourceRef>Resource0001</ResourceRef>
</Operation>
<Operation label="OpID06">
    <InformationElementSet>
        <InformationElementRef>id19</InformationElementRef>
        <InformationElementRef>id20</InformationElementRef>
    </InformationElementSet>
    <InformationElementSet>
        <InformationElementRef>id23</InformationElementRef>
    </InformationElementSet>
    <ResourceRef>Resource0001</ResourceRef>
</Operation>
<Operation label="OpID07">
    <InformationElementSet>
        <InformationElementRef>id21</InformationElementRef>
        <InformationElementRef>id22</InformationElementRef>
        <InformationElementRef>id23</InformationElementRef>
    </InformationElementSet>
    <InformationElementSet>
        <InformationElementRef>id27</InformationElementRef>
    </InformationElementSet>
    <ResourceRef>Resource0001</ResourceRef>
</Operation>
<Operation label="OpID08">
    <InformationElementSet>
        <InformationElementRef>id18</InformationElementRef>
        <InformationElementRef>id27</InformationElementRef>
        <InformationElementRef>id32</InformationElementRef>
        <InformationElementRef>id33</InformationElementRef>
    </InformationElementSet>
    <InformationElementSet>
        <InformationElementRef>id40</InformationElementRef>
    </InformationElementSet>
    <ResourceRef>Resource0001</ResourceRef>
</Operation>
<Operation label="OpID09">
    <InformationElementSet>
        <InformationElementRef>id22</InformationElementRef>
        <InformationElementRef>id27</InformationElementRef>
        <InformationElementRef>id30</InformationElementRef>
    </InformationElementSet>
    <InformationElementSet>
        <InformationElementRef>id31</InformationElementRef>
    </InformationElementSet>
    <ResourceRef>Resource0001</ResourceRef>
</Operation>
<Operation label="OpID10">
    <InformationElementSet>
        <InformationElementRef>id18</InformationElementRef>
        <InformationElementRef>id32</InformationElementRef>
    </InformationElementSet>
    <InformationElementSet>
        <InformationElementRef>id35</InformationElementRef>
    </InformationElementSet>
    <ResourceRef>Resource0001</ResourceRef>
</Operation>
<Operation label="OpID11">
    <InformationElementSet>
        <InformationElementRef>id30</InformationElementRef>
        <InformationElementRef>id31</InformationElementRef>
    </InformationElementSet>
    <InformationElementSet>
        <InformationElementRef>id36</InformationElementRef>
    </InformationElementSet>
    <ResourceRef>Resource0001</ResourceRef>
</Operation>
<Operation label="OpID12">
    <InformationElementSet>
        <InformationElementRef>id31</InformationElementRef>
    </InformationElementSet>
    <InformationElementSet>
        <InformationElementRef>id39</InformationElementRef>
    </InformationElementSet>
    <ResourceRef>Resource0001</ResourceRef>
</Operation>
<Operation label="OpID13">
    <InformationElementSet>
        <InformationElementRef>id35</InformationElementRef>
        <InformationElementRef>id36</InformationElementRef>
    </InformationElementSet>
    <InformationElementSet>
        <InformationElementRef>id39</InformationElementRef>
    </InformationElementSet>
    <ResourceRef>Resource0001</ResourceRef>
</Operation>
<Operation label="OpID14">
    <InformationElementSet>
        <InformationElementRef>id22</InformationElementRef>
        <InformationElementRef>id27</InformationElementRef>
    </InformationElementSet>
    <InformationElementSet>
        <InformationElementRef>id34</InformationElementRef>
    </InformationElementSet>
    <ResourceRef>Resource0001</ResourceRef>
</Operation>
<Operation label="OpID15">
    <InformationElementSet>
        <InformationElementRef>id18</InformationElementRef>
    </InformationElementSet>
    <InformationElementSet>
        <InformationElementRef>id38</InformationElementRef>
    </InformationElementSet>
    <ResourceRef>Resource0001</ResourceRef>
    </Operation>
    <Operation label="OpID16">
        <InformationElementSet>
            <InformationElementRef>id34</InformationElementRef>
        </InformationElementSet>
        <InformationElementSet>
            <InformationElementRef>id41</InformationElementRef>
        </InformationElementSet>
        <ResourceRef>Resource0001</ResourceRef>
    </Operation>
    <Operation label="OpID17">
        <InformationElementSet>
            <InformationElementRef>id34</InformationElementRef>
            <InformationElementRef>id37</InformationElementRef>
            <InformationElementRef>id38</InformationElementRef>
        </InformationElementSet>
        <InformationElementSet>
            <InformationElementRef>id41</InformationElementRef>
        </InformationElementSet>
        <ResourceRef>Resource0001</ResourceRef>
    </Operation>
    <Operation label="OpID18">
        <InformationElementSet>
            <InformationElementRef>id27</InformationElementRef>
        </InformationElementSet>
        <InformationElementSet>
            <InformationElementRef>id42</InformationElementRef>
        </InformationElementSet>
        <ResourceRef>Resource0001</ResourceRef>
    </Operation>
    <Operation label="OpID19">
        <InformationElementSet>
            <InformationElementRef>id39</InformationElementRef>
            <InformationElementRef>id40</InformationElementRef>
            <InformationElementRef>id41</InformationElementRef>
        </InformationElementSet>
        <InformationElementSet>
            <InformationElementRef>id42</InformationElementRef>
        </InformationElementSet>
        <ResourceRef>Resource0001</ResourceRef>
    </Operation>
</ProcessInformationStructure>
<InformationStructureDesign id="Design 1" description="This design contains seven activities.">
    <Activity>
        <WorkflowModelElement>Activity A</WorkflowModelElement>
        <OperationRef>OpID01</OperationRef>
        <OperationRef>OpID02</OperationRef>
        <OperationRef>OpID03</OperationRef>
        <ResourceRef>Resource0001</ResourceRef>
    </Activity>
    <Activity>
        <WorkflowModelElement>Activity B</WorkflowModelElement>
        <OperationRef>OpID04</OperationRef>
        <OperationRef>OpID05</OperationRef>
        <ResourceRef>Resource0001</ResourceRef>
    </Activity>
    <Activity>
        <WorkflowModelElement>Activity C</WorkflowModelElement>
        <OperationRef>OpID06</OperationRef>
        <OperationRef>OpID07</OperationRef>
        <ResourceRef>Resource0001</ResourceRef>
    </Activity>
    <Activity>
        <WorkflowModelElement>Activity D</WorkflowModelElement>
        <OperationRef>OpID08</OperationRef>
        <ResourceRef>Resource0001</ResourceRef>
    </Activity>
    <Activity>
        <WorkflowModelElement>Activity E</WorkflowModelElement>
        <OperationRef>OpID09</OperationRef>
        <OperationRef>OpID10</OperationRef>
```

```
            <OperationRef>OpID11</OperationRef>
            <OperationRef>OpID12</OperationRef>
            <OperationRef>OpID13</OperationRef>
            <ResourceRef>Resource0001</ResourceRef>
        </Activity>
        <Activity>
            <WorkflowModelElement>Activity F</WorkflowModelElement>
            <OperationRef>OpID14</OperationRef>
            <OperationRef>OpID15</OperationRef>
            <OperationRef>OpID16</OperationRef>
            <OperationRef>OpID17</OperationRef>
            <ResourceRef>Resource0001</ResourceRef>
        </Activity>
        <Activity>
            <WorkflowModelElement>Activity G</WorkflowModelElement>
            <OperationRef>OpID18</OperationRef>
            <OperationRef>OpID19</OperationRef>
            <ResourceRef>Resource0001</ResourceRef>
        </Activity>
    </InformationStructureDesign>
    <InformationStructureDesign id="Design 2" description="This design contains nine activities.">
        <Activity>
            <WorkflowModelElement>Activity A1</WorkflowModelElement>
            <OperationRef>OpID02</OperationRef>
            <ResourceRef>Resource0001</ResourceRef>
        </Activity>
        <Activity>
            <WorkflowModelElement>Activity A2</WorkflowModelElement>
            <OperationRef>OpID03</OperationRef>
            <ResourceRef>Resource0001</ResourceRef>
        </Activity>
        <Activity>
            <WorkflowModelElement>Activity A3</WorkflowModelElement>
            <OperationRef>OpID01</OperationRef>
            <OperationRef>OpID09</OperationRef>
            <ResourceRef>Resource0001</ResourceRef>
        </Activity>
        <Activity>
            <WorkflowModelElement>Activity A4</WorkflowModelElement>
            <OperationRef>OpID11</OperationRef>
            <OperationRef>OpID10</OperationRef>
            <OperationRef>OpID13</OperationRef>
            <OperationRef>OpID12</OperationRef>
            <ResourceRef>Resource0001</ResourceRef>
        </Activity>
        <Activity>
            <WorkflowModelElement>Activity B</WorkflowModelElement>
            <OperationRef>OpID04</OperationRef>
            <OperationRef>OpID05</OperationRef>
            <ResourceRef>Resource0001</ResourceRef>
        </Activity>
        <Activity>
            <WorkflowModelElement>Activity C</WorkflowModelElement>
            <OperationRef>OpID06</OperationRef>
            <OperationRef>OpID07</OperationRef>
            <ResourceRef>Resource0001</ResourceRef>
        </Activity>
        <Activity>
            <WorkflowModelElement>Activity D</WorkflowModelElement>
            <OperationRef>OpID08</OperationRef>
            <ResourceRef>Resource0001</ResourceRef>
        </Activity>
        <Activity>
            <WorkflowModelElement>Activity F</WorkflowModelElement>
            <OperationRef>OpID14</OperationRef>
            <OperationRef>OpID15</OperationRef>
            <OperationRef>OpID16</OperationRef>
            <OperationRef>OpID17</OperationRef>
            <ResourceRef>Resource0001</ResourceRef>
        </Activity>
        <Activity>
```

```
                <WorkflowModelElement>Activity G</WorkflowModelElement>
                <OperationRef>OpID18</OperationRef>
                <OperationRef>OpID19</OperationRef>
                <ResourceRef>Resource0001</ResourceRef>
            </Activity>
        </InformationStructureDesign>
        <InformationStructureDesign id="Design 3" description="This design contains six activities.">
            <Activity>
                <WorkflowModelElement>Activity AE</WorkflowModelElement>
                <OperationRef>OpID01</OperationRef>
                <OperationRef>OpID02</OperationRef>
                <OperationRef>OpID03</OperationRef>
                <OperationRef>OpID09</OperationRef>
                <OperationRef>OpID10</OperationRef>
                <OperationRef>OpID11</OperationRef>
                <OperationRef>OpID12</OperationRef>
                <OperationRef>OpID13</OperationRef>
                <ResourceRef>Resource0001</ResourceRef>
            </Activity>
            <Activity>
                <WorkflowModelElement>Activity B</WorkflowModelElement>
                <OperationRef>OpID04</OperationRef>
                <OperationRef>OpID05</OperationRef>
                <ResourceRef>Resource0001</ResourceRef>
            </Activity>
            <Activity>
                <WorkflowModelElement>Activity C</WorkflowModelElement>
                <OperationRef>OpID06</OperationRef>
                <OperationRef>OpID07</OperationRef>
                <ResourceRef>Resource0001</ResourceRef>
            </Activity>
            <Activity>
                <WorkflowModelElement>Activity D</WorkflowModelElement>
                <OperationRef>OpID08</OperationRef>
                <ResourceRef>Resource0001</ResourceRef>
            </Activity>
            <Activity>
                <WorkflowModelElement>Activity F</WorkflowModelElement>
                <OperationRef>OpID14</OperationRef>
                <OperationRef>OpID15</OperationRef>
                <OperationRef>OpID16</OperationRef>
                <OperationRef>OpID17</OperationRef>
                <ResourceRef>Resource0001</ResourceRef>
            </Activity>
            <Activity>
                <WorkflowModelElement>Activity G</WorkflowModelElement>
                <OperationRef>OpID18</OperationRef>
                <OperationRef>OpID19</OperationRef>
                <ResourceRef>Resource0001</ResourceRef>
            </Activity>
        </InformationStructureDesign>
    </Process>
</InformationStructure>
```

H: Zur Muehlen's policies and their impact on JCM-dimensions

In this section, we will examine the impact of the assignment and synchronisation policies on the job dimensions in more detail. As described in section 4.4, the way in which the policies are carried out can influence the autonomy, skill variety, task identity, task significance and feedback an employee perceives in executing his/her job.
To be more clear about this impact we consider the assignment and synchronisation policies as some kind of axes or dimensions on which a "variable" can be changed. For example, when we consider the planning of new work items, this variable can be "tuned" to a net change policy or to a re-planning policy. When we vary the "value" of the variable on this axis, the "value" of the job dimensions will be affected too. Thus, the "tuning" of a policy can have a positive or negative effect on the job dimension. Of course, in the further elaboration of this research we will focus on the way in which a positive effect on the job dimensions can be achieved. But first we will explain the impact a policy has on the job dimensions and which job dimensions it concerns.

We will start again with the example of planning of new work items. When the policy dimension is "tuned" to a re-planning strategy (indicated by ↑re-planning) the employee will experience less autonomy in the performance of his or her tasks (indicated by ↓autonomy), because the work items that are assigned to him at a time, can disappear the next time the system re-plans the available work items. On the contrary, when a net change strategy is used (↑net-change), there will be more autonomy to the employee than in a re-planning situation. Thus, the net change policy has a positive impact on the autonomy (↑autonomy). In this case, we do not think the policy will affect another job dimension than the autonomy. For a graphical explanation of the meaning of the "axes" and the "tuning" of the variables in this example, see Figure 26.

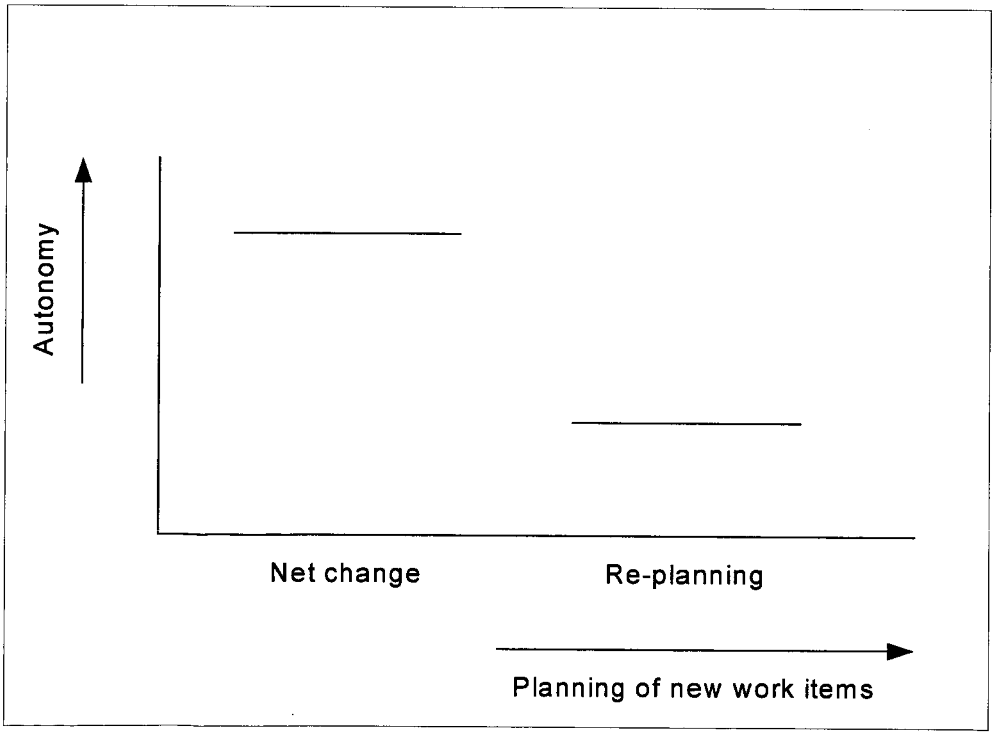


Figure 26: A graphical representation of the impact on autonomy of the planning of new work items assignment policy. This diagram shows that a net change strategy has a more positive effect on the autonomy an employee will experience than a re-planning strategy.

Similar to this track of reasoning the other policies can be examined on all job dimensions. This is summarised in Table 12 and will be explained in more detail below.

Assignment policy	**Autonomy**	**Skill variety**	**Task significance**	**Feedback**	**Task identity**
Planning of new work items					
↑ Net change	↑				
↑ Re-planning	↓				
Time of notification					
↑ Upon availability	↑				
↑ At latest start time	↓				
Queuing of new work items					
↑ Queue	↓	↓			
↑ Pool	↑	↑			
Activity execution					
↑ Individual	↑	↑	↓	↓	
↑ Collaborative	↓	↓	↑	↑	
Decision hierarchy					
↑ Final assignment	↓				
↑ Delegation possible	↑	↑/↓			

Table 12: The impact of changing an assignment policy on the job dimensions.

When the *time of notification* is upon availability, the employee will have more time to execute the work item, and does not immediately have to start the execution of the activity. It is possible to perform other activities first or to partially execute the work item, wait for a moment and then resume the work item. Therefore, (s)he has more autonomy (to decide when he wants to do what) than in a situation in which the time of notification is at the latest start time.

When work items are offered in a *queue* they have to be executed in the same order as they are offered. This means an employee has less freedom in deciding which work item (s)he wants to perform at what time, because (s)he has to follow the order in the queue. In comparison to a pool strategy for the queuing of new work items, the queue provides less autonomy. Moreover, when the employee is able to choose the work items he or she wants to execute from a list, he or she has the possibility to decide what kind of work he or she wants to do and he/she is able to vary the kind of work. Therefore, we think a pool will have a positive impact on the autonomy as well as the skill variety dimension. And of course, on the contrary, a queue will have a negative impact on these dimensions.

In our opinion, the individual *execution of activities* has two sides. On the one hand, we think the employee will experience more autonomy, because he/she can make his/her own decisions and does not have to meet the requirements of the team. In addition, (s)he will have more skill variety, because (s)he has to perform all work items that are offered, while in a team the employees usually execute activities they are good at. For the same reasons we think the collaborative execution of the work will have a negative impact on autonomy and skill variety. But on the other hand, working in a team can cause more task significance, because an employee is more able to place his/her work as a part of the whole. Moreover, because the close collaboration with colleagues it is more easy to get feedback on the work that is performed by an employee. These dimensions will be influenced negatively when an individual execution strategy is chosen.

The kind of decision hierarchy first of all has an impact on the autonomy of an employee. When a final assignment is chosen, the employee has to perform the activity, even if (s)he thinks it is too difficult, or if (s)he does not have time to do it. On the contrary, when delegation is possible he or she is allowed to send the work item to some one else or back to the system. This provides an employee with more autonomy. He or she can use this autonomy also to create more skill variety by sometimes rejecting a work item. But this can lead to misuse too, when an employee returns everything he does not like to perform.

Synchronisation policy		Autonomy	Skill variety	Task significance	Feedback	Task identity
Coordination						
↑	Hierarchical	↓	↓	↓		
↑	Market based	↑	↑	↑		
Allocation mechanism						
↑	Fully automated	↓	↓			
↑	Manual	↑	↑			
Participant selection						
↑	Direct	↑				
↑	Indirect	↓				
Assignment specification						
↑	Static			↓		↓
↑	Dynamic			↑		↑
Assignment of work items						
↑	Push	↓				
↑	Pull	↑				
Participant autonomy						
↑	Rejection possible	↑	↑			
↑	Assignment is final	↓	↓			

Table 13: The impact of changing a synchronisation policy on the job dimensions

When a hierarchical *coordination* mechanism is used, the employee cannot influence the kind of work items that are assigned to him/her. When a market based coordination policy is used employees can negotiate and consult about who has to do what. Clearly this can have a positive impact on the autonomy of an employee. He or she will experience more freedom in choosing work items, possibly providing more skill variety. In addition, a market based policy can also influence the task significance. By consulting about the division of work items among colleagues, all colleagues will have a notion of the total amount and kind of work that has to be performed. With this, they also have a better feeling of the part they have to do. Therefore we think a market based coordination policy has a positive influence on autonomy, skill variety and task significance, while a hierarchical coordination has a less positive impact on these dimensions, because the work is enforced.
We have to note here that it is not very clear from zur Muehlen's overview of the policy in Table 6, which strategies are opponent to each other on the axis of the coordination policy. However, from the description zur Muehlen gives, it is clear that a hierarchical and market based coordination strategy are opposites.

When a manual *allocation mechanism* is used the employee will experience more autonomy because it is possible to negotiate on the work items to be performed. Through this manual allocation a positive effect in skill variety can be achieved too. A manager can make sure himself that a particular employee gets varied work items or by negotiating the employee can make sure he/she gets different work items himself/herself. However, often a fully or partially automated allocation mechanism will be used, which provides less autonomy in comparison to the manual allocation.

By using indirect *participant selection*, the work items are not directly assigned to a physical employee, but to for instance a role. This makes it easier to facilitate a shared worklist, which provides more autonomy for the employees. When work items are assigned directly to an employee, this employee has no choice anymore of which items (s)he wants to perform or not and in comparison to indirect participant selection the autonomy will be influences negatively.

When a dynamic *assignment specification* mechanism is used it is possible to assign the work items belonging to a specific case, to the same employee as much as possible (cf. case management). In this way the employee will experience more task identity and more task significance than in case a static assignment specification is used.

The push mechanism in the *assignment of work items* to an employee forces a certain order in which the work items have to be executed and at what time the execution has to be started. In this case the system decides when and what has to be done. Therefore, it provides less autonomy than the use of a pull mechanism. Because in the pull mechanism the employee can decide himself/herself at what time he/she wants to start a work item.

If it is possible for an employee to reject a work item that has been assigned to him. He will experience more autonomy (control of the kind of work he has to perform). Through this autonomy his/her skill variety can be influenced positively too. On the contrary, when it is not possible to reject a work item, the employee is forced to perform it, which will result in a feeling of less autonomy and skill variety.

As mentioned before, it has become clear in this explanation that the job dimensions are strongly related to each other. It is justifiable that skill variety almost every time is caused by an improvement in autonomy. When an employee has more control over the thinks he has to do he also has more opportunities to make his work more varied.

I: List of ideas to "tune" a workflow system in a human oriented way

Below you will find the list of 21 easy to implement ideas to "tune" a workflow system in a human oriented way. The ideas are indicated by numbers and acronyms. After each idea you can also find the corresponding number of the idea from the original list between '()' and the corresponding policies, if they can be linked, between '{}'.

Release of work item / participant selection by workflow engine

1. [CASEMAN] Case management: let an employee work on the same case as much as possible. (11) {assignment specification}
2. [REDIRECT] Give employees the possibility to send a work item to another employee, who is better in performing the job, who has more knowledge about the case, who is not busy, etc. (26) {decision hierarchy, participant autonomy}
3. [REJECT] Give employees the possibility to reject a work item (with a valid reason) and return it to the workflow enactment service. (27) {decision hierarchy, participant autonomy}
4. [RELEASE] Release a new work item directly. (Time of notification is upon availability.) (28) {time of notification}
5. [REPLAN] Do not 're-plan' work items by workflow enactment service. (29) {planning of new work items}
6. [RESUBMIT] When a work item has to be performed again after a (negative result of a) check, return it to the same employee to execute it again. (32) {assignment specification}

Worklist

7. [PRIV PULL] Let an employee choose work items from the private worklist himself/herself: pull mechanism. (17) {queuing of new work items, assignment of work items, participant selection}
8. [SH PULL] Use a shared work list, from which an employee can choose himself/herself: pull-manner. (18) {queuing of new work items, assignment of work items, participant selection}
9. [#ITEMS] If possible, show more than one work item on the worklist of an employee, even if a push mechanism is used. (19)
10. [RANDOM] The queuing of work items in the work list should be random. (20) {queuing of new work items}
11. [BATCH] Offer an employee "batches" of work items. In this way the batch is pushed, but the employee can choose the order of execution of work items within this batch. (Here we assume the work list is private.) (21) {queuing of new work items, assignment of work items, participant selection}
12. [APPEAR] Give employees the opportunity to adjust the appearance of work items in their worklists to their own preferences: FCFS, earliest due date, random, etc. (Here we assume the assignment of work items is in a pull manner and the worklist is private.) (22) {queuing of work items}
13. [TEAM BAT] Create 'team batches' of work items. A team of employees (having the same competences/role) can divide the work according to their own preferences. (Here we assume the allocation mechanism is manual, but is not necessarily controlled by a team leader or manager.) This idea is quite similar to the concept of a self-managing team, which is one of the hot items in organisational psychology. (24) {coordination, allocation mechanism}
14. [TEAM WI] Create 'team work items'. Employees (with different competences) have to cooperate to execute an activity. (25) {activity execution}

Administration and monitoring

15. [PREFS] Keep up with the kind of activities an employee likes and make sure he or she will get more of this kind of activities (and less of activities (s)he does not like). (4) {assignment specification}
16. [HISTORY] Offer a variety in work items to an employee. Remember the kind of work items an employee has executed and decide, based on this history, what kind of new work items will be offered to him or her. (8) {assignment specification}
17. [STAT MOD] Design a possibility for the employee to examine the static process model in a comprehensible way (static aspect). (12)
18. [ROUTE] Give an employee the possibility to check the progress and route of a case during the process (dynamic aspect). (13)
19. [RESULT] Give each employee authorisation to view the final decision or result of a case in the process. (14)
20. [RANKING] Make available an employees place within the ranking of good employees (for instance 'hard working', 'producing high quality work'). (30)
21. [TARGET] Show an employee if he or she works hard enough, if he or she is satisfying the targets. (31)

J: Questionnaire 'human oriented "tuning" of workflow systems'

Release of work item / participant selection by workflow engine

	Positive impact on worker?	Easy to implement?	Comment(s)
1	yes / no / don't know	yes / no / don't know	....
2	yes / no / don't know	yes / no / don't know	....
3	yes / no / don't know	yes / no / don't know	....
4	yes / no / don't know	yes / no / don't know	....
5	yes / no / don't know	yes / no / don't know	....
6	yes / no / don't know	yes / no / don't know	....

Worklist

	Positive impact on worker?	Easy to implement?	Comment(s)
7	yes / no / don't know	yes / no / don't know	....
8	yes / no / don't know	yes / no / don't know	....
9	yes / no / don't know	yes / no / don't know	....
10	yes / no / don't know	yes / no / don't know	....
11	yes / no / don't know	yes / no / don't know	....
12	yes / no / don't know	yes / no / don't know	....
13	yes / no / don't know	yes / no / don't know	....
14	yes / no / don't know	yes / no / don't know	....

Administration and monitoring

	Positive impact on worker?	Easy to implement?	Comment(s)
15	yes / no / don't know	yes / no / don't know	....
16	yes / no / don't know	yes / no / don't know	....
17	yes / no / don't know	yes / no / don't know	....
18	yes / no / don't know	yes / no / don't know	....
19	yes / no / don't know	yes / no / don't know	....
20	yes / no / don't know	yes / no / don't know	....
21	yes / no / don't know	yes / no / don't know	....

The five best ideas are:

1. ....
2. ....
3. ....
4. ....
5. ....

K: Interview results

In this appendix the interview results of the expert validation of the ideas for a human oriented "tuning" of workflow systems are included.

In Table 14, the experts' opinion on the positive impact of an idea on the worker is represented. In this overview, a "+" means that the respondent thinks the idea has a positive impact, "-" means a negative impact and a "?" means the respondent did not know. Note that respondent R5, doubted the positive or negative impact in three cases and did not follow the format of the questionnaire.

K.1 Positive impact on worker?: interview results

	R1	R2	R3	R4	R5	R6
1	+	+	+	+	+	+
2	?	+	+	+	+/-	+
3	?	+	+	+	+	+
4	+	+	+	+	+	+
5	+	+	+	?	?	+
6	+	+	+	+	-	+
7	+	+	+	+	+	+
8	+	+	+	+	+	+
9	?	?	?	+	+	+
10	-	+	-	-	?	-
11	+	+	+	+	?	-
12	-	+	+	+	+	+
13	+	+	-	+	+	+
14	+	+	+	+	+	+
15	+	?	?	+	+/-	-
16	+	?	+	-	+/-	+
17	+	+	+	+	+	+
18	+	+	+	+	+	+
19	+	+	+	+	+	+
20	-	-	-	-	?	?
21	+	+	+	+	?	?

Table 14: Interview results on "Positive impact on worker?" for the 21 ideas.

Table 15 shows whether each of the respondents thinks the ideas are easy to implement. Here, a “+” means they think it is easy, a “-” means it is not easy, and a “?” again means they did not know. Note that R3, was not able to indicate the implementability for all of the ideas, for reasons mentioned before.

K.2 Easy to implement?: interview results

	R1	R2	R3	R4	R5	R6
1	+	?	?	+	+	-
2	?	+	?	+	-	+
3	+	+	?	-	+	+
4	?	+	?	+	+	+
5	?	+	?	+	+	-
6	+	+	?	+	+	+
7	+	+	?	-	+	+
8	+	+	?	+	+	+
9	?	+	?	+	+	+
10	-	+	?	-	+	-
11	+	+	?	-	+	-
12	-	?	?	+	+	+
13	-	?	?	+	+	+
14	+	-	?	+	-	+
15	-	-	?	-	-	-
16	-	?	?	-	-	?
17	+	+	?	+	+	+
18	+	+	?	-	+	-
19	?	+	?	-	+	?
20	?	+	?	?	-	-
21	?	+	?	?	-	+

Table 15: Interview results on "Easy to implement?" for the 21 ideas.

The respondents also depicted their top five of ideas, which is shown in Table 16. Each of them indicated their number one idea with a 1, number two with a 2, etc.

K.3 Top five of ideas: interview results

	R1	R2	R3	R4	R5	R6
1	2			3		
2			5	2		
3		5	5			
4						
5			2			
6		4	4		5	3
7	4					
8	3	3	1			1
9						
10						
11						
12				1		4
13				4	2	5
14	1			5		
15					3	
16					3	
17		2				
18	5				4	
19					1	
20						
21		1	3			2

Table 16: Interview results on the top five of the ideas for each respondent. They indicated the best idea on the first place (nu mber 1) and so on.