Effectiveness of lightweight workflow management systems

Sonnenberg, C.B.

Award date:
2007

Eindhoven, August 2007

Effectiveness of Lightweight Workflow Management Systems

by
Chris B. Sonnenberg

Student identity number 0578541

in partial fulfilment of the requirements for the degree of

Master of Science
in Operations Management and Logistics

Supervisors:
Dr.ir. H.A. Reijers, TU/e, IS
Dr.ir. J.J. Berends, TU/e, OSM

Ir. P. Eertink, Pallas Athena
P.H.M. Klaassen, Pallas Athena
Ir. IJ.G. Schilstra, Pallas Athena

TUE. Department Technology Management.
Series Master Theses Operations Management and Logistics, no. 13

ARW 2006 OML / 13

Subject headings: Workflow, Business Process Management

Summary

In search for better and more acceptable Workflow Management Systems (WMSs), lightweight WMSs were introduced. In the literature, one can find arguments that these lightweight WMSs can solve certain problems that regular WMSs face.

This master thesis project investigated what lightweight WMSs are, whether they can work in theory and if they can work in practice. Three research questions were formulated.

Research Question 1: *What is lightweight workflow management?*

Research Question 2: *Does lightweight workflow management work in theory?*

Research Question 3: *Does lightweight workflow management work in practice?*

The research questions each represent one part in this master thesis. Conclusions from each part are presented now.

Part I: What is lightweight workflow management?

In order to make a good definition of lightweight WMSs, different perspectives were researched, i.e. the *literature perspective*, the *workflow vendor perspective* and the *research specialists' perspective*. The basis for the literature perspective was two literature studies conducted earlier (Sonnenberg 2006a & 2006b). The different perspectives revealed what problems lightweight WMSs should address and what characteristics lightweight WMSs have. The problems that lightweight WMSs should address are related to implementation, footprint and support of organizational changes. The characteristics showed that lightweight WMSs are basically stripped versions of regular WMSs. Besides this difference in functionality, lightweight WMSs can be further be characterized by the properties illustrated in table I. The first two are expected to be the most important.

Number	Characteristic
1	Less functionality than regular WMSs
2	Faster implementation phases than regular WMSs
3	Smaller footprint than regular WMSs
4	Support of flexibility with respect to work procedures
5	Easy and fast support of organizational changes
6	Bottom-up implementation instead of the regular WMSs' top-down approach

Table I: Characteristics of lightweight WMSs

The characteristics were used next to make a definition of lightweight WMSs. To the best knowledge of the author, no definition of lightweight WMSs can be found in literature. However, literature did state that lightweight WMSs should have less functionality than regular WMSs (Agostini and De Michelis 2000a & Muth et al. 1999).

Definition Lightweight Workflow Management System

Lightweight workflow management systems are workflow management systems that only support basic functionality and are characterized by a short implementation phase and larger involvement of business users during the implementation and configuration phase.

Part II: Does lightweight workflow management work in theory?

The second part discussed the theoretical effectiveness of lightweight WMSs. The methodology that was used consists of three steps. In the first step a feature list was made for regular WMSs. An initial feature list was made first by conducting a literature study and extended next with information of Gartner, a prominent consultancy company (Gartner, 2003 and 2006). The feature list was then checked with employees of two prominent workflow vendors, i.e. TIBCO and Pallas Athena. In second step the author extracted features from the regular WMSs feature list that were suitable for lightweight WMSs. In the third step, these lightweight features were classified in essential, desirable and optional features, a classification made by the author himself. This was necessary because some features are more important than other features for lightweight WMSs. The result is the ideal lightweight feature list, presented in figure I.

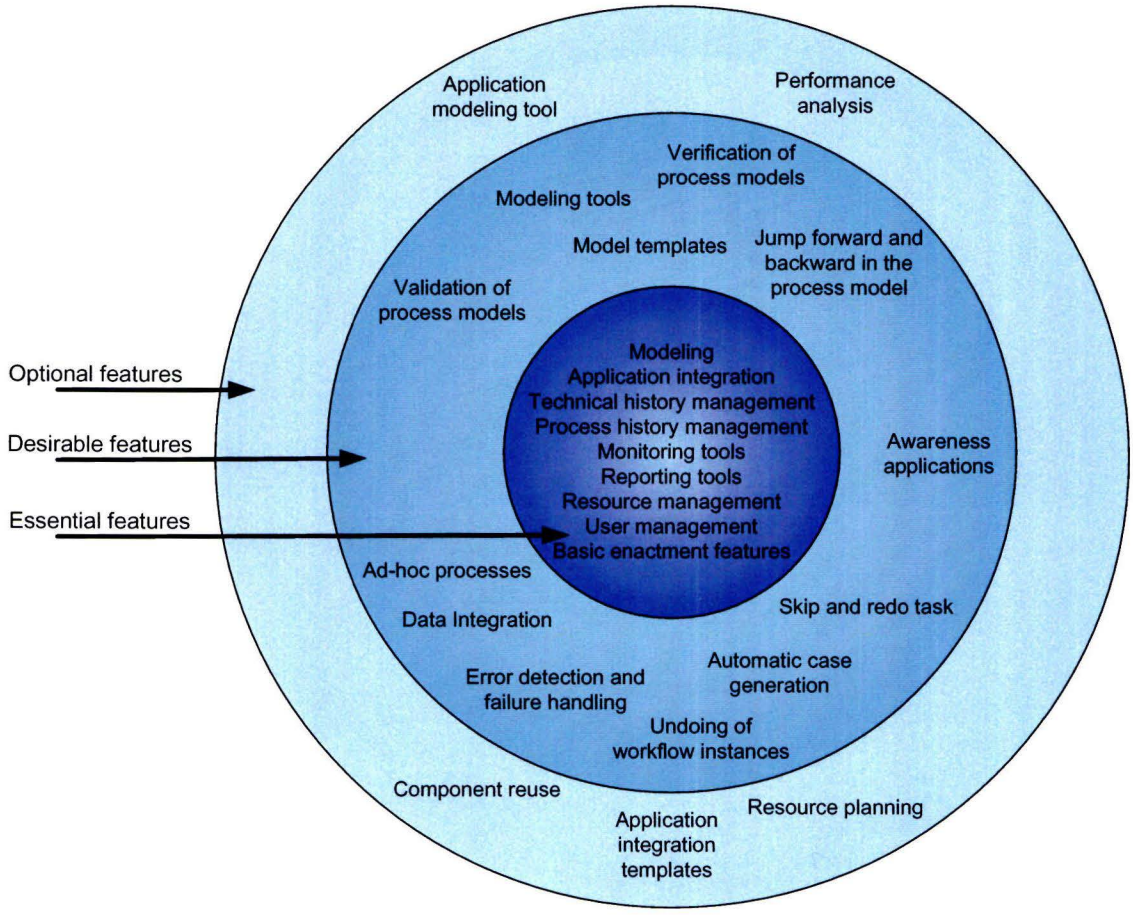

Figure I: Graphical representation of the ideal lightweight feature list

The theoretical effectiveness of lightweight WMSs was determined by checking whether lightweight WMSs' features can solve the problems that lightweight WMSs should address (implementation, footprint and organizational changes). The conclusion was that the lightweight WMSs' features can solve the problems that lightweight WMSs should address in theory.

However, one addition was made. The theoretical effectiveness also depends on the environment in which the lightweight WMS is used. According to the author, an environment is characterized by several aspects, like e.g. complexity of the situation and the volume of information that needs

to be processed. A situation has a mix of these aspects and the degree in which e.g. complexity is present differs among environments. These environmental aspects were therefore called *dimensions*, to indicate that they can differ and are more or less orthogonal. In this research, the suitability of an environment for lightweight WMSs was determined by using four different dimensions. The author first analyzed which features were relevant for a dimension and then determined the suitability of that dimension. The result is presented in table II.

Dimension	Suitable for lightweight WMS
High Process Complexity	**No**
Much Data Integration	**No**
High Volume Workflow Processes	**No**
Long Implementation Time	**Yes**

Table II: Suitability of workflow environments summarized

In general one can say that when an environment scores low on a dimension, it is suitable for lightweight WMSs and when it scores high, it depends on the type of dimension. Lightweight WMSs are i.e. not suitable for environments with high process complexity, much data integration or a high workflow process volume. Lightweight WMSs are suitable for environments that require a fast implementation.

Part III: Does lightweight workflow management work in practice?

The practical effectiveness was evaluated in part three using two case studies with the lightweight WMS Protos Activate. The first case study concerned a computer manufacturer, the second a mortgager. The case studies were classified using the three most relevant dimensions from part two, as indicated in table III. It was not possible to use more dimensions.

	Process complexity	Volume	Data integration
Computer manfacturer	High	Low	Medium
Mortgager	Low	High	Medium

Table III: positioning of cases studies

The evaluation of the case studies was done in two parts. The first part is the open evaluation and systematically covered all relevant aspects of lightweight WMSs by using a special, by the author developed, framework. The second part of the evaluation is the closed evaluation. This closed evaluation focused on the specific problems that occurred during the implementation phase at the case studies. Both case studies that are used in this research were successful and the lightweight WMSs improved the business processes in both situations.
The open evaluation showed that the lightweight WMS Protos Activate performed better than the regular WMS FLOWer in both case studies. The lightweight WMS performed better at the initial implementation, incremental implementation and the organizational changes part. It only performed slightly poorer on footprint.

The closed evaluation indicated several problems during the implementation phase. The first problem, *modeling patterns are not supported by the modeling tool of the WMS,* was encountered at the case of the computer manufacturer. In addition, this problem was also signaled during another implementation that the author carried out at Pallas Athena. The problem seems to have a

relation with the amount of process complexity and the author's proposition *the more process complexity an environment has, the likelier it is to encounter problems in modeling*, was confirmed. The second problem, *integration with other systems is not supported by default*, was encountered at the case of the computer manufacturer and at the case of the mortgager. Both cases had an environment with a medium amount of data integration and therefore this problem seems to have a relation with the amount of connections between the lightweight WMS and other systems. The author's proposition *the more data integration needed in an environment, the likelier it is to encounter problems when connecting the lightweight WMS to other systems*, was confirmed. The third problem, *customization of task lists is not possible*, occurred in the case of the computer manufacturer. The author's proposition *the more workflow process volume an environment has, the likelier it is that customization of task lists is needed* was, however, not confirmed. While the case of the computer manufacture was classified as *low* workflow volume, it did have problems with customization of task lists. Moreover, the case of the mortgager was classified as an environment with *high* volume workflow process and did *not* have problems with customization of task lists.

Despite the above problems that were encountered during the implementation phase, lightweight WMSs seem to work in practice. However, for the lightweight WMS Protos Activate, the support of flexibility should be improved to enlarge the effectiveness in practice. Protos Activate does support the features from the ideal feature list (figure I) except for the exception handling features that are needed for flexibility. When these features are supported as well, the lightweight WMSs probably performs better than the regular WMSs on all elements of the open evaluation. Thus, the conclusion on practical effectiveness of lightweight WMSs is that when lightweight WMSs have all features from the ideal lightweight feature list, they are effective in practice.

Acknowledgements

The author wishes to thank his supervisors from the Eindhoven University of Technology, Dr.ir. H.A. Reijers and Dr.ir. J.J. Berends, for their valuable input. In addition, the author likes to thank his supervisors at Pallas Athena: Ir. P. Eertink, P.H.M. Klaassen and Ir. IJ.G. Schilstra.

Furthermore, specials thanks go out to: Prof. W. van der Aalst (TU/e), J. Bosma (Pallas Athena), Prof. B.K. Choi (KAIST), H.C. Jeon (Realweb), G. Heijink (Pallas Athena), F. Knaapen (Pallas Athena), Dr. M. zur Muehlen (SIT), Dr. P. Muth (Mentor-Lite), S.M. Ryoo (Handysoft) and B. van Tol (TIBCO).

Last, the author would like to thank Rosalie Dijkman for her continues support.

Index

PART III: DO LIGHTWEIGHT WMSS WORK IN PRACTICE?

1 Introduction

Workflow management systems (WMS) have never become as popular as many thought two decades ago. Despite many attempts by workflow vendors to make better workflow management systems (WMSs), there are still drawbacks to the use of WMSs. In search for better and more acceptable WMSs, lightweight WMSs were introduced. In the literature, one can find arguments that these lightweight WMSs can overcome the problems that regular WMSs face. This master thesis project investigates what lightweight WMSs are, whether they can work in theory and if they can work in practice.

Business processes are becoming more complex every day and the need for automating support of business processes by WMSs has therefore increased. This trend is not exclusively encountered in large organizations; also small organizations have a greater need for automating support of business processes. Current WMSs are developed for usage in large companies or small organizations that handle many cases. They offer extensive functionality, are custom made and are expensive to purchase. These characteristics make them unsuitable for small scale usage in e.g. small companies or departments within larger organizations. As a result, there is a need for a variety in WMSs that support small scale usage.

Many variations in workflow management exist. Some notable variations are Flexible workflow management (see e.g. Faustmann, 1999) and Adaptive workflow management (see e.g. Kammer et al., 2000). These variations mainly address the rigidity of current WMSs. Lightweight workflow has another approach to current WMSs; it tries to make the WMSs more suitable for small scale usage. This is interesting for small organizations or small departments within organizations, since the usage of a full-fletched WMS is an excess in those situations.

The availability of lightweight WMSs is still limited. Only a few organizations offer smaller WMSs while it is expected that there is a sufficiently large potential market. Additionally, the literature studies conducted earlier by the author (Sonnenberg 2006a & 2006b) indicated that the information on lightweight workflow offered in literature is limited. A few research papers in literature indicate that lightweight WMSs should have less functionality than regular WMSs (e.g. Muth et al., 1999) but literature does not offer other information on e.g. other characteristics that lightweight WMSs should have. Furthermore, no definition on lightweight WMS can be found in literature. Therefore, the main objective of this master thesis project is to contribute to the literature and find out whether lightweight workflow is effective. If lightweight workflow appears to be effective, workflow vendors can consider offering a lightweight variant of their system for small scale usage.
In the next section, the research questions are presented to illustrate the objectives of this study more precise. The research methodology of this master thesis project is presented last.

1.1 Research questions

The main purpose of this master thesis project is to **find out whether lightweight WMSs are effective**. To answer this question, it has been divided into three research questions, presented below.

Research Question 1: *What is lightweight workflow management?*

It is not possible to determine the effectiveness of lightweight WMSs when it is not clear what lightweight WMSs are. Since no good definition of lightweight WMSs exists in literature, defining lightweight WMSs is the first research question in this master thesis.

Research Question 2: *Does lightweight workflow management work in theory?*

The second research question determines whether lightweight WMSs are theoretically effective.

Research Question 3: *Does lightweight workflow management work in practice?*

Besides the theoretical effectiveness, this master thesis project takes the practical effectiveness of lightweight WMSs into account as well.

This thesis is composed of three parts and each research question represents a part. Part one gives a definition of lightweight WMSs, part two considers the theoretical effectiveness and part three discusses the practical effectiveness.

1.2 Methodology

This methodology of master thesis project is based on the reflective cycle of Van Aken et al. (2007). The reflective cycle is depicted in figure 1.1 and can be divided into four parts. The first part is the so called *gap* and addresses the difference between existing literature and the business problem. This gap exists because literature does not offer a tailored solution or framework for the business problem. The purpose of this master thesis project is to investigate the gap between the literature and the business problem and likewise contribute to literature. The last paragraph of this section elaborates on the contribution of this master thesis.
The second part of the reflective cycle is the *case selection* and the *regulative cycle* (Van Strien, 1975). These are shown in the right-bottom corner of figure 1.1. As stated before, the gap between the actual design knowledge and the business problem needs to be closed. It is impossible to do this in a generic way and hence case studies need to be selected. In this master thesis project, two case studies were selected. The case studies follow the regulative cycle in figure 1.1. The cycle contains five steps, i.e. problem definition, analysis & diagnoses, plan of action, intervention and evaluation.
The third part of the reflective cycle is the reflection. In this part the cases are evaluated with the aim of learning for future cases (or projects). Case-specific elements are removed in order to develop general knowledge from the cases. According to Van Aken (2007), reflection is the broader interpretation of evaluation and can be divided into three subjects. The first subject addresses the learning for future problems, the second subject is the usage in advanced scientific knowledge about business processes and the final subject is necessary for personal and professional development.
The final part of the reflective cycle is codification. In this part the research questions can be answered. The contribution to the literature is also emphasized during this phase.

Positioning of the research questions
Three research questions were presented to find out whether lightweight workflow is effective. These questions can be seen as the business problem in this research. The business problem is defined differently in this research than one might expect and the reason for this is the nature of the master thesis project. The research does not handle a typical business problem but discusses a scientific topic (i.e. lightweight workflow).
Each research question represents a different part in this master thesis. The first two parts, or questions, are theoretical in nature and therefore are situated in the upper part of the reflective cycle. This is shown in figure 1.1. The third part in this research is practical in nature and hence it is situated in the regulative cycle in figure 1.1. For each case, the regulative cycle is executed.

After one cycle is completed, a reflection is made and then the cycle is executed again for the next case.

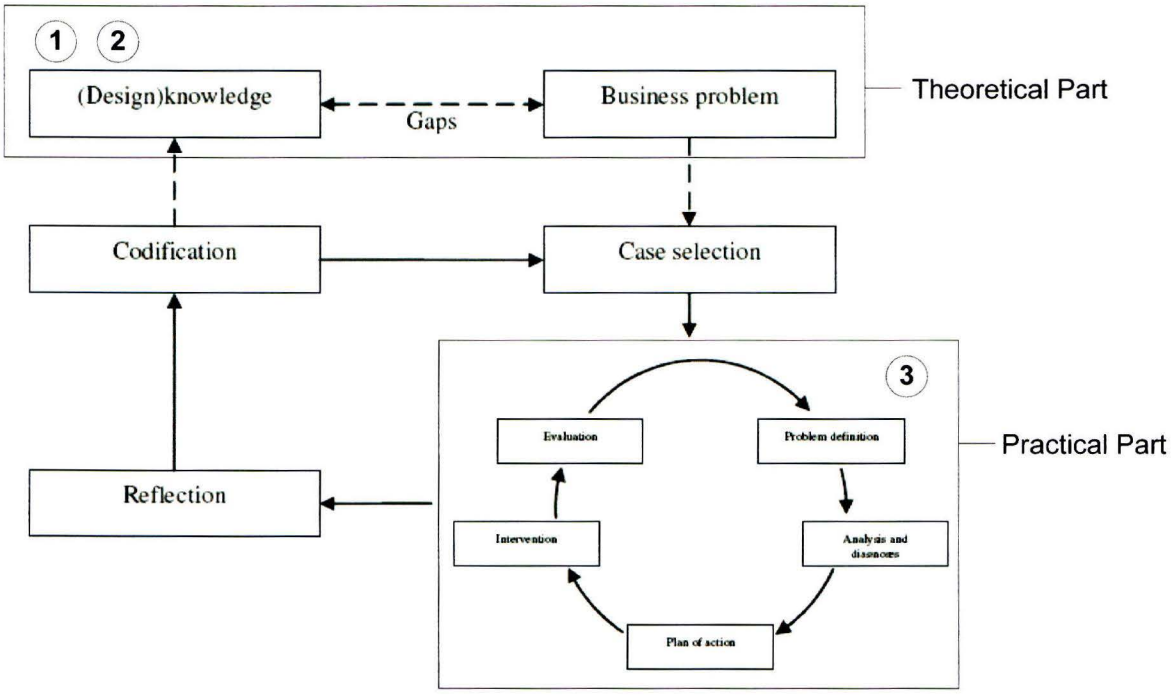

Figure 1.1: Reflective cycle (after Van Aken et al., 2007)

The remainder of this section describes the methodology for the three research questions, which each represent one part of this master thesis. The methodology of each part is explained into more detail in the corresponding chapters.

1.2.1 Methodology part I: What is lightweight workflow management?

Lightweight WMSs is a relatively new topic, not only in literature but in practice as well. Because it is a new topic, no definition of lightweight WMSs is made yet and individual opinions on lightweight WMSs differ. In order to make a good definition of lightweight WMSs, different perspectives need to be taken into account. Three perspectives were used, i.e. *literature perspective*, *workflow vendor perspective* and *the research specialists'* perspective. The methodology of all perspectives is briefly presented below, starting with the literature perspective.

First, a literature study was conducted on problems which regular WMSs face (Sonnenberg, 2006a). The outcome was then used in a second literature study (Sonnenberg, 2006b) that had a stronger focus on lightweight WMSs. This literature study investigated the problems that lightweight WMSs can address and the corresponding solutions that were discussed in the literature. Next, the results of this literature study were used in this master thesis project. A classification was made on the problems addressed by lightweight workflow. In addition, an overview was made on solutions that are presented in literature.

The second perspective is from workflow vendors. Interviews with two workflow vendors were conducted to reveal which solutions for future lightweight WMSs are initiated by workflow vendors. The involved workflow vendors are *Handysoft* and *Realweb*, two companies that operate in South-Korea. The reason for this choice is that the preparation of this master thesis was conducted in South-Korea. The interview at Handysoft was conducted with a senior engineer, the interview at Realweb with the Chief Technology Officer. Both persons have good knowledge on future developments of WMSs.

The third perspective is the research specialists' perspective. The initial idea was to conduct an extensive interview with three eminent researchers: Van der Aalst, Muth and Agostini. Van der Aalst is an important researcher in workflow management at Eindhoven University of Technology. Muth and Agostini were both leading researchers in the development of respectively the lightweight WMS *Mentor-Lite* and *Milano*.
Unfortunately, no comprehensive interviews could be held with all researchers. Muth and Van der Aalst were only available for a short interview which excluded the possibility to discuss lightweight workflow into detail. Furthermore, interviews with Agostini could not be held at all.

1.2.2 Methodology part II: does lightweight workflow work in theory?

The second research question discusses whether the principle of lightweight workflow can work theoretically. The methodology for answering this question is shown in figure 1.2.

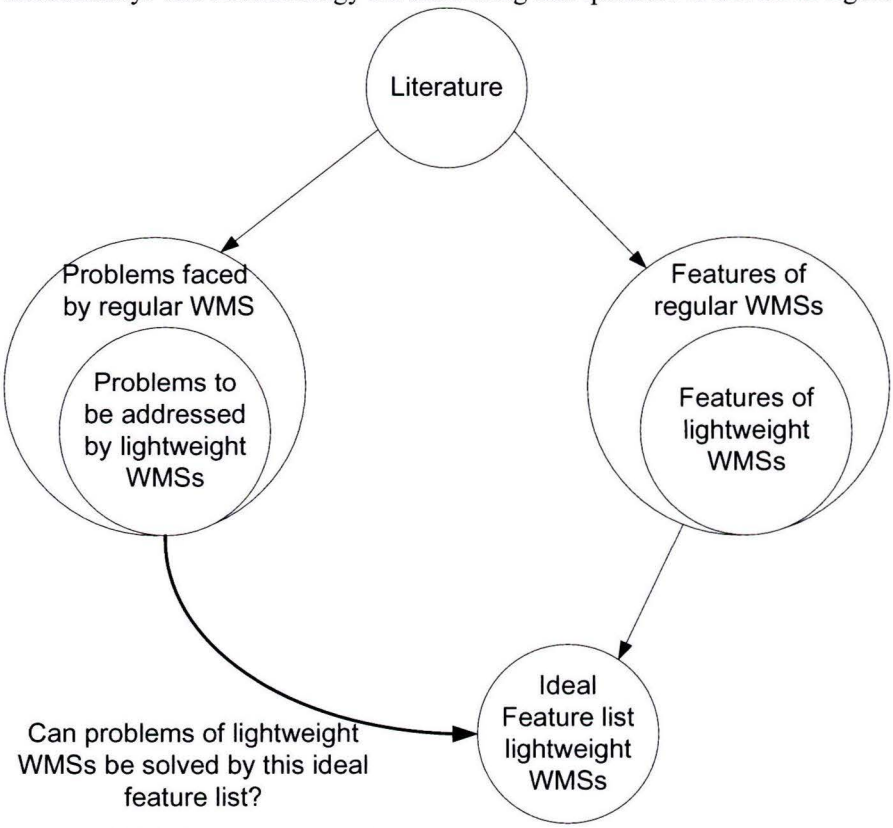

Figure 1.2: Methodology part two

The bigger circle on the left side represents general problems which regular WMSs face. This information was obtained from the first literature study (Sonnenberg, 2006a). A follow-up

literature study was conducted next (Sonnenberg, 2006b) to determine which problems are relevant for *lightweight* WMSs. The problems that should be addressed by lightweight WMSs are a subset of the problems that are relevant for regular WMSs. At the time of the second literature study, the author expected that lightweight WMSs have less functionality and are easier to implement than regular WMSs. These characteristics were used by the author to make the subset of problems that are relevant for lightweight WMSs.
The circle on the right side represents the features that regular WMSs have. The smaller circle in the bigger circle represents the features that could be supported by lightweight WMSs. The features of this small circle were then used to make an *ideal feature list*. This list represents features that lightweight WMSs should have in any case. The theoretical effectiveness of lightweight WMSs was then determined by answering the following question: '*Can the problems lightweight WMSs address (represented by the inner circle on the left side) be solved with the ideal lightweight WMS feature list (represented by the circle below)*?

1.2.3 Methodology part III: does lightweight workflow work in practice?

Two case studies were used to determine the practical effectiveness of lightweight WMSs. The first case study was conducted at the software department of a large computer manufacturer, the second at a mortgage selling company. In both case studies, Protos Activate was used. Moreover, both companies are based in The Netherlands.
Since lightweight WMSs are rather new, the choice in case studies was limited. Not many implementations of lightweight WMSs were executed yet and even fewer implementations were done during time of research. Additionally, a considerable amount of implementations were too specialized in nature which made them unsuitable for usage. The two case studies used in this research were the only suitable case studies that were available.

Both case studies follow the regulative cycle of Van Aken (2007), presented in figure 1.1. However, they do not cover all five steps in this cycle. The case study of the computer manufacturer is an evaluation of an implementation which was already finished before the start of this master thesis project. Consequently, only the evaluation step in the cycle is executed.
In the case study at the mortgager, the lightweight WMSs is not actually in use, because authorization is needed from a third party financial institution. Consequently, not all five steps from the regulative cycle can be executed. The initial plan was to cover all steps. However, the authorization from the third party financial institution has obstructed this. Nonetheless, the lightweight WMS is implemented and tested at the mortgager. Therefore, only the first three steps in the regulative cycle are executed, i.e. problem definition, analysis & diagnoses and plan of action.

1.3 Contribution to literature

The two literature studies on lightweight workflow that were conducted earlier (Sonnenberg 2006a & 2006b) have revealed the current (design)knowledge in literature. The corresponding gap between this (design)knowledge and the business problem is twofold. (1) A clear definition of lightweight WMSs cannot be found in literature, while this is important in research on lightweight WMSs. This master thesis project can contribute to the literature by providing this definition. (2) The literature does not offer much insight about the effectiveness of lightweight WMSs. The second and the third research questions address this issue in the way that they are investigating the theoretical and practical effectiveness of lightweight WMSs. The contribution to the literature is hereby twofold; (a) two case studies that determine the practical effectiveness of the lightweight WMS Protos Activate and (b) a theoretical reflection on the usefulness of lightweight WMSs with respect to regular WMSs.

Part I: Defining lightweight WMSs

2 What is lightweight workflow management?

This chapter discusses lightweight workflow management in general. It tries to reveal what lightweight workflow management exactly is, the problems lightweight workflow addresses and available solutions. This chapter provides the basis for later parts of this master thesis. The methodology of this chapter is presented first.

Methodology
The approach used in this chapter can be subdivided into three sections and is presented in figure 2.1

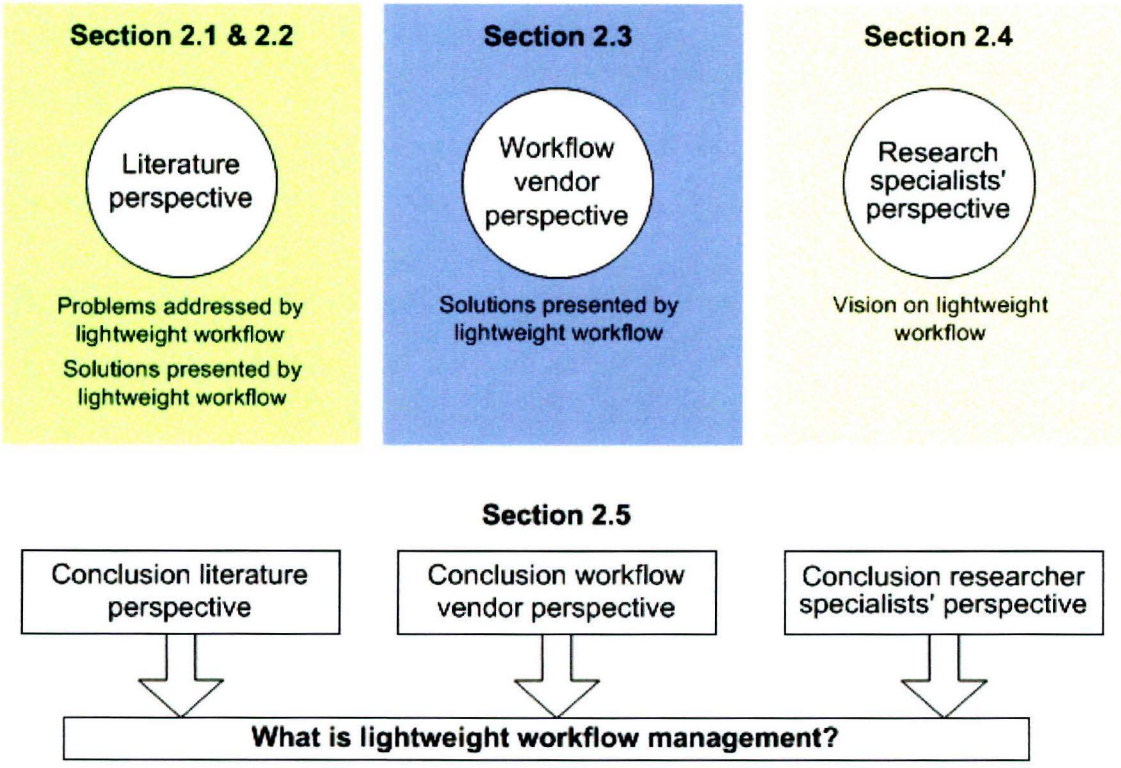

Figure 2.1: Methodology part 1

Section 2.1 lists problems and solutions described in literature regarding lightweight workflow management. The fundamentals for this section are two literature studies that were conducted earlier (Sonnenberg, 2006a & 2006b). Section 2.3 focuses on the perspective of workflow vendors. Workflow vendors have come up with lightweight solutions for the problems which WMSs face. The relation between these solutions and the problems stated in section 2.1, are presented in this third section. Section 2.4 focuses on the research specialists' perspective of lightweight WMSs. Interviews are conducted to gather the information of this perspective. The chapter concludes with a definition of lightweight WMSs. This definition is based on the conclusions from the first three sections.

2.1 Literature perspective: problems lightweight WMSs should address

This section discusses the literature perspective on lightweight workflow management. First, the problems that lightweight workflow addresses are presented. The next section, 2.2, discusses the solutions that are found in literature.
Lightweight WMSs are developed to overcome particular problems which exist in the traditional WMSs. Lightweight WMSs can be seen as stripped versions of regular WMSs with limited functionalities. Sonnenberg (2006a and 2006b) indicated that lightweight WMSs should address problems related to (1) implementation of the WMS, (2) footprint of current WMSs and (3) support for organizational changes. These different problems and other relevant aspects of Sonnenberg 2006a and 2006b are described in sections 2.1.1, 2.1.2 and 2.1.3.

2.1.1 Implementation related problems

It seems that the main problem lightweight WMSs are trying to solve is related to the implementation of the system. Implementation can be divided into two types; (1) the initial implementation of the WMS and (2) the incremental implementation(s). This division was initiated in Sonnenberg (2006b) with help from Gonzalez (2002).
The initial implementation is the actual implementation of the system. When a company has decided to use a WMS, this system is usually bought from a workflow vendor. The WMS cannot be used immediately; an (often) complicated configuration is needed to adapt the WMS to the business processes. Usually, a third party consultancy firm is needed for the initial implementation. Thus, the initial implementation is often complex and requires significant effort.
The literature study Sonnenberg 2006b furthermore indicated that

The second type of implementation is the incremental implementation. The incremental implementations are changes to the system once the system is running. There are four reasons for incremental implementations. (1) Errors in the development of the software which causes the WMS to malfunction. These design-related mistakes are generally made in the development phase of the WMS. Design-related mistakes should be prevented because it is much more expensive to change the software when the system is already in use than during the design phase. Another type of change is related to the (2) administration/management of the WMS. An example of this is security patches which should keep the WMS safe from intruders. The third type of change is related to the (3) organization. Business process change due to organizational changes and this requires changes in the WMS. The fourth type is (4) errors in process design and can be seen as errors in the configuration of the WMS. For example, a business process can be modeled incorrectly by a consultant because insufficient information on the business process was obtained from the business users.
The author expects that that the third type of changes occurs the most, however, administrative/management related changes and process design errors are also expected to be rather frequent.

The incremental implementations will be divided into Run-time and Build-time implementations (see e.g Rinderle 2004). Run-time implementations are changes made while the system is running. An example is a change in the routing of cases. Another example is modifications to the data that is processed in the WMS. Ideally, these changes should be made on-the-fly without stopping the system.
Build-time changes require the WMS to be stopped and therefore should be avoided when possible. An example of build-time changes are adjustments to the process models. Weber et al. (2004) state that it is better to make Run-time changes than Build-time changes.

2.1.2 Footprint problems

Regular WMSs leave a big footprint in the organization (Muth et al., 1999). The footprint denotes the negative impact a WMS brings to an organization. Lightweight WMSs address this footprint problem. Three different types of footprints are distinguished; the financial footprint, the human footprint and the technical footprint.

Financial footprint

One aspect of the footprint of a WMS is the price. The organization has to pay for the whole WMS if they buy a license of that WMS, independent of the features that will be used. It is thus important to compare the benefits of the WMS to the price which has to be paid. If only a few features will be used, cost savings or increased revenues could be less than the total purchase price of the WMS. In addition, when a company buys a WMS, it also has to pay for the maintenance and implementation of the WMS. These aspects belong to the Total Cost of Ownership of a WMS. According to Schilstra (appendix 4), the implementation and maintenance costs are much higher for a regular than a lightweight WMS. However, TCO is not considered in this master thesis project because gathering information on this topic takes too much time.

Human footprint

The second aspect of the footprint of a WMS is the influence it has on the work procedures. Consider the following example.

An employee of a municipality was used to make his own letters and e-mails for clients. The employee kept a file on his correspondence on his PC to review things later on. With the introduction of the WMS, the work procedure changed. The employee was not able to make his own letters and e-mails anymore and had to stick to the new company template for correspondence to clients. The file he kept on his PC was replaced by a database on the network.

In this example, the employee has approximately the same flexibilities as before. Yet, WMSs sometimes can bring rigidity to work procedures (see Agostini and De Michelis, 2000b). For example, an order process for which an order can *only* be accepted when a certain form is handed in via e-mail, hence telephonic orders are not possible anymore. WMSs can thus influence the work procedures and therefore cause a footprint with respect to work procedures.
According to Heijink (Interview Heijink, appendix 4), the human footprint is the most important in WMSs in comparison to the financial and the technical footprint. More attention should be given to this human aspect since the user acceptance is very important for the success of a WMS' implementation. Currently, WMS-vendors focus on technical issues of the WMS and should focus more on the acceptation of the WMS by the business users (Interview Heijink, appendix 4). The importance of user acceptance, stated above, is in line with conclusions from Davis and Venkatesh (2004). They state that many ICT projects fail. It is believed that a main reason for this is lack of user acceptance (Davis and Venkatesh, 2004). User acceptance is therefore very important, also in WMS implementations.

Technical footprint

The third aspect in footprint of a WMS is the technical footprint. Some WMSs e.g. require installing a fat client, a Database Management System (DBMS) or an application server. Others require merely more than a web-based interface. It is evident that the technical footprint in the first situation is bigger than in the latter. The examples show that the technical footprint can differ in size among different WMSs.
The technical footprint in current WMSs is mainly related to the network available in the company. Current WMSs rely heavily on the network infrastructure (Interview Heijink, appendix

4) and this network often becomes a bottleneck (Fakas, 2004). When not enough network capacity is available this can cause decreased performance of the WMS. Consequently, delays can occur that disturb the usage of the WMS.

2.1.3 Problems on support of organizational changes

Some lightweight WMSs also try to solve the problem of supporting organizational changes. Changes in the organization lead to changes in the workflow (model) and this is a problem with traditional workflow systems. Regular WMSs do not support these changes well (see e.g. Agostini and De Michelis 2000a & 2000b). Addressing organizational changes problems is not specific for lightweight WMS; flexible WMS also address this issue. In this research organizational changes in lightweight WMSs will be taken into account because it is expected that lightweight issues also play a part on this aspect.
Agostini and De Michelis (2000b) made a list which sums up the reasons why WMSs fail to support organizational changes. These are (i) no interruption of the system is possible, (ii) it is difficult to exit the normal flow and then reenter it again, (iii) breakdowns occur too often, (iv) changes are time consuming because of the complex architecture, (v) for making changes, experts are often needed, (vi) workflow systems do not support a process to be viewed from multiple perspectives and (vii) workflow systems make processes too rigid.
This list was then used by Agostini and De Michelis to determine which features WMSs should have to support organizational changes. These are: (i) it must be possible to interrupt the workflow system so that changes can be made, (ii) it must be possible to exit the normal flow and then reenter it again, (iii) it must be possible for end-users to change the workflow themselves (e.g. in order to deal with breakdowns), (iv) a simple system architecture is needed so that changes can be made fast, (v) it must be possible to let non experts execute changes, (vi) it must be possible to view workflow systems from multiple perspectives and (vii) workflow systems must not make processes too stiff. These seven features are related to each other and some of them are very similar. For this reason, the list is revised in the next paragraphs. Features which resemble each other are combined. This resulted in three elements in which the organizational changes aspect can be subdivided. These are *interruption of regular flow of work*, *easy and fast change of process models* and *representation of business processes*.

Interruption regular flow of work
In a WMS, many cases are processed at the same time. These cases are usually processed according to predefined process models. Sometimes it can be necessary to take a case out of the normal, predefined, flow (model) and let it reenter at another position in the flow. For example, it might be needed to jump back in the process model, or to jump forward in the model. There are numerous scenarios possible that ask for these flexibilities. Thus, an interruption of the flow should be possible in order to deviate from the regular workflow model. Features i, ii, iii and vii concern deviations from the normal workflow and for this reason these four are combined together.

Easy and fast change of process models
Another issue concerns changes in the process model and is related to features iv and v. Making changes is complex in current WMSs and therefore experts are often needed to make these changes. This causes changes in the process model to be expensive and time consuming. When process models are simpler and thus easier to change, business users can make changes themselves. This results in faster changes (no waiting on experts) and reduced costs. For these reasons, WMSs should have a simple architecture and the process models should be easy to change.

Representation of business processes
Users of WMSs should be able to handle events (such as breakdowns or exceptions) without being too restricted by the WMS. Some researchers say that this cannot be done because the formal models used in WMSs cannot deal with this (Dourish et al. 1996). Others also say that it *is* possible with formal models but that these need to be used with a different perspective (Agostini and De Michelis 2000b). Norman (1992) argues for the use of cognitive artifacts to make sure that users are not restricted by the WMS. Nonetheless, literature is very clear about the fact that WMSs are too rigid. The above literature also implicitly states that WMSs need to represent the business processes better, as they are in reality.
This better representation of the business processes implies several aspects. The first and most important aspect is related to the modeling language and its modeling power. When a modeling language does not have the potential to represent the business processes, the process models cannot represent reality at all. An example of this can be found in workflow patterns (Van der Aalst 2003 & www.workflowpatterns.com). When a modeling language does not support patterns which are essential in the business process, the process models are not likely to represent the business processes well.
The second aspect is that process models should be made in such a way that changes in the models are not needed too often. Business processes are flexible and this should be supported in process models. Thus, the process models in the WMS need to be made in such a way that exception handling is not required too often. It is necessary to have some sort of flexible interpretation of process models, or something similar, so that process models are good representatives of the many ways that business processes can be carried out. Whether this must be done by formal models, by cognitive artifacts or something else is not clear yet and is a topic for further research.

2.1.4 Summary

The previous sections are summarized in figure 2.2 below. The problems that lightweight workflow should address can be divided into three main categories: implementation, footprint and supporting organizational changes. The figure also depicts the main sources which have been used to make this classification.

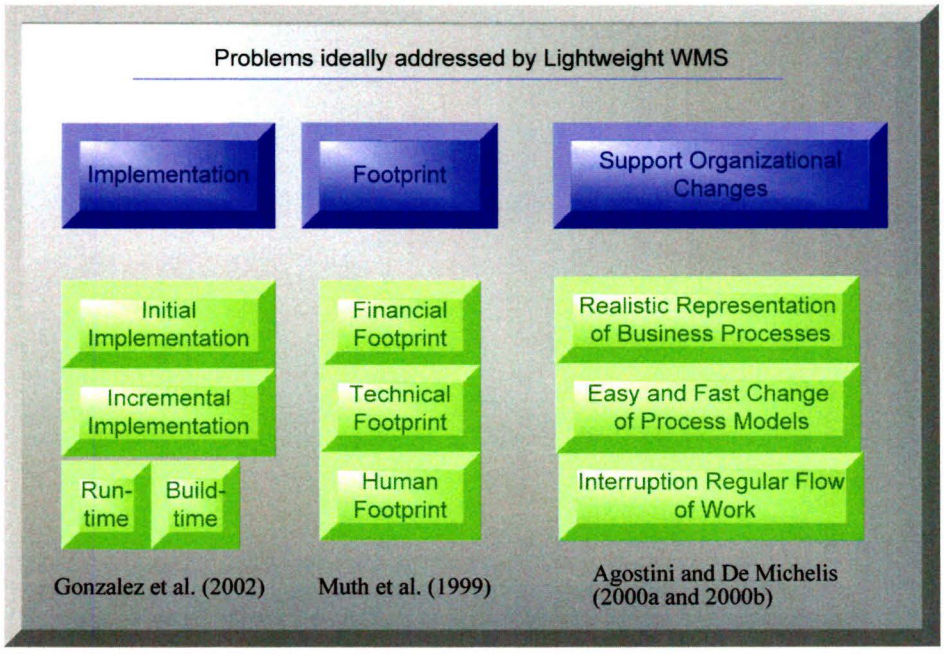

Figure 2.2: Problems addressed by Lightweight WMS

2.2 Literature perspective: solutions presented by lightweight WMSs

The previous section 2.1.4 provided an overview of the problems that lightweight workflow addresses. This section focuses on the solutions for these problems. A literature study was conducted (Sonnenberg 2006a & 2006b) on how lightweight WMSs deal with the problems regarding implementation, footprint and organizational changes and which techniques are used for this. The results and the argumentation for the results are described in this section in such way that these can be read without prior knowledge of the literature studies. However, the value of this section for the rest of this master thesis is limited. This section is included in this master thesis to provide a complete overview about the literature on lightweight WMSs.

2.2.1 Implementation related solutions

The implementation related problems were divided into two categories in this research; initial implementation and incremental implementation. In this section three solutions are presented that address the problems regarding implementation of the WMS. These are described below.

Faster initial implementation by reducing integration with other systems
The initial implementation of a WMS shall always be a task for which some technical expertise is required, as is the case for all IT systems installations. However, the extent to which technical expertise is needed can vary between IT systems. When numerous features need to be installed, the installation is more complex. All features need to be fine-tuned with the organization and furthermore it may be necessary to couple some features to other software packages used in the company. Integrating the WMS with other (legacy) systems is usually the most complex task in the initial implementation (more details can be found in the interviews of Handysoft and Realweb in appendix 1 and 2). Lightweight WMSs have fewer features than regular WMSs and offer only simple, straightforward connections. As a result the implementation phase becomes shorter.

Incremental implementations are changes to the WMS after the initial implementation has taken place. Two categories in incremental implementation are considered in this research; build-time changes and run-time changes. The next solutions can be implemented at either run-time or build-time.

No shutdown for incremental changes
Run-time changes are changes which are made while the WMS is still running. Lightweight WMSs allow run-time changes and some are described in Muth et al. (1999) and Shan et al. (2006). Shan et al. describe workflow views in their paper. A workflow view is a part of a workflow system and includes all aspects which are also present in a workflow. Usually, a system needs to be shut down when something like a workflow view is implemented. Shan et al. however, present a solution in which the workflow system does not have to be shut down. The basic idea can be found in the workflow kernel which does not need adaptation when view functions are implemented. This idea is similar to that of Weissenfels et al. (1998), described in the next paragraph.

No shutdown for changes in process model
An example of a build-time change in regular WMS is a change in the process model. Lightweight WMSs solve this problem partly by making it easier to adapt the process models. When changes are easier, they take less time. In addition, changes in the process models do not necessarily have to be build-time changes. The lightweight WMS Mentor-Lite supports changes in the process model while the system keeps running. The basic idea behind this is a small kernel in the architecture on which other components are built as extensions (Weissenfels et al., 1998).

2.2.2 Footprint related solutions

The footprint can be divided into the technical, financial and human footprint. The first solution addresses the technical and financial footprint. The second solution addresses the human footprint.

Limited support of functionality

The technical footprint is related to the amount of features of a WMS, in that fewer features are likely to contribute to a smaller technical footprint. The more features need to be installed or configured, the bigger the technical footprint. The same logic can be applied for the financial footprint. Fewer features reduce the financial footprint because the developments costs are related to the amount of features. Thus, fewer features thus reduce technical and financial footprint.

Exception handling

Lightweight WMSs can also help in reducing the human footprint. As stated in section 1.1.1.2, WMSs are often somewhat rigid and this causes reduced user acceptance. As described in section 1.1.1.3, lightweight WMSs' exception handling features provide flexibility that WMSs need. This flexibility reduces the human footprint.

Bottom-up implementation instead of top-down

A WMS is usually implemented via a top-down approach. This means that the use of a WMS is initiated via top management. A disadvantage of this is that fewer demands from the actual users are taken into consideration in the beginning of the WMS implementation, because the users are not directly involved. Furthermore, higher management levels, which stand between the actual users of the WMS and the workflow vendor, cause a delay in communication and might leave out some part of the information (Interview Realweb, appendix 2). Important details may be omitted and this can be harmful for the acceptance of a WMS. In general, a top-down approach leads to lower acceptance than a bottom-up approach (Interview Realweb, appendix 2).

Lightweight WMS can be implemented by a bottom-up approach. Lower (or middle) management can initiate a lightweight WMS because the impact on the organization is less. This means that the actual users can initiate the use of a WMS and this result in more commitment of the WMS (smaller human footprint). According to Realweb, a lightweight WMS should be implemented via the bottom-up approach (Appendix 2).

According to Pallas Athena, their lightweight WMS Protos Activate is usually bought because a business problem occurs. In these situations the middle management that is involved in the business problem initiates the usage of the lightweight WMSs Protos Activate. Therefore, Protos Activate is mainly implemented via the bottom-up approach. Protos Activate is bought for solving a specific problem and not as a general solution for e.g. improving information exchange organization wide. Solving a specific business problem is more visible and tangible to business users than solving a general issue like improving information exchange organization wide and hence the acceptance of the WMSs by business users is higher in lightweight WMSs.

2.2.3 Organizational changes related solutions

Solutions for organizational changes problems can be divided into three parts: realistic representation of business processes, easy and fast change of process models and support interruption of regular flow of work. As stated in section 1.1.1.3, all three parts should be addressed by lightweight WMSs.

Support interruption of regular flow of work by linear jumps

Some lightweight WMSs support interruption of the regular flow of work by having a feature that allows *exiting and reentering the flow (*Agostini and De Michelis, 2000b). This solution addresses

the problem which was categorized in section 1.1.1.3 as *interruption of regular flow of work*. It may be necessary to execute a flow in a different order for a certain customer and for this it must be possible to jump back and forward in the process model. This problem is addressed by some lightweight WMSs but not all. For example, Milano (see Agostini and De Michelis 2000a and 2000b) supports this interruption of the regular flow of work. By using formal models in which it is allowed to make so called linear jumps, a user can jump back and forward in the process. The solution Agostini and De Michelis developed is based on Petri Nets in which tokens jump between states. They distinguish strong linear jumps and weak linear jumps. Strong linear jumps are jumps which involve one token in the process model and these jumps require no authorization. Weak jumps require authorization from the process initiator and are somewhat more complex. In weak jumps two or more tokens are cancelled and another token is written in the process model. The linear jumps Agostini and De Michelis introduced allow interruption of the regular flow of work and are a solution for this problem.

Easier changing of process models by using simple models
Lightweight WMSs should be kept as simple as possible (Agostini and De Michelis 2000a). Simple systems induce a distinction between the control flow, resources, data flow and operation. Agostini and De Michelis call this *divide et impera*, separation of elements. With this separation of elements and simple models it becomes easier to change the process model. This addresses the problem of easy and fast change of process models, something which is necessary to support organizational changes. For example, the lightweight WMS Protos Activate makes it easier for the business users to make adaptations in the process models and thus no technical experts are needed all the time. After all, when editing process models is easy the business users can adapt the models themselves. The time between signaling a need for changing a process model and putting it into service can be reduced if changes are easy. This contributes to a good representation of the real business processes because models can change quicker and therefore the 'wrong' models do not need to be used for a longer time anymore. For this reason simple models can partly solve the problem of easy and fast change of process models.

The third category in organizational changes is realistic representation of business processes. Lightweight WMSs such as Milano and Mentor-Lite do not have solutions for this problem according to literature. Realistic representation of business processes is strongly related to the modeling language used in a WMS and this is beyond the scope this research.

2.3 Workflow vendors' perspective on lightweight workflow

The previous sections have focused on problems and solutions presented by the literature. Workflow vendors also have come up with some solutions to overcome problems which regular WMSs face. This section presents an overview of the solutions that could possibly be used in lightweight WMSs. Detailed information can be found in appendices 1 and 2 which include interviews with two workflow vendors: *Handysoft* and *Realweb*. The solutions described in this section are mainly implementation related solutions.

Faster initial implementation by standardization
Customization of the WMS to the customers' wishes requires a lot of time. The experience of the workflow vendor *Handysoft* is that the implementation time can be reduced by 40% if no special customization is needed. The workflow vendor *Realweb* also recognizes this issue and they also believe that standardization can reduce the implementation time considerable. Both companies argue that lightweight WMSs should be standardized. Handysoft and Realweb mention several reasons that make lightweight WMSs more suitable for standardization in comparison to regular WMSs. Lightweight WMSs have fewer features and a simpler architecture than a regular WMS.

Furthermore, lightweight WMSs generally have *fewer connections* to existing (legacy) systems. Last, the situation in which lightweight WMSs are used is less complex in comparison to regular WMSs.

Easier modeling by more suitable design language
Realweb is developing an easy to use modeling language which can be used with their new lightweight WMS. This language is designed in such a way that business users themselves can model the business processes. Pallas Athena also recognizes this and they already have developed a modeling tool which can be used by business users, i.e. Protos. Handysoft states that they are also looking for a better design language for their future lightweight WMS. This solution is not only an *implementation* related solution; it also has an impact on *organizational changes*.
The modeling tool Realweb is making requires practically no training for new users. Pallas Athena has developed a tool which requires some training (approximately two days) before it can be used by business users. Handysoft currently gives new users 40 hours of training in modeling with their tool. Their tool is difficult to use and based on the BPMN[1] and BPEL[2] standard.

Faster modeling using templates
Like regular WMSs, lightweight systems should also support the use of process model templates. Model templates can shorten the implementation time. Handysoft, Realweb and Pallas Athena all use templates with their regular WMSs. Pallas Athena uses templates in their lightweight system Protos Activate as well. This solution is not only an *implementation* related solution; it also has an impact on *organizational changes*.

No shutdown for updates
A WMS needs to be updated every now and then. A disadvantage of updates is that the workflow system has to be shut down for this. While the system is shut down, the users cannot do their regular work and this makes shutdowns costly. Some organizations bypass this problem by performing updates during the night, so fewer users are affected. However this does not solve the problem.
Another problem of a shut down of the system is the risk for losing valuable data. *Current data*, that has not been stored in databases at time of a shutdown is lost and cannot be recovered. Handysoft thinks the problem of shutdowns can possibly be overcome by lightweight WMSs. The lightweight WMS does not need to be shutdown for updates or patches when a certain technique is used. This technique is based on the backup of current data and is explained in appendix 1.

Faster initial implementation by reducing integration with other systems
Reducing integration does not only lead to better performance of the WMS, it also can shorten the initial implementation time. It is difficult to say to what extent the initial implementation time can be shortened, further research is necessary to reveal this.

The last solution presented in this section is not related to the implementation phase and cannot be categorized among footprint or organizational changes either.

Better performance by reducing integration with other systems
Usually, a WMS is integrated with other systems. The new lightweight WMS of Realweb has preferably no connections with other systems. The problem with integration with other systems is that it usually takes considerable time before information can be retrieved from these other systems (e.g. a delay of a couple a seconds). This may lead to performance problems of the

[1] Business Process Modeling Notation, www.BPMN.org
[2] BPEL Business Process Execution Language, www.ebpml.org/bpel_2_0.htm

WMS. Therefore, the new lightweight WMS of Realweb does not support connections to other systems. This makes the lightweight WMS faster and furthermore makes the initial implementation easier. When no connections are made, the WMSs architecture is less complex. Realweb calls their new lightweight WMS therefore 'stand alone'.

2.3.1 Overview solutions from literature and workflow vendors

The problems and solutions discussed in this chapter are summarized in the table below. The solutions from literature and workflow vendors are categorized according to the problems they address. One solution could not be categorized because it is relevant for all categories.

Category	Origin	Solutions
Implementation	Literature	Faster initial implementation by reducing integration with other systems
		No shutdown for incremental changes
		No shutdown for changes in process model
	Workflow vendors	Faster initial implementation by standardization
		Easier modeling by more suitable design language
		Faster modeling using templates
		No shutdown for updates
		Faster initial implementation by reducing integration with other systems
Footprint	Literature	Limited support of functionality
		Exception handling
		Bottom-up implementation instead of top-down
Organizational changes	Literature	Support interruption of regular flow of work by linear jumps
		Easier changing of process models by using simple models
No catagorization possible	Workflow vendors	Better performance by reducing integration with other systems

Table 2.1: Solutions from literature and workflow vendors

2.4 Research specialists' perspective on lightweight workflow

In addition to information gathered from literature and workflow vendors, research specialists' on workflow management were also interviewed. Involved researchers were Van der Aalst (see appendix 3) and Muth (see appendix 6). First, the results on the interview with Van der Aalst are discussed, next the results on the interview with Muth.

Van der Aalst states that lightweight WMSs can be lightweight on two aspects: technical and functional. Lightweight WMSs can be technical lightweight when they are e.g. easy to install and configure. An example of a WMS that is technical lightweight is the workflow module in Windows Vista. The Windows Workflow Foundation[3] has included a workflow module in Windows Vista. Since this module is integrated in the operating system, it is very easy to install. The workflow module is designed in such a way that it can be configured easily, also by non IT-experts. The WMS in Vista is not as advanced as regular WMSs and thus seems to be closer to lightweight WMSs.
Functional lightweight is light with respect to the usage of the system. When a WMS can be maintained and used easily without the use of experts, it is light in functionality. An example is

[3] http://msdn2.microsoft.com/en-us/library/ms735967.aspx

Protos Activate, the lightweight WMS from Pallas Athena. With Protos Activate, business users can create or adapt the process models themselves. This makes the WMS light in usage.

Van der Aalst identifies two future trends in lightweight workflow. The first trend is that WMSs are embedded in other applications like Enterprise Resource Planning systems or Product Data Management systems. ERP systems generally are larger, more complex and more embedded in organizations than WMSs. ERP systems are extended with a workflow module to enlarge their support to the organization. Van der Aalst thinks that future lightweight WMSs are thus not stand alone but will be integrated in applications like ERP systems.
The second future trend is the integration of lightweight WMSs in operating systems. An example of this is the Windows Workflow Foundation which included a WMS in the operating systems Windows Vista. Another example is described below.

In the early nineties, many companies used the program WordPerfect for word processing. Especially WordPerfect version 5.1 became a widely used application. In the late nineties, Microsoft promoted their own word processing application called Microsoft Word. Despite Microsoft did not actually *include* Word in their operating system, Word was practically *integrated* in Microsoft's operating systems.

In both trends mentioned by Van der Aalst, lightweight WMSs are embedded in other applications.

The second interview with research specialists' on workflow management was conducted with Muth. Muth was closely involved in the development of the lightweight WMS Mentor-Lite (Muth et al., 1999 and Weissenfels et al., 1998). This lightweight WMS was earlier described in section 1.2.1.
Mentor-Lite was developed for research purposes. The basic idea was to separate the workflow definition from the application logic by using services. The workflow is specified with e.g. BPMN and can call a service, independent on how the service is implemented. In this situation, it is indifferent on which platform the service operates or how the service is implemented.
Muth states that this separation is very relevant for the future of lightweight workflow. Business processes can be changed much faster when the application logic is separated from the workflow definition. As a result, organization changes can be implemented in the WMS relatively easy as no manual changes in application code are required.

The workflow definition and the application logic thus need to be separated according to Muth. This allows the WMS to adapt better to organizational changes as changes can be made effortless and straightforward.

However, the perspective of Muth is contradictorily to the perspective of Pallas Athena. Pallas Athena does not separate the workflow definition and the application logic in their lightweight WMS, it integrates them. Their argumentation for this is twofold: (1) by integrating the two layers, the implementation of the WMSs can be done faster. (2) Integration makes it easier for business users to change the process models because changes in the application logic are difficult to make. Since changes in application logic occur frequently, Pallas states that business users can not make the changes themselves they could make when the two layers would have been integrated.

2.5 Conclusion

The goal of this chapter was to describe what lightweight workflow management is. Perspectives from literature, workflow vendors and researchers' specialists were taken into account. The

conclusions from each perspective are discussed first. The definition of lightweight WMSs is presented last.

Literatures perspective
It seems that the main problems that lightweight workflow can solve are related to the implementation of the WMS. It is expected that lightweight workflow is most beneficial during the implementation phase. Other problems which lightweight WMSs address are related to the footprint of a WMS and the support for organizational changes.
Literature studies also identified solutions for the problems. Three solutions were found which address the implementation of a WMS, three that address the footprint of a WMS and also two that address the support for organizational changes. These solutions were shown in the table 2.1.

Workflow vendors' perspective
Workflow vendors have also come up with solutions which can be used by lightweight WMSs. The solutions are mainly implementation related solutions and are also depicted in table 2.1.

Researchers' perspective
In addition to literature and workflow vendors, researchers on workflow management were also interviewed. Involved researchers were Van der Aalst and Muth. Van der Aalst stated that WMSs can be lightweight in two ways: technically and functionally. Technical lightweight WMSs are easy to install and configure, functionally lightweight WMSs are easy to use by e.g. business users. In addition, Van der Aalst pointed out that lightweight WMS are likely to be included into other applications in the future. Van der Aalst believes that lightweight WMSs can potentially be integrated in operating systems or large software packages like ERP systems or PDM systems.

Muth's vision on lightweight workflow indicated that the workflow definition and the application logic need to be separated. This allows the lightweight WMS to adapt better to organizational changes as changes can be made effortless and straightforward. However, this perspective was in contrary to the perspective of Pallas Athena. They argue that the two layers need to be integrated.

2.5.1 What is lightweight workflow management?

This chapter concludes with a definition on lightweight WMSs. The characteristics of lightweight WMSs and the problems they solve are presented first. The definition is presented last.

Characteristics of lightweight WMS
This chapter showed that lightweight WMSs are basically stripped versions of regular WMSs. Besides this difference in functionality, lightweight WMSs can also be characterized by other properties. The most relevant properties are illustrated below in table 2.4. According to the interviews held with Handysoft and Realweb (appendix 1 and 2), the first two are expected to be the most important.

Number	Characteristic
1	Less functionality than regular WMSs
2	Faster implementation phases than regular WMSs
3	Smaller footprint than regular WMSs
4	Support of flexibility with respect to work procedures
5	Easy and fast support of organizational changes
6	Bottom-up implementation instead of the regular WMSs' top-down approach

Table 2.4: Characteristics of lightweight WMSs

Problems lightweight WMS addresses
Earlier conducted literature studies (Sonnenberg 2006a & b) revealed which problems lightweight workflow management should addresses. These are implementation, footprint and support of organizational changes and were depicted the figure 2.2.

Definition of lightweight workflow management
In order to define lightweight workflow, the definition of workflow in general and regular WMSs are required first. These two definitions were formulated by the Workflow Management Coalition (Hollingsworth, 1995) and are quoted below.

Definition Workflow:

> *The computerised facilitation or automation of a business process, in whole or part.*

Definition Workflow Management System:

> *A system that completely defines, manages and executes "workflows" through the execution of software whose order of execution is driven by a computer representation of the workflow logic.*

These definitions, together with the characteristic of lightweight workflow which were presented in this chapter, can now be used to define lightweight workflow. To the best knowledge of the author, no definition of lightweight WMSs can be found in literature. Literature did state (Agostini and De Michelis 2000a & Muth et al. 1999) that lightweight WMSs should have less functionality than regular WMSs. Furthermore, table 2.4 showed that the most important characteristics of lightweight WMSs were a fast implementation phase and support of less functionality. These aspects have led to the following definition in which lightweight WMSs are compared to regular WMSs.

Definition Lightweight Workflow Management System:

> *Lightweight workflow management systems are workflow management systems that only support basic functionality and are characterized by a short implementation phase and larger involvement of business users during the implementation and configuration phase.*

Part II: Do lightweight WMSs work in theory?

3 Theoretical effectiveness of lightweight WMSs

The previous chapter discussed lightweight WMSs in general. It explained what lightweight WMSs are, the problems lightweight WMSs addresses and available solutions. This chapter builds on the previous chapter by discussing lightweight WMSs from a theoretical perspective.

The structure of this chapter is as follows. In the first section an analysis is made about which features should be included in lightweight WMS and which in regular WMS. The second section looks closer at these lightweight features and classifies them. The third section discusses whether these lightweight features can solve the problems lightweight WMS should address, as presented in chapter 2. Section four and five discuss environments in which WMSs operate and their suitability for lightweight workflow. The chapter concludes with the theoretical effectiveness of lightweight WMSs.

Methodology
The second part of this master thesis tries to determine whether lightweight workflow can work in theory. The figure below shows this methodology.

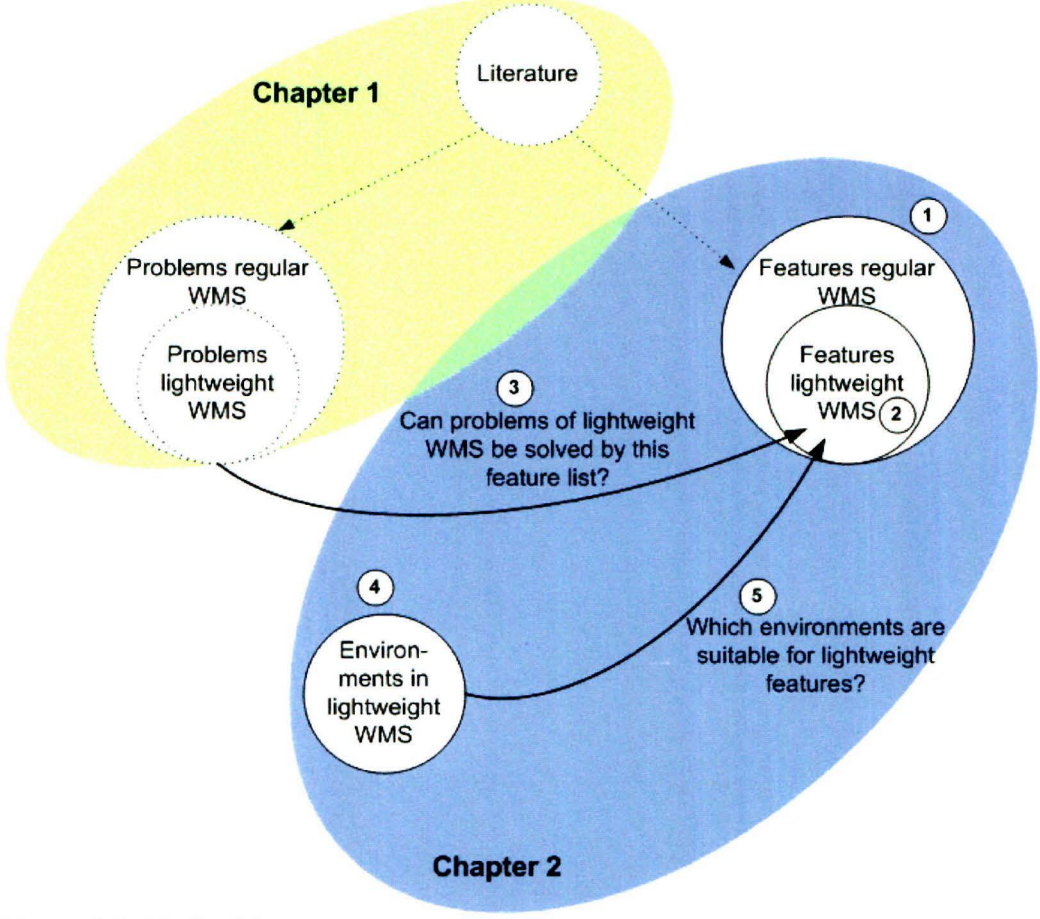

Figure 3.1: Methodology part 2

In this figure five parts are identified for chapter two, which represent the sections of this chapter. The first section is the basis and provides the fundamentals for the remainder of the chapter. Section one lists all features of a WMS and is put together in several phases. These phases are shown below, in figure 3.2.

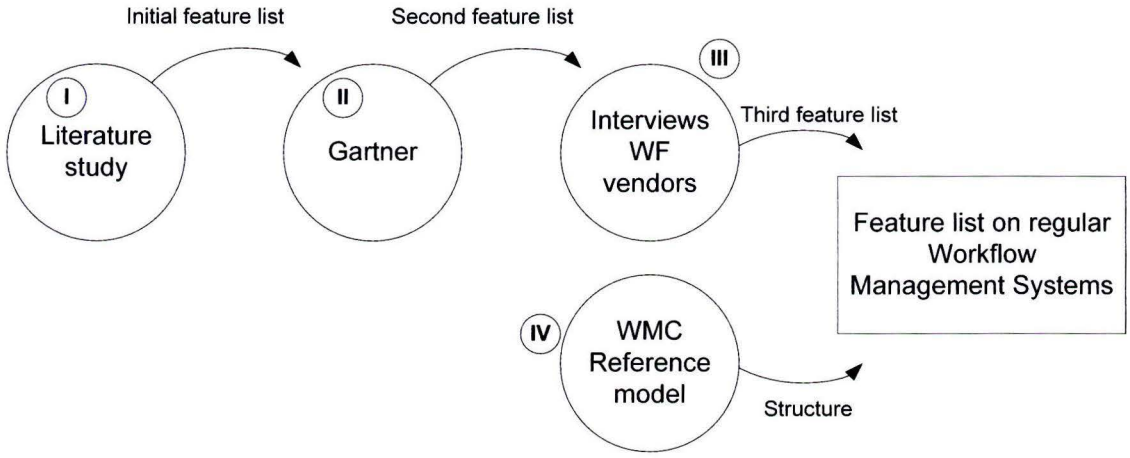

Figure 3.2: Phases in establishing the initial feature list

In the first phase (I) a literature study is carried out on features of WMS and the result is an initial list of features. In the second phase (II) this list is extended with information of Gartner, a prominent consultancy company (Gartner, 2003 and 2006). This results in a more complete list of WMS' features. In the third phase (III) this list is checked with several prominent workflow vendors so that missing features could be added and non relevant features could be deleted. Two big workflow vendors were used in this, i.e. TIBCO (TIBCO BPM) and Pallas Athena (FLOWer). TIBCO was chosen because it has a large market share in Europe and a large market share in the rest of the world as well. Pallas Athena's FLOWer was chosen for its innovative architecture. FLOWer is a case-based WMS and therefore works somewhat differently than other WMSs. The fact that FLOWer is different makes FLOWer suitable for adding features to the feature list. In the last phase (IV), the structure of the list was established using the workflow reference model (Hollingsworth, 1995). All features were positioned according to interfaces in the workflow reference model.

In the second section, a list of features suitable for lightweight WMSs was extracted from the general list of WMS features, that was established in section one. The extraction of the features is done by the researcher himself. Motivation for excluding features from the lightweight feature list can be found in appendix 9. Next, these lightweight features are classified in essential, desirable and optional features. This is necessary because some features are more important than other features for lightweight WMS. This classification is also made by the researcher himself. The result of this section is the ideal lightweight feature list.

The third section discusses whether the problems presented in part I of the thesis (figure 2.2) can be solved by the ideal lightweight feature list. This analysis is made by the researcher himself.

The fourth and fifth section elaborate on the environments in which WMSs can be found. Some environments are suitable for a lightweight WMS, others might not be appropriate. The sections analyze these workflow environments using seven different dimensions. Examples of dimensions

are process complexity and implementation time. The seven dimensions were established by the researcher himself, consultants at Pallas Athena and the first mentor of this master thesis project. At the end of the fifth section an overview is presented on the suitability of each environment. The required analysis was conducted by the researcher himself. The final section lists the conclusions of this chapter.

3.1 Features of regular and lightweight WMS

This section focuses on the features that regular WMSs offer. Appendix 8 provides an overview of all features which are commonly supported by WMSs including a brief description. In this appendix the features are also ordered according to the *workflow reference model* (Hollingsworth, 1995). This model covers the main aspects of a WMS and therefore provides a good structure for framing all features. The result is depicted below in figure 3.3.

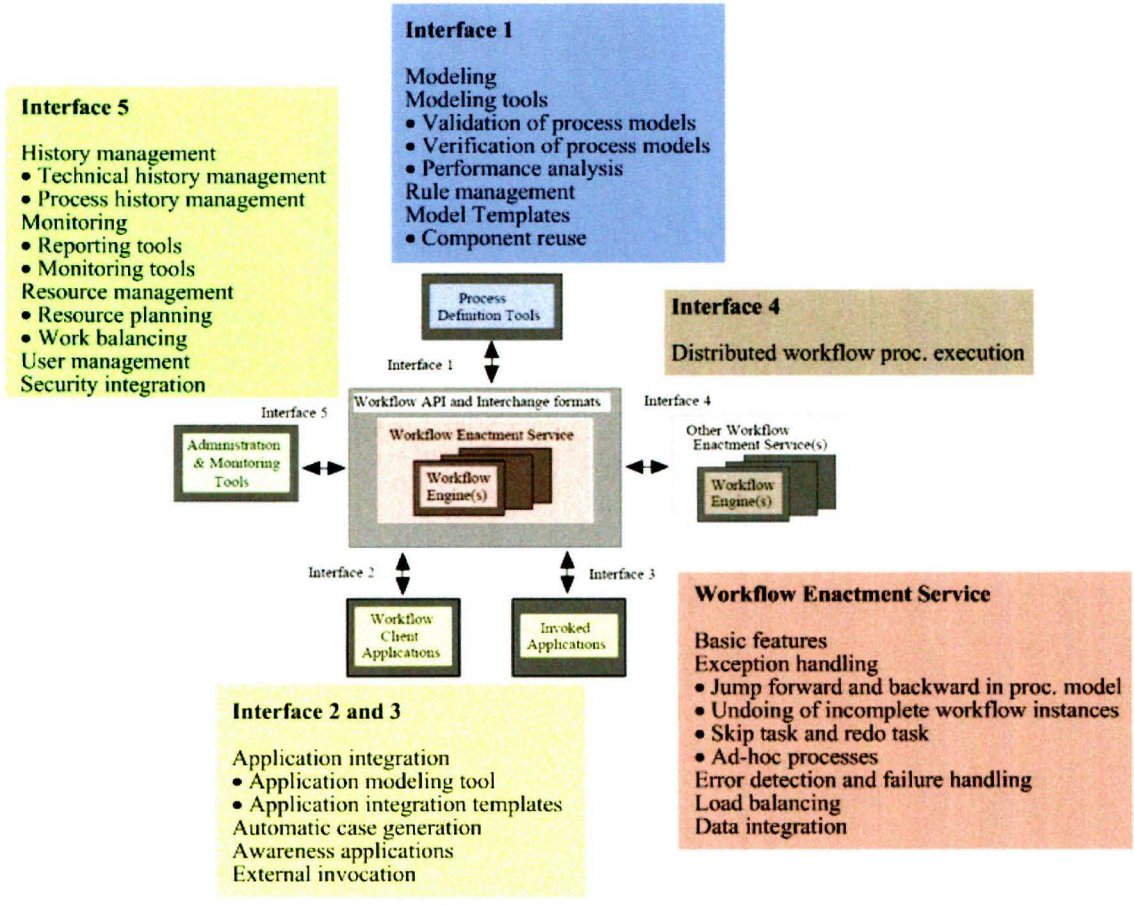

Figure 3.3: Features positioned according to the workflow reference model

3.2 Features Lightweight WMS

After classifying features according to the workflow reference model, the features are categorized in *lightweight features* and *Regular WMS only* features. This is done because not all features are suitable for lightweight WMSs (see appendix 9) and hence should not be included. In addition, when a lightweight WMS supports too many features it is not *light* anymore. Features that are classified as *regular WMS only* are excluded from lightweight WMSs because they are often

complex and cannot be integrated in the lightweight WMS easily. Nonetheless, most features can be included in both lightweight and regular WMSs. This classification is shown in table 3.1. Appendix 9 discusses the motivation for including/excluding features in detail. In the remainder part of this section the lightweight features are classified in three categories: essential, desirable and optional. The section concludes with an ideal feature list.

Essential features are the core of the WMS and should always be included. Without these essential features it is not possible to use the WMS. Desirable features are very useful features and are needed in most situations. Features in the desirable class are relatively straightforward and do not bring too much complexity to the system. This discerns them from the third class of features, the optional features. Features that are optional are either not needed for the majority of situations, or they bring a lot of complexity to the system. In general, they do not offer good value for money when considering using them by default in a lightweight WMS. If a feature is only used a few times than the costs for developing this feature are high in comparison to the benefit it brings to the lightweight WMS. Therefore, the value for money aspect on these features is poor and consequently it is better not to include these features by default in a lightweight WMS. High complexity and little usage are thus reasons for categorizing some features as optional features.
The classification in essential, desirable and optional can be clarified with some examples. The features described in the examples below are *modeling*, *modeling tools* and *performance analysis*. Modeling enables capturing business processes into process models. Modeling tools are programs that aid in modeling.

The modeling feature is an *essential* feature. Without process models the WMS cannot be used and hence this feature is an essential feature.
The modeling tool feature is, for example, a *desirable* feature. Modeling tools make it much easier to model the business processes. Some tools even enable non IT experts (like business users) to model the business processes. Furthermore, modeling is needed frequently in WMSs because process models are the basics. Hence the modeling tools will be used frequently. The frequent usage of this feature and the benefit it brings (easier modeling) make it a desirable feature.
An example of an *optional* feature is performance analysis. This feature can be seen as an advanced simulation feature and is mainly used in high volume workflow processes. For this reason this feature is probably not used often in lightweight WMSs. When one looks at the cost for developing this feature and the usage of it, it is very likely that this feature does not offer good value for money. Developing costs are high, usage is low. For this reason the performance analysis feature is classified as an optional feature.

In table 3.1 on the next page, lightweight features are classified in essential, desirable and optional. In this table also regular WMS features are shown. There are 33 WMS features in total of which 8 are essential, 14 desirable, five optional and six Regular WMS only.

One problem remains when categorizing features in essential, desirable and optional, i.e. some features can be considered as *slightly desirable*. For example, the data integration feature is a desirable feature for lightweight WMS according to the classification in table 3.1. However, one can imagine that *extensive* support of data integration is not very useful since this is a complex issue in WMS. Support of extensive data integration brings along a lingering implementation phase because much effort is required for all connections to other systems. To make these connections, customization of the WMS is needed which takes considerable time. This long implementation time is an inverse of the lightweight approach of a short and easy implementation stage. Therefore, in these situations it is likely that a regular WMS will perform better. Thus, for some features it is important to take the extent to which they are supported into account. In appendix 10, two additional examples of features which are not fully desirable are presented.

WF Reference model Interface[1]	Features	Classification	
		Regular or Lightweight WMS	*For lightweight WMS: Essential, Desirable, Optional*
1	Modeling	R + L	Desirable
	Modeling tools	R + L	Desirable
	Validation of process models	R + L	Desirable
	Verification of process models	R + L	Desirable
	Performance analysis	R + L	Optional
	Rule management	R	n.a.
	Modeling templates	R + L	Desirable
	Component reuse	R + L	Optional
2 & 3	Application integration	R + L	Essential
	Application modeling tool	R + L	Optional
	Application integration templates	R + L	Optional
	Automatic case generation	R + L	Desirable
	External invocation	R	n.a.
	Awareness application	R + L	Desirable
4	Distribution of workflow instances	R	n.a.
5	History management	R + L	Essential
	Technical history management	R + L	Essential
	Process history management	R + L	Essential
	Monitoring tools	R + L	Essential
	Monitoring tools	R + L	Essential
	Reporting tools	R + L	Essential
	Resource management	R + L	Essential
	Resource planning	R + L	Optional
	Work balancing	R	n.a.
	User management	R + L	Essential
	Security integration	R	n.a.
Workflow enactment service	Basic enactment features	R + L	Essential
	Exception handling	R + L	Desirable
	Jump forward and backward in process model	R + L	Desirable
	Skip and redo task	R + L	Desirable
	Undoing workflow instances	R + L	Desirable
	Ad-hoc processes	R + L	Desirable
	Error detection and failure handling	R + L	Desirable
	Load balancing between workflow engines	R	n.a.
	Data integration	R + L	Desirable

Legend

[1] Position according to the Workflow Reference Model (Hollingsworth, 1995)

n.a. Not applicable

Table 3.1: Classification of features

Ideal lightweight feature list

The ideal lightweight feature list is derived from table 3.1. This is graphical represented in the sphere in figure 3.4. The sphere is composed of three parts. The core of the sphere shows the

essential features. The circle around the core represents the desirable features and the outer circle depicts the optional features.

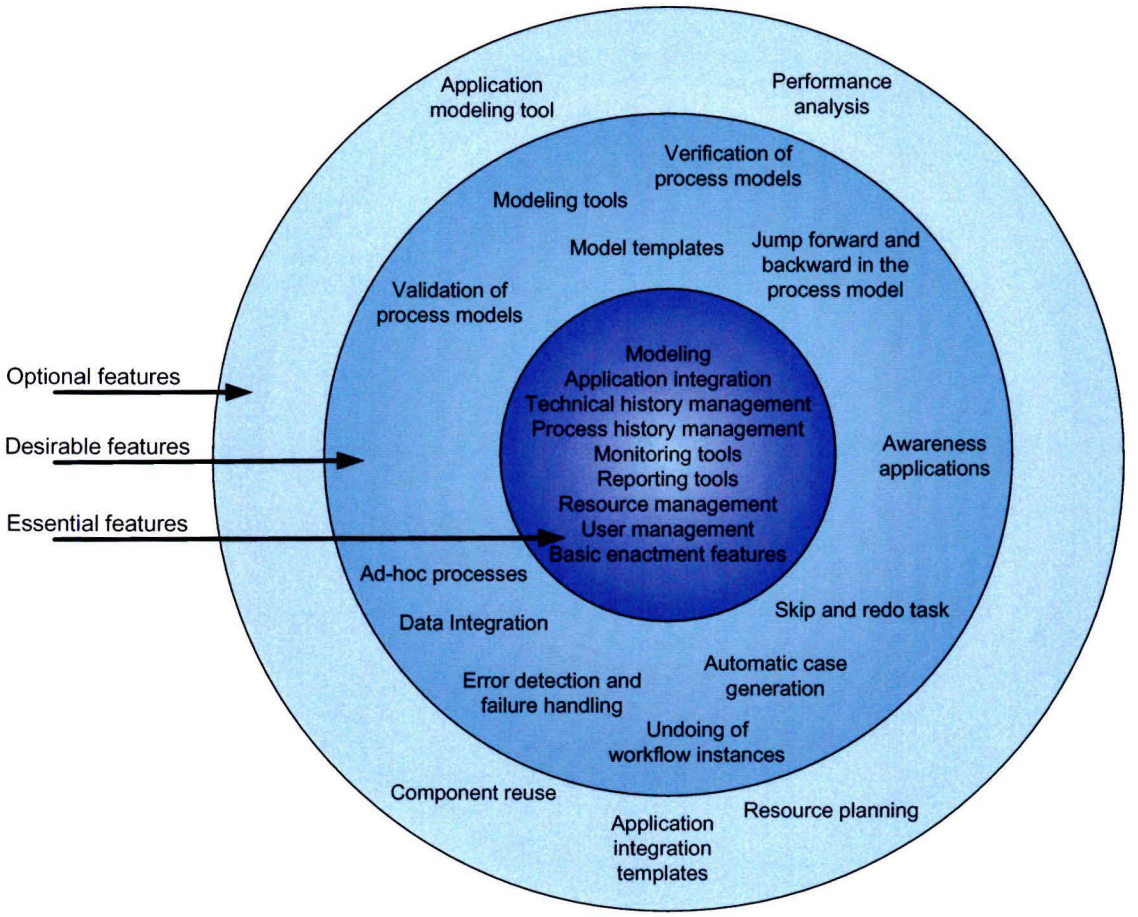

Figure 3.4: Graphical representation of the ideal lightweight feature list

Appendix 16 describes the lightweight WMS Protos Activate using the ideal lightweight feature list. Protos Activate resembles to the ideal lightweight features quite well. It does support all essential features as for the desirable features (except for exception handling features). The optional features are not supported by Protos Activate. Since optional features do not have to be included in a lightweight WMS in any case, Protos Activate is not less lightweight because it does not have these optional features.

3.3 Lightweight features vs. lightweight WMS problems

To determine whether lightweight WMSs work in theory, lightweight WMSs' features should be able to solve the problems presented in section 2.2. In this section, implementation related problems are discussed first. Next, the problems related to footprint are described and last, the problems on organizational changes are investigated.

Implementation related problems

The first problem lightweight WMSs should address is to shorten the time required for implementation. As stated in part I of this thesis, extensive use of data integration requires

customization of the WMS. Customizing is time consuming and consequently lengthens the implementation phase. Standardization can shorten this implementation time. Standardization of WMSs denotes limited use of data integration and additional support of connections to regular, frequently used systems by default. Therefore, lightweight WMSs need to have a suitable data integration feature that reduces customization and thus the implementation time. This concerns both initial and incremental implementations.

Footprint related problems
The second problem lightweight workflow should addresses is the footprint of a WMS. Footprint can be divided into technical, financial and human footprint (section 2.2.2). Fewer features are likely to contribute to a smaller technical footprint. In addition, fewer features reduce the financial footprint because the developments costs are lower. In section 3.2 a division was made in lightweight features and *regular WMS only* features. By eliminating the *regular WMS only* features, resulting in having 27 instead of the initial 33 features, the financial and technical footprint is reduced. These footprints can further be reduced by excluding the five optional features as well. Some features of lightweight WMS can help in reducing the *human* footprint. The exception handling features in the sphere in figure 3.4 can help in reducing the rigidness of a lightweight WMS. They can provide flexibility and the problem regarding human footprint can thus be addressed by the exception handling features. The overall conclusion on footprint of the WMS is that lightweight features can help reducing the footprint problem.

Organizational changes related problems
The problem of supporting organizational changes can be divided into three sub problems: (1) realistic representation of business processes, (2) easy and fast change of process models and (3) interruption of regular flow of work. All sub problems are now discussed separately.
The related feature to realistic representation of business processes is the modeling tool feature. The modeling tool of a lightweight WMS like Protos, can improve communication between the business users and the consultants (who model the process models). Improved communication is likely to contribute to better process models and subsequently to better representation of the business processes.
The related feature to *easy and fast change of process models* is also the modeling tool feature. As described above, a modeling tool like Protos makes it easy and fast to change process models. A tool that is easy to use allows business users to make changes themselves, this saves time because changes can be made instantly and an organization does not have to wait for consultants to make the changes. Lightweight WMS can thus contribute to easier and faster changes of process models if (1) the modeling tool is easy to use and (2) the modeling tool can be used to communicate between business users and consultants (this is necessary when the business user cannot make the change).
The problem of *interruption of regular flow of work* concerns the limited flexibility in WMSs as described in section 2.2. The exception handling features of lightweight WMSs address this problem. They provide flexibility which allows users to e.g. exit and reenter the flow of work. Especially the exception handling features *jump forward and backward in the process models* and *skip and redo task* make it possible to exit and reenter the flow of work. These features provide the flexibility to address interruption of regular flow of work.

3.4 Environments in workflow management

This section looks closer at the environments in which lightweight WMS can be found. By looking at the characteristics of environments a better understanding is obtained on suitable situations for lightweight workflow.

WMSs are used in different environments. Traditionally they were used in environments where work procedures were highly standardized. Nowadays, their application is broadened and can they also be found in less standardized work environments. The situation in which a WMS is used is very relevant in the context of lightweight WMS. Lightweight WMS can contribute to solving problems which regular WMS face in certain situations.
According to the author, a situation is characterized by several aspects, like e.g. complexity of the situation and the volume of information which needs to be processed. A situation has a mix of these aspects and the degree in which e.g. complexity is present differs among situations. These situational aspects are therefore called *dimensions*, to indicate that they can differ and are more or less orthogonal. In this section, seven dimensions are classified in which WMSs can be used. The dimensions are depicted in figure 3.5.

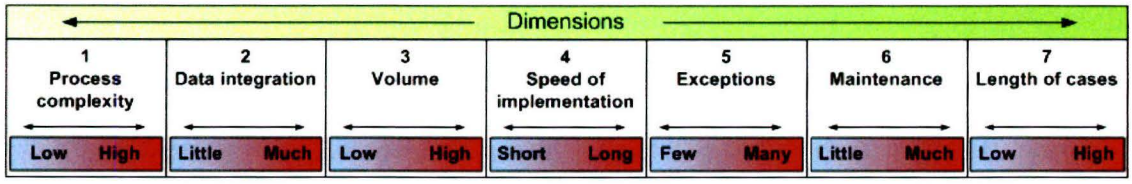

Figure 3.5: Dimensions in workflow environments

The first three dimensions are expected to be very relevant for lightweight WMSs. In the description of the dimensions below, propositions are framed for these three dimensions. These propositions will be considered in the next chapter (section 4.4) where the practical effectiveness of lightweight WMSs is discussed.

Dimension 1: Process complexity
The first dimension is called process complexity. Modeling the business processes is a challenging task and often the modeling possibilities are used to the full extent of the WMS. Consequently, business users have difficulties with modeling the processes. This conflicts with the idea that in lightweight workflow, business users should also be able to model the business processes themselves (see section 2.2.3). For common situations business users can model the process models, yet for complex situations, detailed knowledge of modeling is required. Furthermore, workarounds in modeling are frequently needed in complex situations and the problem appears that business users cannot come up with these workarounds themselves. Hence, modeling has to be done by experts. A proposition that can be made on the relation between process complexity and modeling in lightweight WMSs is that *the more process complexity an environment has, the likelier it is to encounter problems in modeling.* This proposition is considered later in the master thesis in section 4.4 where a case study of a high process complexity environment is researched at a computer manufacturer.
The process complexity dimension should not be mixed up with the exception dimension, described later in this section. Although complex processes are likely to have more need for flexibility and therefore exception handling, this is not always the case. An example of a situation with high process complexity, yet low exception handling, is the case of the computer manufacturer, described in section 4.1.

Dimension 2: Data integration
Situations which score high on this dimension are characterized by their integration with other systems. WMS in these situations have more connections to other systems than usual and this requires intensive use of the features related to data integration. A characteristic of extensive data integration is the customization (e.g. development of interfaces) of the WMS during the

implementation phase. Data integration is limitedly supported in a lightweight WMS like Protos Activate and consequently, when a lot of data integration is needed this has to be developed and configured for that specific situation. As a result of this customization the implementation phase lasts longer. This matter is also described in the interview with Realweb, included in appendix 2. The Chief Technology Officer of Realweb states in this interview that implementations which require customization last longer. For this reason, he claims that lightweight WMS should not be used primary in situations with complex data integration as this eliminates the fast implementation advantage of the lightweight WMS.
The researcher expects that there is a relation between the amount of data integration needed and problems which are encountered during the implementation of a WMS concerning connection to other systems. It can be formulated like this: *the more data integration needed in an environment, the likelier it is to encounter problems when connecting the lightweight WMS to other systems.*

Dimension 3: Volume
Another characteristic in workflow environments are high volume workflow processes. They can be found in situations where many cases need to be processed. High volume workflow processes demand a higher capacity from the workflow server. Another consequence is that business users see a high number of tasks in their task lists. A possible danger is that business users have no overview anymore on their tasks and this can affect proper handling of cases. Some cases can be signaled too late which can cause a deadline violation. This relation between high volume workflow processes and task lists in lightweight WMSs can be formulated in the following proposition: *the more workflow process volume an environment has, the likelier it is that customization of task lists is needed.* This proposition is concerned later in the master thesis in section 4.4 were a case study of a high volume workflow process environment is researched at a mortgager.

Dimension 4: Speed of implementation
The fourth dimension is speed of implementation. In some WMS projects it is crucial to have a fast implementation. Consequently, the focus is on speed of the implementation. An example can be found in a case of a big municipality in the Netherlands. This municipality had a problem with registering naturalization requests. The old system suddenly failed and had to be replaced as soon as possible with a new system to avoid chaos. Since many requests very received everyday, a downtime of a couple of weeks would have been disastrous. The focus of this project was therefore on speed. One approach of fast implementation projects is the so called prototyping approach. Business users often cannot tell in advance what they exactly want with the WMS. The specific requests become clear during the implementation of the system since business users are forced to think about this during the implementation. In the prototyping approach, process models are released earlier in the implementation despite the fact that there are some errors in it. The advantage is an earlier start with the usage of the system so that the specific requests become clear. Process models are adapted frequently until they represent the business processes as they are or should be.

Dimension 5: Exceptions
The fifth dimension is the level of exceptions in the processes. Sometimes the amount of exceptions is very high and therefore the business users have to diverge from the regular flow frequently. This issue is often encountered in a situation with low standardization of work procedures. The dimension on exceptions is different from the first dimension, process complexity, despite the seemingly similarities. Complex processes *can* be standardized to a high amount and therefore have little need for exception handling. There are many ways to deal with exceptions and some are described in appendix 8 (features of WMSs).

Dimension 6: Maintenance
The sixth dimension is related to the maintenance of WMS and can be divided into technical maintenance and organizational maintenance. Technical maintenance is related to the hardware of the WMS or other technical aspects like software updates. An organizational maintenance example is updates of process models due to organizational changes. Lightweight WMS are likely to be used in small organizational entities which do not have much ICT support. The maintenance dimension might therefore be more relevant for lightweight WMS than for a regular WMS.

Dimension 7: Length of cases
The last dimension considers the length of the cases in the WMS. At first sight, this does not seem an important dimension. However, the following example shows that the length of the cases is an important issue, especially for lightweight workflow.

Consider a running WMS in which multiple cases are currently being processed. Processing these cases is done according to the current process model, say version 1. At a certain moment, a change needs to be made in the process model, i.e. two tasks are switched. Now, what happens to the cases in the system that already have executed one of the switched tasks? The cases which are already present in the system cannot be transferred to the new process models with current technology (interview Schilstra, appendix 7). Consequently, they need to be finished according to the old, version 1, process models. As long as the WMS can use two versions of a process model at the same time (one for the old cases, one for new cases), it does not appear that there is a problem. Then again, what if cases have an average processing time of one year and process models change every month? In this scenario the WMS has 10 different versions of the process model. It is evident that this makes maintenance of the WMS quite complicated.

The example illustrated that length of cases can be an important issue in lightweight WMSs. When cases have a long duration this can cause many different versions of process models to be present in the WMS. Since this increases complexity and changes in process models are very likely in a lightweight WMS environment, the length of cases is a relevant dimension.

3.5 Which environments are suitable for lightweight workflow?

Different workflow environments were described using seven dimensions. When a workflow environment scores high on a dimension, e.g. the environment has a high process complexity, this has an influence on the needs or functionality of a WMS. This relation between the dimensions and features is described in appendix 11. The conclusions from the relation between dimensions and features are shown in table 3.2. The results are used next, to indicate which environments are suitable for lightweight WMSs and which not.

An environment with high process complexity needs features like *rule management* and *work balancing* (see table 3.2). Since these features were classified earlier as *regular WMS only*, it is likely that a lightweight WMS is not suitable for an environment with high process complexity.
For environments with much data integration, the features *data integration* and *application integration* are needed. The application integration feature is classified as essential and the data integration feature as desirable and hence both are likely to be included in a lightweight WMS. The conclusion which one would draw from these facts is that lightweight WMSs seem suitable for environments with much data integration. However, this conclusion is not entirely right. One important aspect is not taken into account and that is the amount of data integration which is needed. Environments which demand much data integration are likely to require *customized* connections to other systems which are *not* supported by the data integration feature. The required connections in these environments therefore either cannot be made at all or take much time to make. In the first situation the lightweight WMS cannot be used at all, in the second

situation the long implementation time is an inverse of the lightweight approach of a short and easy implementation stage. In these situations, a regular WMS will perform better. Therefore, environments with much data integration are not suitable for lightweight WMSs.
Environments with high volume workflow processes make use of the *performance analysis* and the *resource planning* features. Since these two features are classified as optional features, their applicability for lightweight WMS is small. Hence, high volume workflow process environments are not suitable for lightweight WMS.
Six features are relevant for environments with a long implementation time. Three of those features (*application modeling tool, modeling templates* and *application modeling templates*) are classified as optional and therefore probably not included in a lightweight WMS. The other three features (*modeling tool, modeling templates* and *data integration*) are desirable and therefore likely to be included in a lightweight WMS. The first three features are not vital for obtaining a short implementation time; they only provide a little assistance. However, the last three features are very important in reducing the implementation time and since they are also likely to be included in a lightweight WMS, lightweight WMSs are suitable for environments with usually have long implementation times. They can help reducing this implementation time.

The last three environments (*many exceptions, much maintenance and long length of cases*) are not checked on suitability for lightweight WMS. The dimensions *many exceptions* and *much maintenance* occur less frequently and are therefore less relevant. For this reason, they are not included. The dimension *length of cases* has no direct relation with features which makes it difficult to test for suitability for lightweight WMS. Moreover this is also a less relevant and occurring dimension.

In the table below, the results of this section are presented. It shows which environments are suitable for lightweight WMSs. In general one can say that when a dimension scores low it is suitable for lightweight WMSs (e.g. low process complexity) and when it scores high, it depends on the type of dimension (e.g. long implementation times are suitable for lightweight WMSs).

Dimension	Relevant features	Classification	Suitable for lightweight WMS
High Process Complexity	Modeling tool Rule management Work balancing	Desirable Regular WMS only Regular WMS only	**No**
Much Data Integration	Data integration Application integration	Desirable Essential	**No**
High Volume Workflow Processes	Performance analysis Resource planning	Optional Optional	**No**
Long Implementation Time	Modeling tool Application modeling tool Modeling templates Component reuse Application integration templates Data integration	Desirable Optional Desirable Optional Optional Desirable	**Yes**
Many Exceptions	Exception handling features	Desirable	n.a.
Much Maintenance	Error detect. and failure handling	Desirable	n.a.
Long Length of cases	n.a.	n.a.	n.a.

Table 3.2: Suitability of workflow environments

Interaction effects between dimensions are not investigated in this master thesis project. This takes too much time and remains a topic for further research.

3.6 Conclusion

This chapter of this master thesis tried to find out whether lightweight workflow can work in theory. The method in doing this was twofold. First, a *lightweight* feature list was made. This list was then used to investigate whether it can solve the problems lightweight WMSs should address. Second, environments in which WMSs are used were investigated to determine the suitability of each environment for lightweight workflow. The conclusions from these two parts are presented below.

3.6.1 Lightweight features vs. lightweight WMS problems

Section 2.3 discussed the theoretical effectiveness of lightweight workflow management by looking at the similarities in features and the problems addressed. The problems were divided into three groups: (1) implementation of the WMS, (2) the footprint of the WMS and (3) the support for organizational changes by the WMS. The section exemplified whether lightweight WMSs, represented by the lightweight feature list, could solve these problems which lightweight WMSs should address.
The list of *lightweight* features was established from a bigger list with general workflow management features. The general list had 33 features in total, of which 8 features were classified as essential for lightweight WMSs, 14 as desirable for lightweight WMSs, five as optional for lightweight WMSs and six as *Regular WMS only*.

Implementation related problems
Regarding implementation related problems one can state that the data integration features can help in shortening the implementation time. By providing a feature which enables the creation of connections to regular, frequently used systems, customization of the WMS can be avoided in many situations. Hence, lightweight workflow can solve problems of a long implementation phase by supporting a suitable *data integration* feature which reduces the customization phase and therefore the implementation time. This concerns both initial and incremental implementations.

Footprint related problems
The footprint can be divided into the technical, financial and human footprint. The *technical* and *financial* footprint can be reduced by including fewer features in a WMS. By eliminating the *regular WMS only* features, resulting in having 27 instead of the initial 33 features, these financial and technical footprints are reduced. These footprints can further be reduced by excluding the five optional features as well. The exception handling features of lightweight WMSs can help in reducing the *human* footprint because they provide flexibility.

Organizational changes related problems
The organization change related problems were subdivided into three sub problems. The first is *realistic representation of business processes*. If modeling tools provide good communication between business users and consultants, it is likely that this attributes to better process models and therefore to better representation of the business processes.
The second sub problem is *easy and fast change of process models*. Lightweight WMSs can contribute to solving this problem if (1) the modeling tool is easy to use and (2) the modeling tool can be used to communicate between business users and consultants. The third sub problem is *interruption regular flow of work*. Lightweight WMSs can solve this problem when they have the exception handling features mentioned in section 2.1. Especially the exception handling features *jump forward and backward in the process models* and *skip and redo task* make it possible to exit and reenter the flow of work.

3.6.2 Suitability of workflow environments

Section 3.4 discussed the environments in which WMSs are used using seven dimensions. These dimensions were used next in section 3.5 to describe the relation between the workflow environments and features of lightweight WMSs. An important conclusion which can be drawn from this is that different environments require different features. It was not possible to determine the suitability of lightweight WMSs for all seven environments. The results are presented in the table below. Three environments (*many exceptions, much maintenance and long length of cases*) were less relevant and therefore not checked on suitability for lightweight WMS.

Dimension	Suitable for lightweight WMS
High Process Complexity	**No**
Much Data Integration	**No**
High Volume Workflow Processes	**No**
Long Implementation Time	**Yes**

Table 3.3: Suitability of workflow environments summarized

In general one can say that when a dimension scores low it is suitable for lightweight WMSs and when it scores high, it depends on the type of dimension. Lightweight WMSs are i.e. not suitable for environments with high process complexity, much data integration or a high workflow process volume. Lightweight WMSs are suitable for environments with high implementation speed.

3.6.3 Does lightweight workflow management work in theory?

Section 3.6.1 in this conclusion showed that lightweight WMSs do address the problems that lightweight workflow should address (see section 2.2). It was possible to solve the problems using the functionality of the ideal lightweight feature list. Therefore, lightweight workflow could work in theory. However, not every environment is suitable for lightweight WMSs. This chapter concluded that lightweight WMSs do not work in environments with high process complexity, much data integration or a high workflow process volume.

Therefore, the overall conclusion on the theoretical effectiveness of lightweight WMSs is that it does work in theory, except in environments with high process complexity, much data integration or a high workflow process volume.

Part III: Do lightweight WMSs work in practice?

4 Practical effectiveness of lightweight WMSs

In the previous chapter the theoretical effectiveness of lightweight workflow was discussed. The third part of this master thesis project looks at whether lightweight WMSs work in practice.

Methodology
The methodology of this chapter is presented in the figure below.

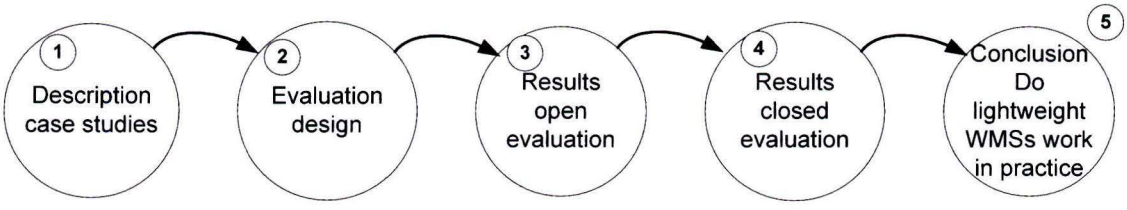

Figure 4.1: Methodology part 3

The first section gives a brief introduction on the case studies used in the master thesis project. In the second section the evaluation of the case studies is explained. The evaluation consists of two parts. The first part is the open evaluation. This part systematically covers all relevant aspects of lightweight WMSs by using a specially developed framework. The framework is explained in section two as well. The second part of the evaluation is the closed evaluation. This closed evaluation focuses on the specific problems that occurred during the implementation phase of the lightweight WMS at the two case studies. Unfortunately, no case studies from literature could be added to this research because case studies on lightweight WMSs do, to the best knowledge of the author, not exist. Both case studies that are used in this research were successful because the lightweight WMSs improved the business processes to a certain extent.

Only the implementation phase is taken into account and not the later phase in which the system is e.g. one year in use. Furthermore, the implementation phase is different for both case studies as depicted in figure 4.2 below. In the figure, only relevant phases in implementation are mentioned.

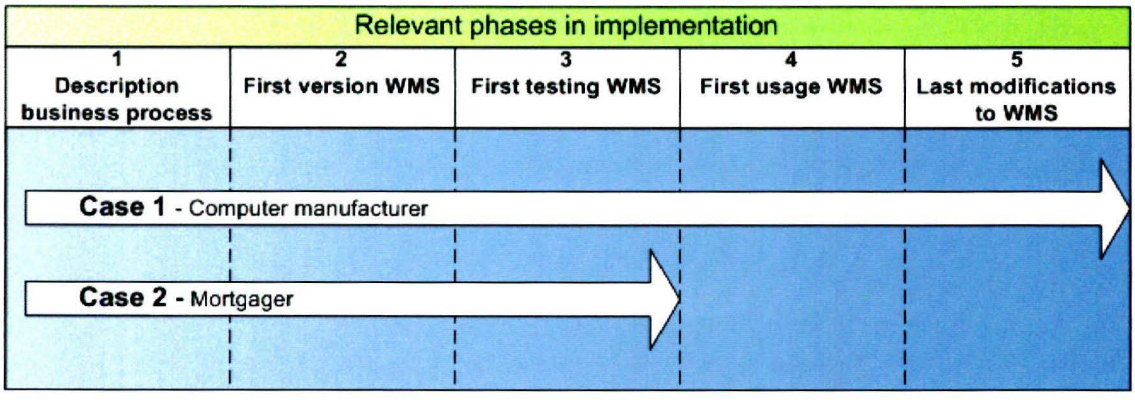

Figure 4.2: Relevant phases in implementation

The implementation phase in figure 4.2 can be subdivided into five parts. The goal of the first part is to acquire a general description of the process in terms of process models and to reveal applications and documents used in the business process. The first phase is an iterative process; first versions of process models are shown to the business users who then give comment for further versions. In the second part of the implementation phase, the process models are actually built. The result is a working lightweight WMS which can be tested by the business users in the next part: testing. First tests of the WMS are done here by a few business users and these first tests should reveal the most common problems with the lightweight WMS. The fourth part is the first usage of the system. All business users use the WMS and thus the WMS is operated extensively in this part. When the system is used for e.g. one month, business users often like to have some things changed or adapted because their wishes become clear after they use the system. These modifications are made in the last part of the implementation.

The fact that only the implementation phase is taken into account during the analysis is a limitation which is beyond control of the researcher. In the case of the computer manufacturer, it was not possible to visit the company and interview e.g. the business users on how satisfied they were after one year of working with the lightweight WMS. In the case of the mortgager, the lightweight WMS was not yet fully in use and for this reason only the first stage could be investigated. Some business users had tested the system and thus worked with it but the actual usage by the whole company will occur after this master thesis is written.
Nevertheless, the results obtained from the case studies are still valuable. Since lightweight WMSs are expected to solve problems especially during the implementation phase (see section 2.1), the obtained results are still useful. Furthermore, it is expected that the problems during the implementation phase give a good impression of the overall problems encountered in other, later phases. For example, a problem like poor performance of data integration with other programs during the implementation phase is also likely to be relevant later during the usage of the lightweight WMS.

In the third section of this chapter the results of the open evaluation with the framework are discussed. The approach in the open evaluation was threefold. First, an interview with one consultant was conducted on the case of the computer manufacturer. Next, a follow-up interview with the same consultant was held on the case of the mortgager. Last, one interview was carried out with another consultant (appendix 5) to validate the results gathered in the first two interviews.
The consultants must have participated in the cases, because knowledge on the cases is necessary. Furthermore, the consultants must have knowledge about the lightweight WMS Protos Activate and the regular WMS FLOWer. This is necessary because in the evaluation a comparison is made between the regular WMS FLOWer and the lightweight WMS Protos Activate.
Unfortunately, it is not possible to involve the companies itself in the evaluation of the case studies. No data can be obtained directly from the companies and therefore it is necessary to rely on the information presented by the consultants. Despite the use of only one type of source (i.e. consultants), the information provided is still valuable for evaluating cases on effectiveness of lightweight WMSs.

The fourth section discusses the results on the closed evaluation of the case studies. The section looks specifically at the problems that were encountered during the implementation phase in the case studies. The information was obtained by interviewing one consultant that participated in both case studies. These problems were again validated with a second consultant (appendix 5). Last, the overall conclusion is presented in the end of this chapter.

Getting to know the lightweight WMS Protos Activate
The author himself has also carried out an implementation of the lightweight WMS Protos Activate. This was done at the marketing department of Pallas Athena. Two business processes were supported with the lightweight WMS, i.e. the content management of the website of Pallas Athena and the failure management of the website. During 12 weeks, the author modeled the business processes in Protos, converted them to Protos Activate and made a working lightweight WMSs. The result is briefly shown in appendix 12.

The implementation at Pallas Athena provided the author with knowledge on Protos Activate. This has helped in understanding the case studies and interpreting the results. It also provided the author with knowledge about practical issues of WMSs in general.
Unfortunately, the implementation of Protos Activate at Pallas Athena could not be used as a case study in this research. The obtained results are not valid because a reorganization at Pallas Athena caused a serious delay in the implementation of the WMS. The lightweight WMS is therefore not yet in use, although it is ready to be implemented. The actual implementation is expected to be finished one month after the time of writing this master thesis. In addition, the implementation of the lightweight WMS was executed without consultants that have extensive knowledge about implementing lightweight WMSs. Since these are normally present in an implementation of Protos Activate, the nature of the implementation phase is different than usual.

4.1 Cases studies description

As mentioned in the methodology, two case studies are used in this research. This section provides a brief description of the cases and lists the main characteristics. The cases are also illustrated using the dimensions in workflow environments from section 3.4. Only the most relevant dimensions are used, i.e. process complexity, volume and data integration. More details on the case studies are included in appendix 13.

Case 1: Computer manufacturer
This case concerns the implementation of a lightweight WMS at a large computer manufacturer, based in The Netherlands. The lightweight WMS used in this implementation is Protos Activate. The implementation concerns the software sales department. The case characteristics are: environment with high process complexity, low volume workflow process and medium need for data integration.
The consultants of Pallas Athena indicated that the case has a high process complexity for several reasons. First, it was difficult to model the business processes. Modeling possibilities of Protos had to be used to the full extent to make the process models. In addition, several workarounds were needed because the modeling tool did not support the required workflow patterns (see appendix 13). Consultants of Pallas Athena indicated (interview Heijink, appendix 4) that they needed about twice the time than they would normally need for modeling.
The case was classified as low workflow volume because not too many cases needed to be handled. According to Pallas Athena, over 2000 cases a year were processed by the lightweight WMSs Protos Activate. This is low in comparison to the second case study were about 5000 cases a year are processed. For this reason the workflow volume is classified as low.
The case was classified as a medium need for data integration because it was not necessary to connect the lightweight WMSs to many other systems. Consultants indicated (interview Heijink, appendix 4) that connections were made to a Microsoft Windows directory, Microsoft Word and Microsoft Excel. These connections were easy to make, except for the connection to Microsoft Excel. As indicated in appendix 13, customization of the lightweight WMS was necessary to make this connection. Since this required considerable effort, the case was classified as medium on the data integration dimension.

In figure 4.3 below, the case is positioned according to the three selected workflow dimensions.

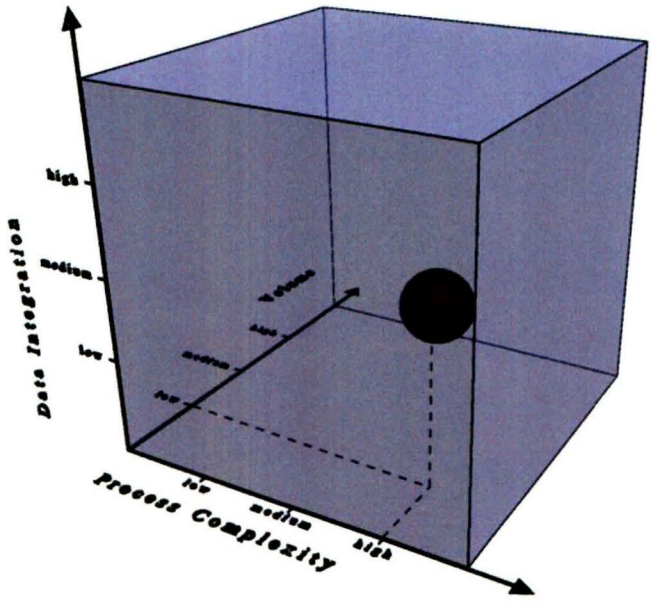

Figure 4.3: positioning computer manufacturer case

The positioning of the case is also presented in table 4.1.

	Process complexity	Volume	Data integration
Computer manfacturer	High	Low	Medium
Mortgager	Low	High	Medium

Table 4.1: positioning of cases studies

Case 2: Mortgager

The second case study concerns a mortgager that is based in The Netherlands. Protos Activate is used in this case as well. The mortgager operates in a business to business environment and does not have direct contact with end users. This case can be described as a high volume workflow process environment. The process complexity is relatively low, and the amount of data integration is medium.

The case was classified as low process complexity. The consultants indicated (interview Heijink, appendix 4) that they could model the business processes easily. Only one workaround was needed in modeling and this was because Protos does not support the use of ad-hoc processes (see appendix 13). Since only one workaround was needed, the case of the mortgager was classified as low process complexity.

The case of the mortgager was classified as high workflow volume because many cases need to be processed. The number of mortgage requests is high: 5000 cases a year need to be processed. This is more than twice the amount of the first case and hence the environment in the second case is classified as high volume workflow process. Unfortunately, no information is available on the maximum amount of cases that can be processed a year. Therefore, future case studies on lightweight WMSs should use more indicators to determine the workflow process volume.

The case was classified as a medium need for data integration. It was necessary to connect the lightweight WMSs to several other systems. Consultants indicated (interview Heijink, appendix 4) that connections were made to a Microsoft Windows directory and Adobe Postscript. The connection to Microsoft Windows directory could be made easily, the connection to Adobe Postscript required more effort. Since Protos Activate does not support Adobe Postscript by default, a considerable amount of customization was necessary (see appendix 13). Therefore, the case was classified as medium on the data integration dimension.
In figure 4.4 below, the mortgager case is positioned according to the three selected workflow dimensions.

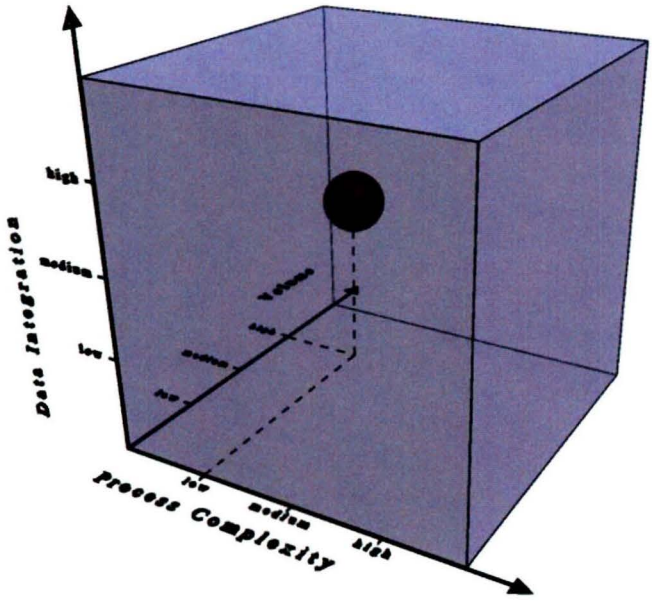

Figure 4.4: positioning mortgager case

4.2 Evaluation design

The case studies of the computer manufacturer and the mortgager are used to determine the effectiveness of lightweight WMSs. This section describes how the effectiveness of lightweight workflow is determined by evaluating the two case studies.

Evaluations can be divided into four types (Van Aken et al., 2007). The first type (1) is Post-test only. This evaluation simply measures afterwards. The second type (2) is comparing a pre-test with a post-test. The situations before and after the change are compared with each other. The third type (3) is a comparative post-test. The comparative post-test is a post-test of which the results are compared to other similar situations, e.g. other countries or other organizations in a similar industry. The last type (4) is a comparative change design. This is similar to the previous type but is extended with a pre-test compared to another pre-test. Thus, the difference in post-test and pre-test is compared with the difference in post-test and pre-test in another similar situation. The types are ordered in increasing strength, the forth type being the strongest.
The type used in the thesis is the first type, post-test only, as the other types cannot be used. Lightweight WMSs is a new topic and to the best knowledge of the author, no earlier similar research has been done. This eliminates the third and fourth evaluation types. Since no pre-test can be done either, the second type cannot be used as well because the pre-test is necessary for

the second (and fourth) type. Yet, despite the disadvantages of the post-test type, the post-test still provides useful information.

The evaluation of the cases studies consists of two parts. The first part is an open evaluation and is described in section 3.3. In this section the cases studies are evaluated with the use of a special developed framework. The second part of the evaluation is a closed evaluation and is described in section 3.4. The closed evaluation focuses on the specific problems that occurred during the implementation phase of the lightweight WMSs. The remainder of this section briefly explains the framework that was used in the open evaluation. The evaluation framework is depicted in figure 4.5 and explained in detail in appendix 14.

Explanation of the evaluation framework
The elements in the evaluation framework are a combination of the problems that lightweight WMSs address, the functionality lightweight WMSs offer and other relevant aspects that come up during the evaluation.
The framework has four columns. The first column contains the different elements in the evaluation. The second column depicts the involved stakeholders or user groups. After all, the elements in the evaluation have an impact on different user groups. The user groups used in this master thesis are *IT department*, *management* and *business users*. Management has different information needs (i.e. management information) and influence than the business users. The business users are the actual users of the WMS. Technical parts in the evaluation, like initial technical installation, are related to the IT department only. The third column contains the evaluation on the different criteria itself and is explained in a qualitative way.
The last column depicts the comparison with the regular WMS and is expressed on a five point scale. A score of one indicates that the lightweight WMS performed poorer than a regular WMS, a score of five indicates that the lightweight WMS performed better than a regular WMS. This comparison is introduced because it emphasizes the objective of this evaluation: Comparing a lightweight WMS to a regular WMS. In addition, the interviewees are reminded constantly to the objective when they need to give a score and they are forced to take a standpoint.
The reason that an ordinal scale is used is that the comparison on an ordinal scale makes it easy to draw a general conclusion. When the average score is below three, it is clear that the lightweight WMS performed worse than the regular WMS. A second reason why a five point scale is used is that a larger scale, e.g. one to 20, suggests a precision in determining the effectiveness which is not appropriate.

The regular WMS used in this evaluation is FLOWer. FLOWer is case-based and thus different than most regular WMSs. Although FLOWer is different, it can still be used as regular WMS in this evaluation. In particular when FLOWer and Protos Activate are compared with each other, FLOWer is unquestionably a regular WMS and no lightweight WMS. Furthermore, consultants on Protos Activate projects also know FLOWer very well, which makes it easier to compare the regular WMS to the lightweight WMS.

4.3 Results open evaluation effectiveness of lightweight WMSs

This section summarizes the results from the open evaluation of the computer manufacturer case and the mortgager case. The approach in evaluating is briefly described first. The results are presented next.
The evaluation was done by interviewing two consultants that were involved in both cases. First, an interview with one consultant was conducted on the case of the computer manufacturer. Next, a follow-up interview with the same consultant was held on the case of the mortgager. Last, one interview was carried out with another consultant to validate the results gathered in the first two

interviews. The validation interview pointed out that the results gathered in the first two interviews were justifiable. Thus, no corrections on the results had to be made. The results of the evaluation are summarized in figure 4.6.
The evaluation results are discussed together for the major part because the results of the first two interviews were almost the same. They differed only on the elements of *involvement of user groups* and *acceptation of user groups* and therefore only these elements will be discussed separately. In appendix 15, the evaluations of the two case studies are discussed in detail.

Evaluation criteria lightweight WMS	**User group**	**Evaluation on crit.**	**Comparison regular WMS**
Problems addressed by lightweight WMS			
Initial implementation			
Technical installation	IT	Text	1 to 5[1]
Configuration of the system	BU, MA	Text	1 to 5[1]
Acceptation by user groups	BU, MA, IT	Text	1 to 5[1]
Involvement user groups	BU, MA, IT	Text	1 to 5[1]
Effort	BU, MA, IT	Text	1 to 5[1]
Duration initial implementation	BU, MA, IT	Text	1 to 5[1]
Incremental implementation			
(Re)Configuration of the system	BU, MA	Text	1 to 5[1]
Acceptation by user groups	BU, MA	Text	1 to 5[1]
Involvement user groups	BU, MA	Text	1 to 5[1]
Effort	BU, MA, IT	Text	1 to 5[1]
Duration incremental implementation	BU, MA	Text	1 to 5[1]
Footprint			
Human footprint	BU, MA	Text	1 to 5[1]
Technical footprint	IT	Text	1 to 5[1]
Effectiveness	BU, MA, IT	Text	1 to 5[1]
Organizational changes			
Representation business processes	BU, MA	Text	1 to 5[1]
Easy and fast change of process models	BU, MA	Text	1 to 5[1]
Interruption regular flow of work	BU, MA	Text	1 to 5[1]
Functionality			
What differs in functionality in comparison to a regular WMS?	BU, MA, IT	Text	n.a.
What are the limitations of the lightweight WMS?	BU, MA, IT	Text	n.a.
What features are related to these limitations?	BU, MA, IT	Text	n.a.
What are potential future problems (limitations)?	BU, MA, IT	Text	n.a.
What features are related to these future problems?	BU, MA, IT	Text	n.a.
Lightweight Specific features:			
Modeling tool	BU, MA	Text	1 to 5[1]
Data integration	BU, MA, IT	Text	1 to 5[1]
Error detection and failure handling	BU, MA, IT	Text	1 to 5[1]
Other aspects			
Other relevant aspects in case	BU, MA, IT	Text	n.a.

Legend
[1] 1 is poorer than regular WMS, 5 is better than regular WMS
n.a. Not applicable
BU User group: Business users
MA User group: Management
IT User group: IT department

Figure 4.5: Evaluation framework

4.3.1 Problems addressed by lightweight WMS

Initial implementation

When comparing the lightweight WMS Protos Activate to the regular WMS FLOWer, Protos Activate performs better during the initial implementation (see figure 4.6). The main benefits the consultants mentioned were faster and easier implementation of the system and better acceptation and better involvement of user groups. A disadvantage of Protos Activate in both cases was related to the configuration of the system. Configuring the lightweight WMS for usage with other systems like Microsoft Excel and Adobe Postscript was more difficult to accomplish in Protos Activate than it would have been in FLOWer.

Involvement and acceptation of the lightweight WMS was slightly better in the case of the mortgager than in the case of the computer manufacturer. The reason for this is that the maintenance in the case of the mortgager was done by the employees of the mortgager and not by Pallas Athena. Technical maintenance was done by the IT department, other maintenance (e.g. changing process models) by management and business users. In order to perform the maintenance of the lightweight WMS by the mortgager itself, its employees had to be involved actively during the implementation to get the required knowledge about Protos Activate. Thus, more involvement of business users leads to better acceptation of the lightweight WMS.

Incremental implementation

The results of the incremental implementation in the two case studies are the same as in the initial implementation.

Footprint

The open evaluation indicated that the human footprint is not smaller for Protos Activate than for FLOWer. This result is unexpected because it does not confirm the proposition from section 3.5 that lightweight WMSs have a smaller footprint than regular WMSs. The reason for the unexpected result is that Protos Activate does not support the *exception handling* features that provide the flexibility for reducing the human footprint.

The technical footprint was larger for the lightweight WMS than for the regular WMS because the *data integration* feature was not supported. This conflicts with the proposition from section 3.5 that lightweight WMSs should support less features to reduce the technical footprint (i.e. less features is better). However, the two case studies show that the technical footprint is larger because some features were *not* supported (thus having more features is better). The issue whether features should be included or not is like a two-edged sword. More features can result to a better performance but on the other hand also to a poorer performance.

Organizational changes

Protos Activate performs generally somewhat better than FLOWer with respect to organizational changes. Modeling in Protos Activate is done with the modeling tool Protos. This tool is easy in usage and understandable for business users. In addition to these advantages, Protos also represents the business processes well. Modeling power in FLOWer is slightly better than in Protos. However, communication to the business users is unquestionable less clear with FLOWer (Interview Heijink, appendix 4). For this reason the consultants indicated that Protos Activate performed better on *representation of business processes*.

On *easy and fast change of process* models, consultants indicated that Protos Activate performs better than FLOWer. When Protos is used for modeling, business users can adapt the process models themselves if they find errors. In FLOWer, the business users cannot model themselves.

On *interruption of regular flow of work*, Protos Activate performs worse than FLOWer. Interruption of the regular flow of work is very easy with FLOWer. In Protos Activate it is not

possible to dissent from the predefined path. Protos Activate scores low on exception handling in general while FLOWer handles exceptions very well.

Evaluation criteria	Results interview with first consultant on case computer manufacturer	Results interview with first consultant on case mortgager	Results validation interview with second consultant computer manufacturer	Results validation interview with second consultant on case mortgager
Problems addressed by lightweight WMS				
Initial implementation				
Technical installation	4	4	4	4
Configuration of the system	2	2	2	2
Acceptation by user groups	4	5	4	5
Involvement user groups	4	5	4	5
Effort	5	5	5	5
Duration initial implementation	5	5	5	5
Average score	**4**	**4,3**	**4**	**4,3**
Incremental implementation				
(Re)Configuration of the system	4	4	4	4
Acceptation by user groups	4	5	4	5
Involvement user groups	4	5	4	5
Effort	5	5	5	5
Duration incremental implementation	5	5	5	5
Average score	**4,4**	**4,8**	**4,4**	**4,8**
Footprint				
Human footprint	3	3	3	3
Technical footprint	2	2	2	2
Effectiveness	2	2	2	2
Average score	**2,3**	**2,3**	**2,3**	**2,3**
Organizational changes				
Representation business processes	4	4	4	4
Easy and fast change of process models	5	5	5	5
Interruption regular flow of work	1	1	1	1
Average score	**3,3**	**3,3**	**3,3**	**3,3**
Functionality				
Lightweight Specific features:				
Modeling tool	5	5	5	5
Data integration	1	1	1	1
Error detection and failure handling	3	3	3	3
Average score	**3**	**3**	**3**	**3**
Total average score	**3,4**	**3,6**	**3,4**	**3,6**

Legend
1 is poorer than regular WMS, 5 is better than regular WMS

Figure 4.6: Results case study open evaluation

4.3.2 Functionality

The functionality aspect of the evaluation is summarized in three subparts. The first describes all present issues which the consultants mentioned in the evaluation. The second subpart describes all future issues for lightweight WMSs. The third part discusses features which are expected to play an important role in lightweight WMSs.

Present issues

When comparing Protos Activate to FLOWer on features, Protos Activate has less functionality in exception handling (see appendix 16). In the mortgager case, e.g. the use of the *ad-hoc processes feature* was needed and a workaround had to be used to overcome the missing of this feature. The regular WMS FLOWer does support all these exception handling features and this makes it more suitable than Protos Activate for an environment where a lot of flexibility is needed. Therefore, one can say that a limitation of Protos Activate is the limited support in flexibility.

Future issues

The consultants indicated in the evaluation that BPM suites are a future trend in workflow management. Pallas Athena is currently busy with their BPM suite and it will contain their programs Protos, Protos Activate and FLOWer. The integration between the lightweight WMS Protos Activate and the regular WMS FLOWer will be increased with the BPM suite. Issues for the future can therefore be found in this domain. Questions like 'how can a company upgrade from a lightweight WMS to a regular WMS' become very relevant. Regarding lightweight WMSs, this is a very relevant topic because this trend can increase sales in lightweight WMSs. Organizations might purchase a lightweight WMS easier if they know that they can extend it later to a regular WMS. It might even take away barriers for purchasing a regular WMS because companies can try a lightweight WMS in an inexpensive and easy way and when they desire, they can upgrade later to the regular WMS.

Another issue which might become relevant in the near future is switching from workflow vendor. Currently, switching from workflow vendor is not easy and often takes much time since the implementation phase has to be done again from scratch. When WMSs are standardized, switching becomes easier. For this reason, standardization of WMSs is also a future topic for lightweight WMSs.
Standardization can also be put in the light of the upgrade perspective. When a lightweight WMS complies with the future workflow standard, it might be possible to upgrade from that *lightweight* WMS to any other *regular* WMS. The benefit for lightweight workflow vendors is that they can market their product as a predecessor for any other regular WMS. Thus, when the company outgrows the lightweight WMS, it is free in the choice for any other regular WMS. Applying to a workflow standard is therefore a worthy consideration and can provide large advantages. The workflow management coalition plays (www.WfMC.org) a very important role in this standardization.

Lightweight specific features

The author expects that some of the features described in section 3.2 play a very important role in lightweight WMSs. These features are *modeling tool*, *data integration* and *error detection and failure handling*.
Modeling in Protos Activate is done with a separate modeling tool called Protos. The process models in Protos are very easy to read and to draw and hence the tool can be used by the business

users. This is an important difference compared to the regular WMS FLOWer as business users cannot model themselves in FLOWer.
The data integration feature in lightweight WMSs is important because when left out, customization of the WMS is needed. Since customization is time consuming, lightweight WMSs should support this feature to some extent. In the cases of the computer manufacturer and the mortgager, the data integration feature of Protos Activate was not sufficient. It did not support connections to the systems Microsoft Excel and Adobe Postscript.
The third feature which was evaluated is *error detection and failure handling*. FLOWer and Protos Activate have similar functionalities in this. However, the consultants stated that error detection in modeling should be investigated when business users model the business processes themselves. This issue is not merely related to the error detection and failure handling feature, it concerns the *validation* and *verification of process models* features. Because business users are no modeling experts, their way of modeling might be inefficient or even wrong. When a modeling tool provides a mechanism which can aid business users in modeling and helps preventing modeling mistakes, better models can be obtained. Currently, some of those checks are made on the models when using Protos or FLOWer and hence this feature is supported already to some extent. Nevertheless, it can be useful to extend this feature further so that process models become better in the future.

4.3.3 Reflection on results open evaluation

In the previous section, several unexpected results were encountered. The evaluation showed that the lightweight WMS performed equal than the regular WMS on human footprint while chapter two and three concluded that lightweight WMSs should perform better on human footprint. Furthermore, chapter two and three concluded that lightweight WMSs should perform better than regular WMSs on organizational changes. However, the results of the evaluation show that the lightweight WMS performs hardly better than the regular WMS on this aspect.

The explanation for the unexpected results is twofold. First, the regular WMS FLOWer, to which Protos Activate is compared, is not a typical regular WMS itself. FLOWer differs itself from other WMSs because it provides more flexibility (Van der Aalst, 2001) than other WMSs. The fact that FLOWer provides more flexibility than other regular WMSs can explain why both lightweight and regular WMSs perform equally on human footprint.
Second, Protos Activate is slightly different than the ideal lightweight WMS. Appendix 16 evaluated Protos Activate on lightweight WMS characteristics and concluded that Protos Activate does not support the exception handling features needed in lightweight WMSs. This can also explain why both lightweight and regular WMSs perform equally on human footprint.

4.4 Results closed evaluation effectiveness of lightweight WMSs

This section focuses on problems that occurred during the implementation phase of the two case studies. This results in an indication of what kind of problems lightweight WMS (could) face in practice. The information on the computer manufacturer and the mortgager case is obtained from interviews with two consultants who have been involved in both implementations (appendix 13 and 15). The consultants were asked to indicate the main problems they encountered during the implementation of the system. In the table below an overview of the main problems is presented and after that, the problems for the cases are discussed individually. A detailed explanation about the problems with illustrations can be found in appendix 13.

Case	Problem
Computer manufacturer	*1) Complex workflow patterns are not supported by the modeling tool of the WMS* *2) Integration with other systems is only limited supported* *3) Customization of the task list is not possible*
Mortgager	*1) No support for ad-hoc processes* *2) Integration with other systems is only limited supported*

Table 4.2: result problem analysis

Case 1: Computer manufacturer
The computer manufacturer case was classified in the beginning of this section as an environment with high process complexity, low volume workflow processes and a medium amount of data integration. When looking at the problems in the computer manufacturer case in the table above, two problems can be explained by the workflow environment.
The first problem, complex workflow patterns are not supported by the modeling tool of the WMS, seems related to the amount of process complexity of the environment. This confirms the proposition made in section 3.4 that *the more process complexity an environment has, the likelier it is to encounter problems in modeling.*
The second problem that can be explained by the environment concerns the integration with other systems. There are problems with integration with other systems. This fact confirms the proposition made in section 3.4 that *the more data integration needed in an environment, the likelier it is to encounter problems when connecting the lightweight WMS to other systems.*
The third problem in the case of the computer manufacturer concerned the customization of task lists. It was not possible to modify the task lists according to the wishes of the customer.

Case 2: Mortgager
The second case was classified as an environment with high volume workflow processes, low process complexity and a medium need for data integration. In this case two main problems were identified. The first problem was the support of ad-hoc processes. The environment required some flexibility because not all possible process flows could be predefined. While the environment required use of ad-hoc processes, this feature was not supported by the lightweight WMS. To overcome this problem, a separate application needed to be programmed.
The second problem was similar to the case of the computer manufacturer and concerned the amount of data integration that was needed. The environment was classified as *medium amount of data integration* and had problems with integration with other systems, i.e. not all connections to other systems could be made by the lightweight WMS Protos Activate. This fact confirms the proposition made in section 3.4 that *the more data integration needed in an environment, the likelier it is to encounter problems when connecting the lightweight WMS to other systems.*
Another proposition that was stated in section 3.4 was that *the more workflow process volume an environment has, the likelier it is that customization of task lists is needed.* This proposition is not confirmed by the case of the mortgager. The environment at the mortgager was classified as high volume workflow processes, yet, there was no problem with customizing the task lists.

4.5 Conclusion

This chapter looked at whether lightweight WMSs can work in practice. Two case studies were evaluated, which was done in two parts. First the conclusions on the open evaluation are presented, next the conclusions on the closed evaluation.

Conclusion open evaluation
The third section of this chapter looked at the performance of lightweight WMSs in comparison to regular WMSs. Therefore, the lightweight WMS Protos Activate was compared with the regular WMS FLOWer. The performance of the lightweight WMS was the best for the incremental and initial implementation part. The consultants indicated that the lightweight WMS performed much better than the regular WMS.
The lightweight WMS performed second best on the organizational changes part. Although the lightweight WMS performed worse than than regular WMS on the subpart *interruption of regular flow of work*, it performed better on *representation of business processes* and *easy and fast change of process models*. Another interesting insight that was obtained is that a higher involvement of business users during the implementation of the lightweight WMSs leads to a higher acceptation of the lightweight WMSs.
An unexpected result was encountered concerning the footprint of a WMS; the lightweight WMS overall performed worse than the regular WMS on footprint. Regarding the *human* footprint, the lightweight WMS performed overall as good as the regular WMS. Regarding the *technical* footprint, the consultants indicated that the lightweight WMSs performed worse than the regular WMS. Both results were unexpected because they do not confirm the assumption from section 2.2.2 that lightweight WMSs have a smaller human and technical footprint than regular WMSs.

The technical footprint was larger for the lightweight WMS than for the regular WMS because the *data integration* feature was not supported. This conflicts with the assumption from section 3.5 that lightweight WMSs should support less features to reduce the technical footprint (i.e. less features is better). However, the two case studies show that the technical footprint is larger because some features were *not* supported (thus having more features is better). The issue whether features should be included or not is like a two-edged sword. More features can result to a better performance but on the other hand also to a poorer performance.

Conclusion closed evaluation
The fourth section of this chapter looked at the problems during the implementation phase of a lightweight WMS. Four problems were identified. The first problem, *modeling patterns not supported by the modeling tool of the WMS,* was encountered in the case of the computer manufacturer. In addition, this problem was also signaled during the implementation that the author carried out at Pallas Athena. The problem seems to have a relation with the amount of process complexity of the workflow environment, because the proposition *the more process complexity an environment has, the likelier it is to encounter problems in modeling,* made in part two, is confirmed in the case study of the computer manufacturer.
The second problem, *integration with other systems is not supported by default*, was encountered in the case of the computer manufacturer and the case of the mortgager. Both cases had an environment with a medium amount of data integration and therefore this problem seems to have a relation with the amount of connections between the lightweight WMS and other systems. The proposition from part two that *the more data integration needed in an environment, the likelier it is to encounter problems when connecting the lightweight WMS to other systems*, is also confirmed in this chapter.
The third problem, *customization of task lists is not possible*, occurred in the case of the computer manufacturer. In part two the proposition was made that *the more workflow process volume an environment has, the likelier it is that customization of task lists is needed.* While the case of the computer manufacture was classified as *low* workflow volume, it did have problems with customization of task lists. Moreover, the case of the mortgager was classified as an environment

with *high* volume workflow process and did *not* have problems with customization of task lists. Therefore, the proposition was not confirmed in this chapter.
The fourth problem, *no support for ad-hoc processes*, was encountered in the case of the mortgager. The lightweight WMS Protos Activate did not support enough flexibility in the case of the mortgager. All possible paths in the process flow had to be modeled in advance while this could not be achieved at the mortgager.

Does lightweight workflow work in practice?
The evaluation showed that the lightweight WMS Protos Activate performed better than the regular WMS FLOWer in both case studies. The lightweight WMS performed better at the initial implementation, incremental implementation and organizational changes part. It only performed slightly poorer on footprint. These results, together with the fact that the lightweight WMS is successfully in use at the computer manufacturer, show that lightweight WMSs can work in practice.
However, regarding the lightweight WMS Protos Activate, one aspect should definitely be improved to enlarge the effectiveness in practice, i.e. flexibility of the WMS. Currently, Protos Activate does not support flexibility enough. It has e.g. no support for the exception handling features mentioned in section 3.2. When it would support these features, the lightweight WMS would perform better on the *organizational changes* aspect in the evaluation. Furthermore, the problem *no support of ad-hoc processes,* encountered in the case of the mortgager, would not have occurred. Thus, Protos Activate should support more flexibility so it can perform better.

Future issues
The evaluation also brought about two important topics concerning the future of lightweight workflow. The first topic is the BPM suite which includes both a lightweight and a regular WMS. In the future it must be possible to upgrade from the lightweight WMS to the regular WMS. Regarding lightweight WMSs, this is a very relevant topic because this trend can increase sales in lightweight WMSs. Organizations might purchase a lightweight WMS easier if they know that they can extend it later to a regular WMS. It might even take away barriers for purchasing a regular WMS because companies can try a lightweight WMS in an inexpensive and easy way first and upgrade to a regular later.

The second topic which might become relevant in the near future is switching of workflow vendor. Currently, switching of WMS vendor is not easy and often takes much time since the implementation has to be done again from scratch. When WMSs are standardized, switching becomes easier. When a lightweight WMS complies with the future workflow standard, it might be possible to upgrade from that *lightweight* WMS to any other *regular* WMS. Applying to a workflow standard is therefore a worthy consideration and can provide large advantages.

5 Conclusion

This master thesis investigated whether lightweight WMSs are effective. Lightweight WMSs were defined first, because no definition of lightweight WMSs exists in literature. Next, the theoretical effectiveness was determined in the second part. Last, the practical effectiveness of lightweight WMSs was discussed in part three. The conclusions of all parts are presented below.

What are lightweight WMSs?
Lightweight WMSs was defined in part one as:

> *Lightweight workflow management systems are workflow management systems that only support basic functionality and are characterized by a short implementation phase and larger involvement of business users during the implementation and configuration phase.*

Furthermore, lightweight WMSs are characterized by six aspects, mentioned in part one. These were (1) less functionality than regular WMSs, (2) Shorter initial implementation phases than regular WMSs, (3) Smaller footprint than regular WMSs, (4) Support of flexibility with respect to work procedures, (5) Easy and fast support of organizational changes and (6) Bottom-up implementation instead of the regular WMSs' top-down implementation. According to the interviews held with Handysoft and Realweb (appendix 1 and 2), the first two are expected to be the most important.
In addition to the definition and characteristics that were presented in part one, part two of this master thesis showed a list of features that lightweight WMSs should have. The features were divided into essential, desirable and optional features and are depicted in figure 3.4.

Do lightweight WMSs work in theory?
The theoretical effectiveness of lightweight WMSs was determined by checking whether lightweight WMSs' features can solve the problems that lightweight WMSs should address. The problems are related to implementation, footprint and organizational changes and are depicted in figure 2.2. The lightweight features are shown in figure 3.4. Part 2 concluded that the lightweight WMSs' features can solve the problems that lightweight WMSs should address in theory.
However, one addition was made. The theoretical effectiveness also depends on the environment in which the lightweight WMS is used. Part two concluded that lightweight WMSs can be used in environments that have a low score on all dimensions or that demand a short implementation time. Though, lightweight WMSs cannot be used in environments with high process complexity, high workflow process volume or in environments where much data integration is needed.

Do lightweight WMSs work in practice?
The practical effectiveness was tested in part three with the lightweight WMS Protos Activate. Two cases studies were used in evaluating the practical effectiveness. The evaluation showed that the lightweight WMS Protos Activate performed better than the regular WMS FLOWer in both case studies. The lightweight WMS performed better at the initial implementation, incremental implementation and organizational changes part. It only performed slightly poorer on footprint. Lightweight WMSs therefore seem to work in practice.
However, for the lightweight WMS Protos Activate, the support of flexibility should definitely be improved to enlarge the effectiveness in practice. Protos Activate does not have exception handling features that are needed for flexibility. When these are supported in the lightweight WMS as well, the lightweight WMSs performs better than the regular WMSs on all elements in both case studies. Thus, the conclusion on practical effectiveness of lightweight WMSs is that

when lightweight WMSs have all features from the ideal lightweight feature list, they are effective in practice.

5.1 Further research

Workflow environments were classified with the use of seven dimensions that could describe the environment quite well. However, when more dimensions are used the environments can be described better. An example of a dimension that could be added is integration standardization. In addition to adding dimensions, the interaction effect between dimensions could be researched. Suitability of workflow environments can be better described when using more dimensions and the interaction effect between them. Moreover, future research should include objective indicators for all workflow dimensions. In this research it was not possible to use utterly objective criteria but when objective criteria for all dimensions can be determined, workflow environments can be described better.

In this master thesis project only two case studies could be used. In addition, in one of these case studies the lightweight WMS was not in use yet. Therefore, more case studies are necessary to determine the practical effectiveness more precise. Hence, future research with more case studies is needed. These cases studies can also be used to determine precisely which environments are suitable for lightweight WMSs when the issues from the previous paragraph are taken into account.
Furthermore, when more case studies are conducted, it is interesting to investigate the way in which they are implemented. For all case studies it is necessary to determine whether they are initiated by middle management or business users (bottom-up) or by top management (top-down). Next, the amount of involvement of business users in the implementation should be determined to see if there is a relation with the acceptation of the lightweight WMSs and the involvement of business users. Moreover, it is interesting to determine the exact relation between involvement/acceptation and human footprint.

In this master thesis a lightweight WMS feature list was made. With this list it is possible to indicate the difference between lightweight WMSs and regular WMSs on a higher level. However, the author and Pallas Athena expect that lightweight and regular WMSs do not only differ on individual features but also within features. For example, data integration is supported by both lightweight and regular WMSs but to a larger extent by regular WMSs. Further research on features is needed to determine the functional difference more precise. This research should consider to what extent individual features are, or should be, supported by lightweight WMSs. The obtained and improved feature list can than be used to precisely determine e.g. which workflow environments are suitable for lightweight WMSs or to what extent lightweight WMSs support flexibility.

Another interesting issue for further research is the support of workflow patterns by lightweight WMSs. Questions like *what basic patterns are supported by different lightweight WMSs* and *to what extent are complex patterns supported* are worthy to research. Workflow patterns might also be used to objectively indicate the process complexity of workflow environments. After all, when many advanced patterns are needed, it is likely that the environment is complex. The author has used workflow patterns in this research as well and discovered that there is no pattern for ad-hoc processes (see appendix 13). Adding a pattern for ad-hoc processes might be necessary to make the list of patterns more complete.

A trend related to lightweight WMSs is the Business Process Management (BPM) suite that could include both a lightweight and a regular WMS. BPM suites are becoming more popular and the

company involved in this master thesis project, Pallas Athena, is developing a BPM suite as well. Future research should reveal how the lightweight WMS can be positioned in this suite. For example, should the lightweight WMS be used as starting point so that the companies first implement the lightweight WMS and upgrade later to the regular WMS? And how can this upgrade from lightweight to regular WMS be done? A related issue was brought up by Van der Aalst in section 2.4. Van der Aalst stated that lightweight WMSs might be embedded in other, larger systems like ERP systems or operating systems. Further research should determine what and how lightweight WMSs can contribute to these systems, how lightweight WMSs can be integrated and how workflow vendors should respond. This is a strategic issue and therefore very important.

The last issue for further research considers switching of workflow vendor. Currently, switching of workflow vendor is not easy and takes much time. Organizations like the Workflow Management Coalition (www.wfmc.org) look at standardization of WMSs so that they are compatible with each other and switching of workflow vendor becomes easier. When a lightweight WMS complies with the future workflow standard, it might be possible to upgrade from that *lightweight* WMS to any other *regular* WMS. Applying to a workflow standard is therefore a worthy consideration and future research is necessary to find out how lightweight WMSs can apply to this standardization.

Last, the author wants to make a special recommendation to Pallas Athena. The comparison of Protos Activate with lightweight characteristics and lightweight features (see appendix 16) has revealed that Protos Activate supports less flexibility than lightweight WMSs should support. This is confirmed by the outcome of the case studies of the computer manufacturer and the mortgager. In both case studies, the human footprint for the regular WMS was similar to the lightweight WMS while a lower human footprint for the lightweight WMSs was expected. When Protos Activate would support more flexibility, the human footprint would probably have been less.
An important remark to this is that support of flexibility is an important distinction between Protos Activate and FLOWer because organizations that demand flexibility now need to purchase the more expensive WMS FLOWer. Thus, when Protos Activate would support the same flexibility as FLOWer, Protos Activate could partly cannibalize the market for FLOWer since organizations can also use the less expensive product Protos Activate.
On the other hand, when flexibility is fully supported in Protos Activate, Pallas Athena might have a superior lightweight WMS product that provides the opportunity for acquiring a large share of the lightweight WMSs market. The profit generated by the increase in market share of Protos Activate might be larger than the reduced revenues in sales of FLOWer. The support of more flexibility in Protos Activate should therefore be considered.

Epilogue

This master thesis was carried out in partial fulfillment of the degree of Master of Science in Operations Management and Logistics. This thesis is particularly interesting for researchers in workflow management or workflow vendors and discusses a new area in workflow management, i.e lightweight workflow management. Since WMSs are used increasingly by smaller organizations, a need has risen for WMSs that are tailored to smaller companies. This research is explorative in nature and tried to define lightweight workflow management and to pinpoint interesting fields for further research on lightweight workflow.

A strong point of this research is the international focus. It was partly done at the Korea Advanced Institute of Science and Technology (KAIST) in South-Korea. Moreover, two Korean workflow vendors contributed by providing their opinion on lightweight WMSs. In addition, the international operating workflow vendor TIBCO played a significant role in the establishment of the WMSs' feature list. A weak point is the fact that only one lightweight WMS could be used. Future case studies with multiple lightweight WMSs are needed to determine e.g. the practical effectiveness more precise.

The author found it very interesting to conduct his master thesis on a new topic and to contribute to the literature in that way. It was very educational, in particular because in the other parts of the master program, the literature is researched but no addition is made. Furthermore, conducting a scientific research individually was something new in the master thesis project and with the completion of this master thesis project, the author has learned to manage a bigger project and e.g. to formulate a research proposal. These skills are very useful in the future career.

When looking back at other parts of the master program in Operations Management and Logistics, the author can say that he was very pleased on the overall quality of his education. Especially the international semester is interesting and provides the opportunity to experience different cultures. The author carried out his international semester in South-Korea at KAIST. This proved to be an excellent place for the preparation of his master thesis project because at KAIST much knowledge was present on workflow management. The author recommends future students in OML to take the knowledge of the university at which they study during the international semester into account in their choice. After all, choosing the right university can help in the success of the maser thesis project!

References

Agostini A. and De Michelis G., (2000a), *A Light Workflow Management System Using Simple Process Models*, Computer Supported Cooperative Work, 9, 3-4, pp 335-363.

Agostini A. and De Michelis G., (2000b), *Improving Flexibility of Workflow Management Systems*, Lecture Notes In Computer Science, 1806, Springer-Verlag, pp 218-234.

Davis F., and Venkatesh V., (2004), *Toward Preprototype User Acceptance Testing of New Information Systems: Implications for Software Project Management*, IEEE Transactions on Engineering Management, 51, 1, pp 31-46.

Dourish P., (1995), *Freeflow: Mediating Between Representation and Action in Workflow Systems*, Proceedings of the Conference on Computer Supported Cooperative Work, ACM Press, pp 190-198, New York.

Fakas G., and Karakostas B., (2004), *A peer to peer (P2P) architecture for dynamic workflow management*, Information and Software Technology, 46, 6, pp 423-431.

Faustmann G., (1999), *Configuration for Adaptation – A Human-centered Approach to Flexible Workflow Enactment,* Computer Supported Cooperative Work 9, 3-4, pp 413–434.

Gartner, (2003), Authored by Sinur, J. and J. Thompson, *Magic Quadrant for Pure-Play BPM, 2Q03 (M-20-0930)*, Gartner Research.

Gartner, (2006), Authored by Hill, J. and Sinur, J., *Magic Quadrant Business Process Management Suites,* Gartner Research.

Georgakopoulos D., (2004), *Teamware: An Evaluation of Key Technologies and Open Problems*, Distributed and Parallel Databases, 15, 1, pp 9–44.

González A., Smari W., Debnath N., (2002), *Toward Flexibility of Workflow Management Systems Based On Task Requirement Classification,* The 2nd International Conference on Information Systems and Engineering, Simulation Series, Vol. 34-2, pp 210-219.

Hollingsworth A., (1995), *Workflow Management Coalition - The Workflow Reference Model (TC00-1003)*, United Kingdom, Workflow Management Coalition.

Kammer P, Bolcer G., Taylor R., Hitomi A., Bergman M., (2000), *Techniques for Supporting Dynamic and Adaptive Workflow*, Computer Supported Cooperative Work, 9, 3-4, pp 269–292.

Mohan C., Alonso G., Gunthor R., Kamath M., (1995), *Exotica: A Research Perspective on Workflow Management Syste*ms, Data Engineering Bulletin, 18, 1, pp 19-26.

Muth P., Weissenfels J., Gillmann M., Weikum G., (1999), *Integrating Light-Weight Workflow Management Systems within Existing Business Environments*, 15th International Conference on Data Engineering (ICDE'99), pp 286.

Norman D., (1992), *Design Principles for Cognitive Artifacts*, Research in Engineering Design, 4, 1, pp 43-50.

Rinderle S., Reichert M., Dadam P., (2004), *Correctness criteria for dynamic changes in workflow systems – a survey*, Data & Knowledge Engineering 50, 2, pp 9-34.

Russel N., Ter Hofstede A., Van der Aalst W., Mulyar N. (2006), *Workflow Control-flow Patterns: A Revised View*, working paper.

Shan Z., Yang Y., Li Q., Luo Y. and Peng Z., (2006), *A Light-Weighted Approach to Workflow View Implementation*, The 8th Asia Pacific Web Conference, 1059-1070, Harbin.

Sonnenberg (2006a), Literature review on lightweight workflow management systems

Sonnenberg (2006b), Master thesis preparation

Van Strien, P., (1997), Towards a methodology of psychological practice, *Theory and psychology*, 7(5), pp. 683-700

Van Aken, J., Berends, J., Van der Bij, H., *Problem solving in organizations*, 2007, Cambridge University Press, Cambridge.

Van der Aalst W., Hauschildt D., Verbeek H., (1997), *A Petri-net-based Tool to Analyze Workflows*, In Proceedings of Petri Nets in system engineering (PNSE), pp78-90, Hamburg.

Van der Aalst W., Ter Hofstede A., Verification of workflow task structures: a Petri-net-based approach (2000), Information systems, 25, 1, pp 43-6

Van der Aalst, W. and Berens P., (2001), *Beyond Workflow Management: Product-Driven Case Handling*, In Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work, pp 42-51, Boulder.

Van der Aalst W., Ter Hofstede A., Kiepuszewski B., and Barros A., (2003), *Workflow Patterns*, Distributed and Parallel Databases, 14, 1, pp 5-51.

Van der Aalst, W., and Van Hee K., (2004), *Workflow management, Models, Methods and Systems*, MIT Press, Cambridge.

Workflow Management Coalition, (1999), *Interface 1: Process Definition Interchange process model WfmC TC-1016-P*, United Kingdom, Workflow Management Coalition.

Weber B., Wild W., Breu R., (2004), *CBRFlow: Enabling Adaptive Workflow Management Through Conversational Case-Based Reasoning*, Lecture Notes in Computer science, 3155, Springer-Verlag, pp 434-448.

Weissenfels J., Muth P., Weikum G., (1998), *Flexible Worklist Management in a Light-Weight Workflow Management System,* In Proceedings of EDBT Workshop on Workflow Management Systems, Valencia.

Zur Muehlen M., (2002), *Workflow-based Process Controlling*: *Foundation, Design, and Implementation of Workflow-driven Process Information Systems*, Logos Verlag, Berlin.

Websites

www.pallas-athena.com, last accessed on 18-08-2007

www.workflowpatterns.com, last accessed on 02-08-2007

www.wfmc.org, last accessed on 02-08-2007

www.adepttechnologies.com, last accessed on 12-06-2007

http://msdn2.microsoft.com/en-us/library/ms735967.aspx, last accessed on 30-07-2007

Index Appendices

PART I: DEFINING LIGHTWEIGHT WMSs

PART II: DO LIGHTWEIGHT WMSs WORK IN THEORY?

PART III: DO LIGHTWEIGHT WMSs WORK IN PRACTICE?

Appendix 1 Interview Handysoft

In this appendix the interviews held at is discussed. First the characteristics of lightweight WMSs are discussed. After that the solutions are discussed.

Facts:
Company: Handysoft
Interviewee: Seokmoon Ryoo
Function: Senior engineer
Date: December 1, 2006
Location: Seoul, South Korea
Expected date of their first lightweight WMS: 2009

Mister Ryoo of Handysoft thinks that lightweight WMSs can overcome some problems which regular WMSs face. In his opinion lightweight workflow can be used in companies with a maximum of 200 users.

Characteristics

In this part specific characteristics of WMSs are listed that can be overcome using lightweight systems. First, performance problems are discussed. Next, the focus is on the customization of a WMS and after that the way in which the implementation is initiated is described. Finally, the way processes are modeled is discussed.

Lightweight WMSs should have specific features according to Mr. Ryoo. First of all, they should support the core functionality of WMSs. They should not only have fewer features than a regular WMS, they are also different in other aspects. Hence making a lightweight WMS is not done by simply removing parts of the regular WMS. Furthermore, the lightweight WMS should be compact. Traditional WMSs often have performance problems. An often heard complaint from customers is that the WMS is too slow. When a user wants to access information this takes too much time if the WMS is very big and this causes unnecessary delays. Users have to wait e.g. a couple of seconds for information they request and when they request information very often (multiple times a minute) this waiting is very disturbing. To solve the performance problems, a lightweight WMS should be fast. Because a lightweight WMS is less complex and handles less data than a regular WMS it is expected that this problem can be solved with lightweight WMS.

Another feature that lightweight WMSs should have concerns standardization. The current WMS of Handysoft, Bizflow, is a 'heavy weight' product. One very important issue in the implementation of their product is customization. For almost every customer the standard version of Bizflow is not sufficient, customers want to have adaptations to Bizflow. Handysoft can realize the customer wishes and customizes Bizflow according to their demands. A big disadvantage of this customization is that it takes quite some time. Not only before Bizflow can be installed customization is needed, also when the system is running customization takes time. Customization plays a big role in the initial implementation and in the incremental implementation. The specific consequences are explained later.

Lightweight WMSs need less customization and can be standardized to a higher degree than regular WMSs. There are several reasons for this. Lightweight WMS are smaller systems with fewer connections to existing legacy systems and this makes standardization easier. No customized solutions are necessary to connect to other systems that are very different in nature. A lightweight system also has fewer features than the regular system and this also contributes to

easier standardization. The situation in which lightweight WMS are used are less complex and when there is less complexity and fewer exceptions, it is easier to standardize.

The method in which a WMS is initialized is also different in lightweight systems. The implementation is bottom-up in nature. A WMS is usually implemented via a top-down approach. This means that the use of a WMS is initiated via top management and not via the actual users of the WMS. The users are sometimes enforced to use the WMS when top management has decided that it should be used. Moreover, when a WMS is initialed by top management the communication in the early phases takes place between higher management and the workflow vendor. A disadvantage of this is that fewer demands from the actual users are taken into consideration in the beginning of the WMS implementation because the users are not directly involved. Because (higher) management levels are between the actual users and the workflow vendor, communication is delayed and fewer details are included. Important details may be omitted and this can be harmful for the acceptance of a WMS. In general a top-down approach leads to lower acceptance then a bottom-up approach.
Lightweight WMSs may enable this bottom up approach. Lower (or middle) management can initiate a WMS if they are lightweight because the impact on the organization is less. Lightweight WMS also have a smaller impact or a smaller footprint. Because lightweight WMSs can be used for a small part of the business, the use of lightweight WMSs becomes easier available for the users of the WMS. This means that the actual users can initiate the use of a WMS and this results in more commitment of the WMS. According to Ryoo a lightweight WMS should be implemented via the bottom-up approach.

Handysoft uses formal notations such as BPMN and BPEL in the designs of the process models. These formal models have the advantage of an unambiguous representation. A disadvantage of using a formal notation is that the business users should first learn how to use them before they can start modeling. Handysoft gives business users approximately 40 hours of training in e.g. BPMN so that they can model the business processes in this notation. Whether this can change in the way in which a lightweight WMS is used is unclear. Handysoft tries to let users model the processes in an easier way with a lightweight WMS. They think of using the Microsoft Office (Word or Excel) for modeling because most users know how to use this.
Handysoft uses a design framework called BP Studio for modeling the business processes. The program uses templates for this and these templates make modeling faster and easier. Templates can make the modeling easier if someone knows how to use them. There is a trade-off between how many processes need to be modeled and the use of design templates.
Since fewer processes need to be modeled in situation where a lightweight WMS is used an easier program for modeling is necessary. If the current BP Studio is used, business users still first need to learn how to use the design templates. When fewer processes need to be modeled this can take too much time. Handysoft therefore thinks of another tool for lightweight WMSs. How this tool, should look like is not clear yet.

Solutions

This section covers the solutions proposed by Handysoft. First, the general implementation cycle of a WMS is described. After that the improvements regarding the implementation are discussed. A distinction is made between the initial implementation and the incremental implementation.

According to Ryoo lightweight WMSs can improve the initial implementation of the system. A general initial implementation is depicted in the figure 1.1.

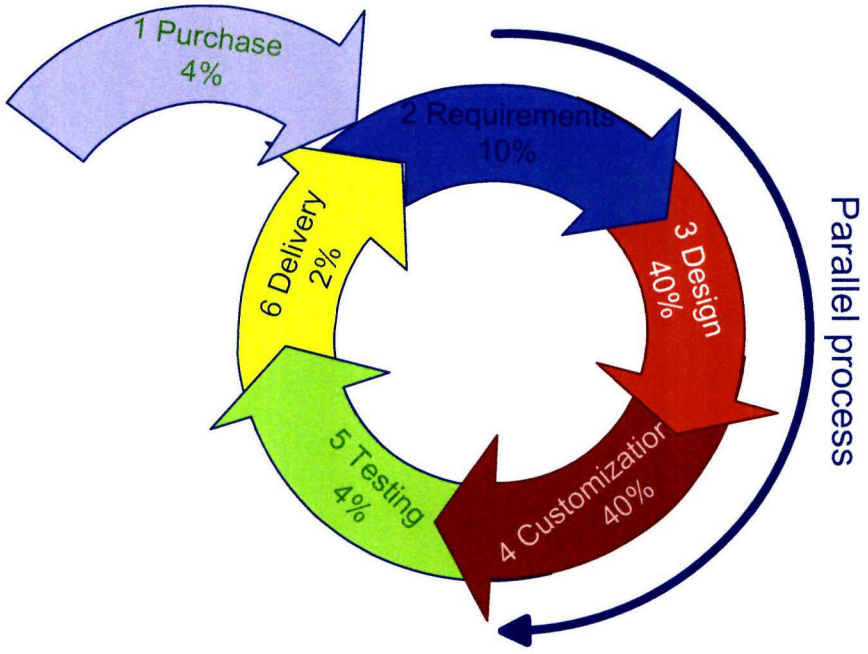

Figure 1.1: implementation cycle

In the first phase, *purchase*, considers the selling of the WMS to a company. In the end of this phase a contract is made with the details of the purchase. The second phase is the *requirements* phase. In this phase the company that bought the WMS gives all the information to the workflow vendor about their needs. With these requirements the workflow vendor can configure the product to the wishes of the customer. If a customer deviates from the standard functions of the WMS than this becomes clear in the requirements phase. Special, not readily available demands should be listed and described in detail during this phase. The more customized features are necessary, the longer this phase lasts. The third phase is the *design* phase. In this phase the process models are designed. Often a third party consultancy firm is used in the design of the process models. The design phase takes usually 40% of the total implementation time. The fourth phase is *customization*. Customers often have specific wishes, often related to already exciting systems in the company. The new WMS should be linked with these systems to fulfill the wishes of the customer. Customization takes a lot of time and brings along lot of complexity. The workflow vendor sometimes has to design modules from almost scratch. The *testing* is the fifth phase. The WMS is tested in this phase and if it performs well it is *delivered* to the customer in the last phase (including the actual installation).

Incremental implementation

Lightweight WMSs can make the initial implementation much shorter and less complex. Lightweight WMS cannot make the first two phases shorter but certainly phases three to five. Ryoo thinks that lightweight WMSs can be highly standardized. This has big benefits in comparison to the regular WMS in that the initial implementation can be much shorter. The requirements phase can be shortened because customization whishes can be left out. These whishes generally are more detailed than the other requirements because the company has to let the workflow vendor know exactly what they want. A higher level of detail is needed then because the requirements are unique. Another issue why the requirements phase can be shortened

is because lightweight WMSs are less complex. A simple system has simpler requirements than a complex system and since lightweight workflow managements systems are less complex than the regular WMS the requirements can be made easier and faster.

If standardization is used in lightweight WMSs than the customization phase can be much shorter. Most of the time in the customization phase is caused due to special wishes from customers. These wishes demand features which are not readily available in the WMS. If there are less or no customization requests then the only thing that needs to be done in the customization phase is building/assembling the WMS with already designed and available features. Hence a standardized lightweight WMS can reduce the initial implementation time by reducing the customization part.

One approach about the design time in lightweight WMSs is that the design phase can not be shortened very much when lightweight systems are used. The design time will only be shorter because less processes have to be designed and not because of a different approach. Only if for example an easier design tool is available the design time can be faster. This tool is, however, not available yet. The other approach says that the design time will be shorter in lightweight WMSs because the design of the process models will be easier. Situations in which lightweight WMS can be used are less complex and therefore process modeling is easier. Business professionals themselves can model the processes in that case which saves time.

The last 2 phases in the implementation cycle are together only 6% of the whole projects time. Ryoo expects that the testing and delivery phase are not shortened by lightweight WMSs.

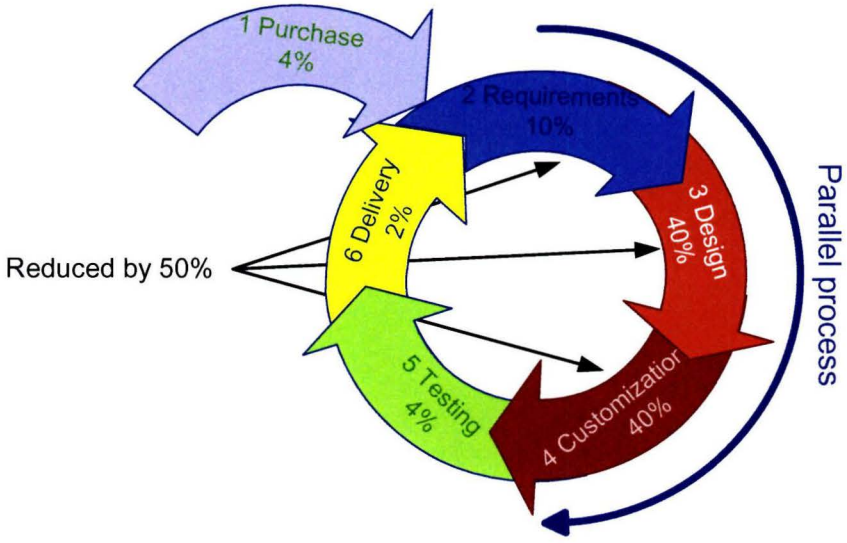

Figure 1.2: Implementation cycle with improvements

Incremental implementation

The implementation cycle is repeated about every year from step two to six. This is necessary because the organization changes. When changes in the WMS are necessary, the time between giving the order to make changes and the completion of the changes is very long. It is quite common that the time between the actual usage (delivery) of the changes and the signaling of the changes is three to four months. So the time between the moment on which the users say what

needs to be changed and the moment on which the changes are implemented is three to four months. This is because first a request has to be send to the workflow vendor by higher management (the requirements phase), then the workflow vendor needs to make the actual changes. This takes usually a long time because WMSs are often customized and furthermore companies want customized changes, e.g. a link with their very own software package for financial services. When a lot of customization is used, making changes is very time consuming. After the customization phase the system needs to be tested before it can be implemented. All together this process can take three to four months.
When lightweight WMSs are used that need no customization because of standardization, the initial implementation time can be reduced considerably. The initial implementation can be reduced even more if the business professionals themselves can make changes in the process models. The higher management does not have to send requests to the workflow vendor (phase two) and the workflow vendor does not need to execute phases three and four (design and customization). Instead, changes can be made by the business users themselves when necessary and tested immediately. The reduction in time for initial implementations is in this case very big.

Organizational changes
A WMS needs updates, patches, every now and then. A big disadvantage of updates is that the workflow system has to be shut down for this. While the system is shutdown the users cannot do their regular work and this makes shutdowns costly. Some organizations can bypass this problem by doing updates at night so fewer users will be affected. The problem is, however, not solved by this and still exists. Another big problem when a system is shutdown is the loss of valuable data. The data not stored in databases at time of a shutdown, called current data, is lost after a shutdown and cannot be recovered. Handysoft thinks the problem of shutdowns can possibly be overcome by lightweight WMSs. The WMS does not need to be shutdown for updates or patches when a certain technique is used. This technique will be explained now.

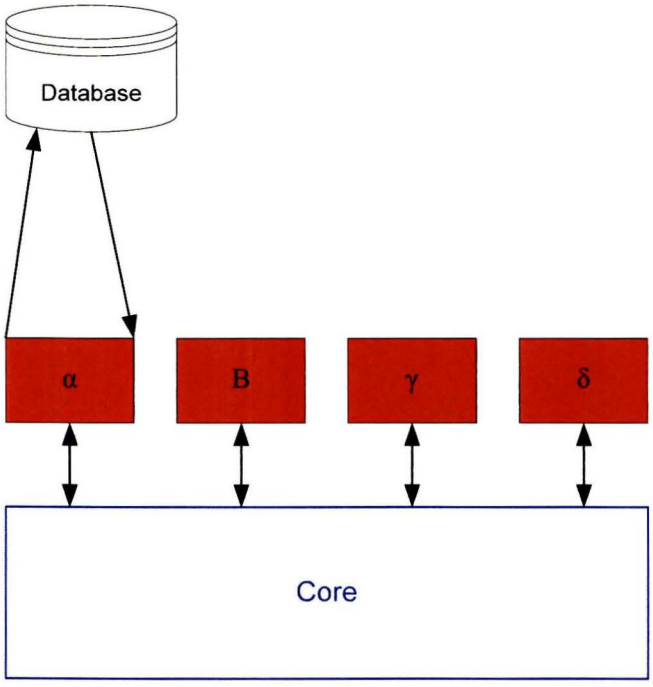

Figure 1.3: General architecture

Figure 1.3 shows a very general architecture of a WMS. The big rectangle represents the core of the system, the kernel. On top of that, the other components of the WMS are built (α, β, γ and δ). These are the other functionalities of the WMS. The data is stored in the database, also shown in the figure. For simplicity reasons only one database is used and only one connection with the components is depicted. Currently, updating a component without a showdown of the system is not possible because backup of current data is not possible. The dataflow is very complex in regular WMSs. One of the reasons for this is the exchange of information with other legacy systems. Data flows are becoming very complex when other systems besides the regular WMS are used and therefore backup of the current data is not possible. The current dataflow is too complex for storage. When a standardized lightweight WMS is used that does not have many connections to other systems, the backup of current data becomes easier. The figure below shows the basic idea of this backup system of current data and how a WMS can be updated without a shutdown.

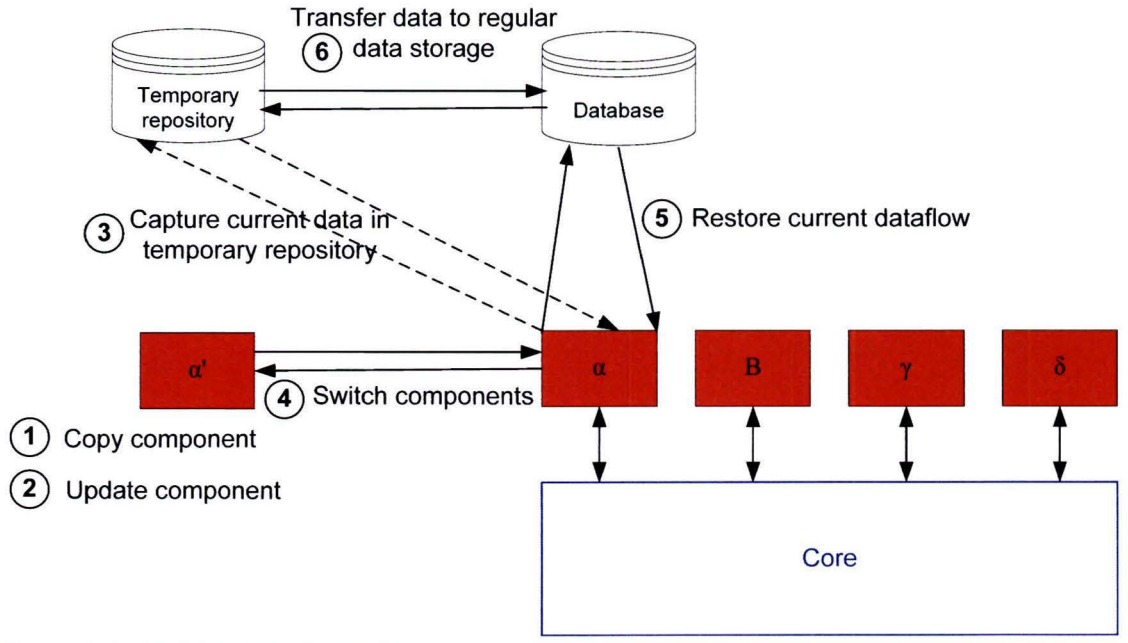

Figure 1.4: Architecture for updates

The component which needs to be updated, called α, is first copied. Second, this copy is updated, resulting is a new version, called α'. The third step is to capture the current data of α in a repository for temporary storage. The current data is not lost in this way; it is stored in a temporary place. The fourth step is the switch of component α with α'. When this new component α' is in place, the dataflow is redirected again as normal (step five) and component α' has now fully replaced α. The data in the temporary storage is stored in the right databases and the update is done.

A big advantage of a technique like this is that updating without a shutdown makes updating more secure. There is less risk of losing data when storage of the current data is possible. Another big advantage is that users can keep working with the WMS when updates are implemented.

Appendix 2 Interview Realweb

Facts:
Company: Realweb
Interviewee: H.C. Jeon
Function: Vice President and Chief Innovation Officer
Date: December 1, 2006
Location: Seoul, South Korea
Turnover: 5.000.000 US $
Number of employees: 30
Expected date of their first lightweight WMS: beginning 2008

Mister Jeon of Realweb thinks that lightweight WMSs can overcome some problems that regular WMSs face. In his opinion, lightweight workflow can be used in small as well as in bigger companies.

Characteristics

In this part specific characteristics of WMSs are listed which can be overcome using lightweight systems. First, the current products of Realweb are briefly described. Next, other characteristics of lightweight systems are discussed.

Realweb currently offers three packages; process analysis, the workflow system and performance management system. The packages can be used together and this is what is usually done by the customers. The figure below shows how the three packages can be used together.

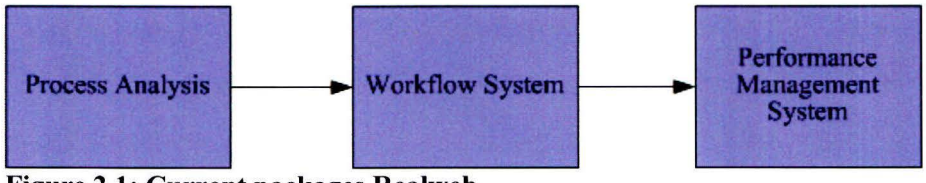

Figure 2.1: Current packages Realweb

When the new lightweight WMS of Realweb is used the customer can still use the process analysis and the performance management system. Figure 2.2 shows this. In this case the lightweight WMS will be used instead of the regular WMS. Details of the new lightweight WMS of Realweb will be explained later.

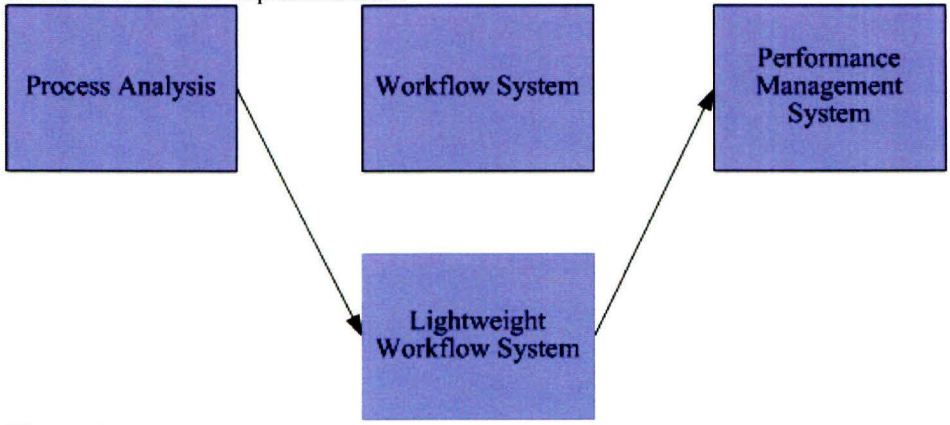

Figure 2.2: New situation

Jeon has ideas about how a lightweight WMS should look like. In his vision the processes should be directly executed by the lightweight WMS. This is also shown in figure 2.2; the lightweight WMS is directly coupled to the process analysis block. Another idea concerns the maintenance of WMSs. Maintenance can be seen as incremental implementation. In Jeons view the business users themselves should be able to maintain the lightweight WMS. They should be able to do this without technical expertise from the workflow vendor or a third party consultancy firm. The third idea is that an easy to use and draw process modeling technique should be used. Like in Realwebs regular WMS, the modeling technique should be very simple. Realweb developed their own, easy to draw, modeling technique. A disadvantage of this easy to draw approach is a somewhat ambiguous representation. Realweb, however, thinks that this easy modeling technique is better for their product than a formal, less easy to understand notation, such a BPMN. It is possible to compare Handysoft in this: they use BPMN and BPEL and give the new users 40 hour of training. Realweb only gives approximately one hour training because this saves time. There is always a trade-off between time and formality of the language used. The fourth idea is that processes should be executed with more flexibility. This idea is strongly related to flexible workflow management. The idea of flexibility can also be found in the lightweight WMS Milano.

Architecture

A general architecture in WMSs is shown in figure 2.3. The number of layers which can be distinguished in this architecture is three. The top layer, the workflow model, represents the new lightweight WMS of Realweb. Several processes are depicted which represents the tasks. The second, or middle, layer represents the adaptors to which the first layer can be connected. Adaptors are *pieces of software that are used in the integration of component-based applications and serve as a "wrapper" that mediates access to an application that was not developed with integration in mind, including legacy applications* (Enterprise Integration Council). The third layer represents the legacy systems.

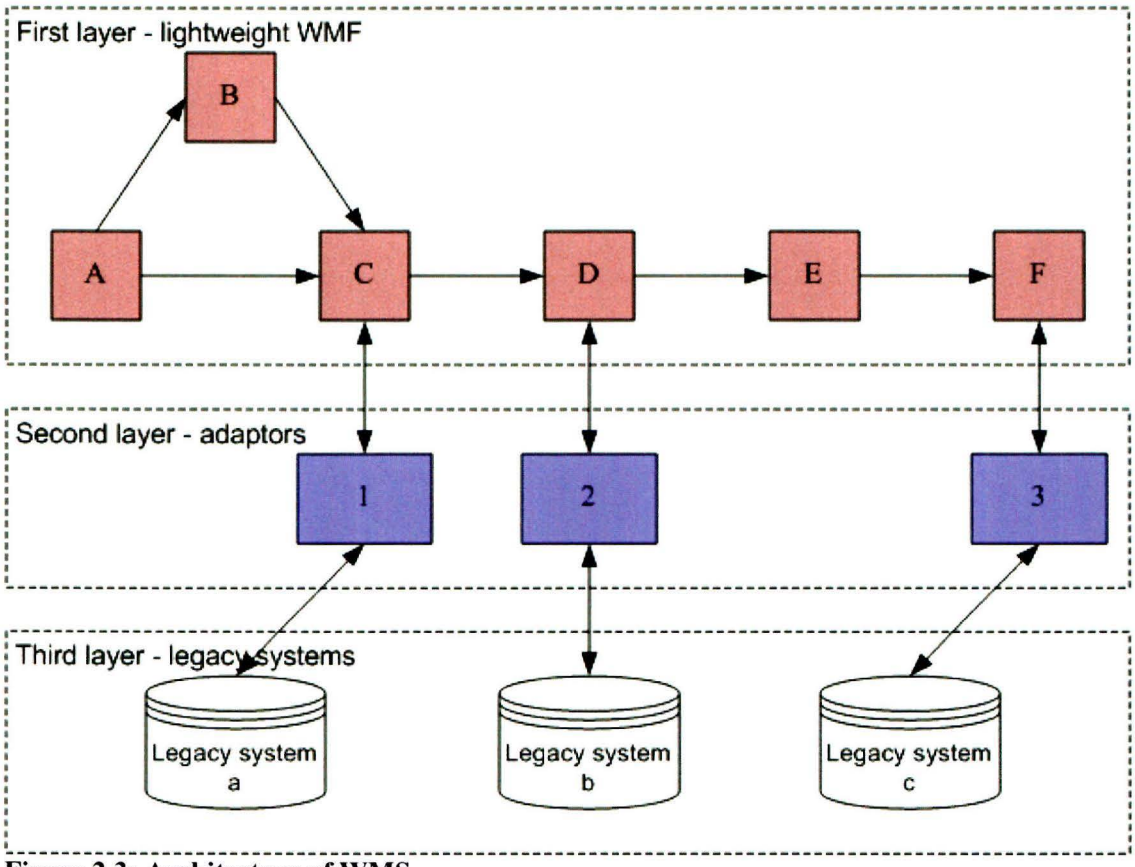

Figure 2.3: Architecture of WMS

Usually a WMS is integrated with legacy systems and therefore is connected to all layers. The new lightweight WMS of Realweb is, however, only the top layer and has no connections with the other layers. The problem with integration with other layers is that it usually takes long time before information can be retrieved from these other systems (e.g. a delay of a couple a seconds). This may lead to performance problems of the whole system. The new lightweight WMS of Realweb does not integrate with other legacy systems. This should make the lightweight WMS faster and make the initial implementation easier. When no connection is made with other systems the total WMS is less complex. Realweb calls the new lightweight WMS therefore 'stand alone'.
Nevertheless, it is sometimes necessary to make a connection with other systems and therefore with the adaptors layer. In this case the lightweight WMS needs to be customized so that this connection can be made. The idea of Realweb is to not customize in general, only when it is necessary (when a customer demands it). When too many adaptations to the lightweight system need to be made, a customer should choose for the regular, heavy weight, version of the WMS. This is because the lightweight WMS has to be customized too much and hence using the lightweight system is not possible.

With these ideas in mind, Realweb recently started with the development of their own lightweight WMS. It is expected that this lightweight WMS is ready in the beginning of 2008. There are some differences between this new lightweight system and the regular WMS. The new lightweight WMS does not handle real business data, only human steps are considered. Real business data can be found in the legacy systems of a company. In the new lightweight WMS there is less functionality which enables integration with other systems. The new lightweight system of

Realweb handles only the human interaction, presented in the first layer in figure 2.3. Another issue is human oriented workflow. The new lightweight WMS focuses on the human tasks. Flexibility is very important in this because the human approach is not rigid but dynamic. The new lightweight system focuses on the administrative processes and not on the production oriented processes.

Realweb also recognizes the bottom-up approach in initialing a WMS. Like Handysoft they think that a lightweight system is not initiated by top management (top-down). Instead the business users feel a need for a WMS and via this bottom-up approach the workflow system is initiated.

Solutions

The regular WMS of Realweb has an initial implementation of three to four months. It looks very much like the implementation cycle described at the Handysoft interview; the same phases can be recognized. Realweb thinks that the initial implementation of their new lightweight WMS can be done in one week if the following requirements are met. The business processes of the customers company should be standardized. Furthermore, the business processes should be stored, for example in a database, where they are available for usage. The information in the database should also include the different roles and procedures used in the company.

The new lightweight system of Realweb is expected to have the following features. It can be installed by the business users themselves, using only one CD. No technical experts from a consultancy company or the workflow vendor are needed for the initial and incremental implementation. Users can directly begin with modeling of the business processes. As said before, the new lightweight system has no connection with legacy systems in default. This should solve e.g. performance problems and enable updates without a shutdown. Only some connections with legacy systems are possible if a user really wants to customize the system. If too many connections need to be made the regular WMS has to be used.

Appendix 3 Topic list Interview Van der Aalst

Interviewee:	W. van der Aalst
Company:	Eindhoven University of Technology Departments of Mathematics and Computer Science
Function:	Professor
Date:	June 22, 2007
Location:	Eindhoven, The Netherlands

In this interview a semi-structured approach was used. The topic list below shows the specific questions in the interview.

Topic list

- Short introduction of the Master thesis project.
- What is lightweight workflow management according to you?
- What is you vision on lightweight workflow? What do these systems look like in ten years?
- What are the main advantages of lightweight workflow?
- What are the main disadvantages of lightweight workflow?

Appendix 4 Topic list Interview Heijink

Interviewee: G. Heijink
Company: Pallas Athena
Function: Consultant
Date: April 26, May 30 and June 11, 2007
Location: Apeldoorn, The Netherlands

In all interviews, a semi-structured approach was used. The topic lists below show the specific questions in the interviews.

First interview
Topic list

- Short introduction of the Master thesis project.
- What is lightweight workflow management according to you?
- What is you vision on lightweight workflow? What do these systems look like in 10 years?

Implementation computer manufacturer case

- How did the implementation of Protos Activate at the computer manufacturer look like?
- What problems were encountered during this implementation?
- What can be done better next time?
- What limitations of Protos Activate were encountered?
- What are the main disadvantages of lightweight workflow?

Second interview
Topic list

- What features are important for lightweight workflow management?
- How would you describe the footprint of a WMS?
- Which elements can be distinguished in footprint?
- Evaluation performance lightweight WMSs.

Implementation mortgager case

- How did the implementation of Protos Activate at the mortgager look like?
- What problems were encountered during this implementation?
- What can be done better next time?
- What limitations of Protos Activate were encountered?
- What are the main disadvantages of lightweight workflow?

Third interview
Topic list

- Evaluation cases on the computer manufacturer and mortgager. The framework in section 3.3 was used for this.

Appendix 5 Topic list Interview Knaapen

Interviewee: F. Knaapen
Company: Pallas Athena
Function: Consultant
Date: June 14, 2007
Location: Apeldoorn, The Netherlands

In the interview, a semi-structured approach was used.

Topic list

- Validation evaluation cases on the computer manufacturer and mortgager. The framework in section 3.3 was used for this.
- Validation problems encountered during implementation on the computer manufacturer and mortgager

Appendix 6 Topic list Interview Muth

Interviewee: P. Muth
Company: Max-Planck-Institution for Information systems Department of Databases and information systems
Function: Researcher
Date: July 26, 2007
Location: Via e-mail

The interview with Muth was done via e-mail. The questions which were presented are shown below.

Topic list

- Short introduction of the Master thesis project.
- Lightweight workflow management can be interpreted in many ways. In what respect can workflow be lightweight according to you?
- Which problems can be addressed (solved) by lightweight workflow?
- How can lightweight workflow address these problems? What solutions would make this possible?
- To what has the Mentor-Lite system resulted? What are the main findings with respect to lightweight workflow?
- Has the Mento-lite system ever been used in practice?
- What is you vision on lightweight workflow? What do these systems look like in ten years?

Appendix 7 Topic list Interview Schilstra

Interviewee: IJ. Schilstra
Company: Pallas Athena
Function: Consultant
Date: May 22, 2007
Location: Apeldoorn, The Netherlands

In the interview, a semi-structured approach was used.

Topic list

- General issues during the implementation of Protos Activate.

Feedback concept thesis

- Total cost of ownership of a WMS

Appendix 8 Features of WMSs related to interfaces

This section describes features of WMSs. All features are categorized according to the workflow reference model. First, the five interfaces of the workflow reference model are discussed, next the Workflow Enactment Service. The main features are presented in bold characters, their sub features are underlined. In the end of this section an overview is presented in figure 8.2 of all features in relation to the workflow reference model.

Interface 1

Modeling

Interface 1 always contains at least one feature, e.g. modeling of processes. This is a basic feature which is used to capture business processes into process models. Since this feature is present in every WMS and cannot be left out we do not elaborate on this. The remainder of this section discusses the more advanced features of interface 1.

Modeling tools

Business processes can be modeled into a workflow (or process) model using a modeling tool (Workflow Management Coalition, 1999). In order to create a workflow model a modeling tool can be used. There are many different modeling tools. Most tools are graphical modeling tools (Gartner, 2006) and they provide a visual representation of the process model. Some are very sophisticated and offer a lot of extra functionalities (e.g. like verification and validation which are described below) while others are basic. Modeling tools use different modeling languages, depending on the language which can be used by the corresponding WMS. Some modeling tools use a formal language such as BPMN or Petri Nets, others use a non formal language. Formal languages have the advantages of an unambiguous representation but have the disadvantage that they are less easy to use and master than non formal languages. Another way to distinguish in modeling languages is by looking to the complexity. Some languages are difficult to read and require coding (BPEL), others are easier to read and are based in graphical modeling (BPMN).
A relevant issue in workflow management research is user friendliness of modeling tools. Historically, workflow and process models were made by consultants or IT experts. Nowadays these models are made more often by the business users themselves. When business users model the business processes themselves, they have the tendency to model the process exactly like the process is in reality. A possible disadvantage of this is that the processes are less redesigned or improved. When a business process is modeled by an outside consultant, the business process is not only modeled as it is in reality, the business process is also redesigned and optimized. In summary it can be said that a formal language does not only give a better and unambiguous representation of the process models, it also indirectly enables better business process redesign by stimulating modeling by outside parties (like consultants).
Nowadays, modeling tools often provide two perspectives; one for business users (which is more user friendly) and one for IT experts (which provides more technical details). An example of modeling tool is Protos. Protos is a modeling tool which can be used by the business users themselves.

> Validation of process models (basic simulation)
> Workflow models have to be validated before they are used. Validation checks whether the process model represents the business process in a good way. Validation is often done by interviewing the business user but can also be done by basic simulation. In this case the process model is simulated with a number of cases to check whether the model works.

Verification of process models
Another feature related to modeling is verification. This feature is less common in WMSs (Van der Aalst and Ter Hofstede, 2000). Verification tests the model for correctness or soundness e.g. no deadlocks of dead tasks). An example of a verification tool is Woflan (Van der Aalst et al., 1997). Woflan is an analysis tool which can find errors in the process model. The tool can be used with TIBCO BPM, COSA, Meteor and Protos. Verification techniques are outside the scope of this master thesis and will not be discussed into further detail.

Performance analysis (advanced simulation)
Another common feature of modeling tools is performance analysis. Performance analysis uses different simulation techniques to predict the behavior and short term impact of a model. Simulations are used to determine practical issues like future bottlenecks or availability of resources (Van der Aalst & Van Hee, 2004). Simulation is also used for validation of the process models. In this case a number of cases are simulated to see whether the model works. In simulation for performance analysis, the process model is checked for throughput time, utilization rate, waiting time etc. Simulation can be used in performance analysis for estimating key performance indicators.

Advanced Rule management

Another feature is rule management (TIBCO BPM). Process models often involve rules, for example claims have to be approved by the manager if they are higher than 1000 euro. If a lot of rules are used in a process model it might be possible that they conflict with each other. An advanced rule management feature helps in avoiding conflicts by e.g. testing rules and checks whether the process model is complete considering rules. An advanced rule management can furthermore indicate which rules possible overlap and whether it is necessary to change rules. Rule management systems make the usage of rules easier and thus also available for non programmers. An additional feature to rule management is a repository where all rules are stored for re-use.

Model Templates

When a new business process needs to be modeled this process often is similar to and older or already existing process. Less modeling effort is needed when this old process can be reused and hence some WMSs provide a database with process models. Even between different companies there is a considerable overlap in process models and for this reason workflow vendors often provide modeling templates with there product. Model templates do not only decrease modeling time when the WMS is already in use (incremental implementation), they are also useful when implementing a new WMS (initial implementation). To illustrate this consider the following example. A workflow vendor implements a new WMS in municipality X. All business processes are modeled and stored in a database. When the workflow vendor wants to implement their WMS in another municipality, say Y, than they can use the process models from their database. Since business processes between municipalities are very similar, the use of model templates can save a lot of effort and time. A feature of many WMSs nowadays is therefore to export, import, translate and share process models.

Component reuse
Workflow vendors like TIBCO use Service Oriented Architecture (SOA). This implies the usage of components, also in process modeling. A process model is composed out of a number of components. Components can be e.g. sending invoice, financial check, order

conformation etc. These components are often similar in different process models and therefore reuse of the components can make the modeling phase much faster.

Interface 2 and 3

Interfaces 2 and 3 provide the mechanism for communication between the workflow engine and other applications. Interface 2 addresses the client applications, interface 3 the invoked (automated) applications. Interface 2 and 3 are strongly related and therefore will be discussed together.

Application integration

Application integration is a common feature of WMSs and enables the workflow system to invoke external applications. Examples of applications are Word processing applications (Zur Muehlen, 2002). The extent to which applications can be invoked differs. Furthermore, application integration sometimes has to be done by IT experts and sometimes by the business users themselves. This depends on the complexity of the integration.

Application modeling tool

Like process models, application integration can also be modeled. An example of application integration is forms used in a business process. A form may be needed in a certain process in which a user needs to fill in information. With an application modeling tool for forms the creation of forms can be made easier. An advantage of this is that business users can make forms themselves and that no IT experts are needed. When the application modeling tool is integrated in the BPM suite a company can start using it immediately. When not integrated in the BPM suite a company first has to look for an application for e.g. form modeling. Consequently the implementation of a WMS is expected to be shorter when a application modeling tool is integrated in the BPM suite.

Application integration templates

This feature is similar to the model templates feature. One can have templates for application integration in a database which helps to speed up the implementation phase of the WMS. Examples of templates are connection to a SAP system or a form for a common situation.

Automatic case generation

In every WMS cases can be generation manually. This is a basic feature, also briefly described at the basic features of the enactment service. A related feature is automatically generation of cases. The following example illustrates this. When a company uses a Customer Relation Management system (CRM system) this system is used in customer contact. Hence, when a customer calls the company to place an order, this is registered in the CRM system. An example of an automatically generated case is when the CRM system makes the case for the WMS. In this situation the CRM system generates the case for the WMS automatically.

External invocation

A strongly related feature is the external invocation (Zur Muehlen, 2002). External invocation means that the workflow engine can be used for services from outside parties. These outside parties can evoke a process in the WMS and e.g. pass on data to a process instance. Examples of ways in which outside parties can use invocation are e-mail and the web.

Awareness applications

Awareness applications are programs that warn users of the WMS that a certain event has happened. Awareness applications are related to monitoring tools, discussed at interface 5. An important difference between awareness application and monitoring tools is that awareness applications are more advanced and are used by the end users. Examples of advanced awareness applications are a comprehensive (real time) graphical user dashboard for the end users (Gartner, 2006), e-mail or pop-up notifications to users when a new work item is available or when a queue is very high. There is a considerable overlap between awareness applications and monitoring tools. The most important difference is that awareness applications are more advanced and present information to the users actively (by e.g. a pop-up).
In order to pass awareness information about workflow instances to a user it has to be sent to outside parties (or programs). These parties (or programs) then send a signal to the user about the event and next the user can take the appropriate action. Examples of awareness information related to interface 2 and 3 are activity status (start or completion of an activity) and activity deadline violations. The WMS can provide e.g. an e-mail notification or a notification in a workflow client application to inform the user about an event.
Another example of 'awareness information' is the system load of the workflow engine. This information is less relevant to the business user and therefore this type of information is send to external management tools, used by e.g. workflow administrators. The application awareness feature is hence also related to interface 5 since programs for workflow administration are positioned in that interface.

Interface 4

Distributed workflow process execution

Distributed workflow process execution means that different workflow engines can be used at the same time for one process. In this way more workflow instances can be supported and this make the WMS suitable for handling processes with more than, say, 10.000 claims a day.

Interface 5

History management

This feature logs e.g. all systems events. Events can be divided into technical related and (workflow) process related. Process related history management keeps track of process related data. This is a so called audit trail and represents the whole route a case has taken. Examples of these logs are activity start time, activity duration and activity completion.
Technical history management is system related, such as user log-on and log-off. Hence, history management can be subdivided into technical history management and process history management.

Technical history management
Technical events are e.g. user log-on and log-off.

Process history management
Record of case related information, e.g. completion time, involved roles etc.

Monitoring

Monitoring in a WMS should be divided into monitoring tools and reporting tools (see interview TIBCO).

Monitoring tools

Monitoring tools report about the operational processes and often do this real-time. TIBCO's product I-process inside is an example of a monitoring tool. It can monitor e.g. the number of cases in a process in real-time. Monitoring tools make it possible to analyze the workflow process from different dimensions. They can show several aspects of the workflow process such as the current activities, the current deadline violations or the priority in activities.

Reporting tools

Reporting tools are somewhat different from monitoring tools. They provide information on a higher level and they do not do this real-time. TIBCO's product I-process analytics is an example of a reporting tool. It provides information for management, e.g the number of cases which have been started in month X. In this situation the number of cases started represents e.g. the number of new customers. Statistics is an important part of the reporting tool feature. Statistics can be used to determine e.g. the average amount of workload, task duration and user performance. The statistics feature of a WMS is an important feature since it can indicate which business processes need to be improved. Statistics can thus initiate a business process redesign. If statistics show that the task duration is longer than agreed in a service agreement than this can trigger an improvement project for task duration.

Resource management

The WMS can push a work item to a certain user. In order to do this the WMS needs to be aware of the available resources and corresponding roles. The feature role management enables the WMS to integrate resources. WMSs sometimes can gather the available resources (participants) from an external organizational directory which is already present in a company. Resource management can be subdivided into resource planning and work balancing.

Resource planning

An example of resource planning is TIBCO's work prediction feature. It predicts the amount of work for the next day for a specific user (group). Managers can get this forecast and anticipate on it by e.g. recruiting extra employees for the next day. This is an example of active planning.

Work balancing

The WMS can push a work item to a specific employee. It is important that all employees have an equal amount of work and that no employee is given too many tasks. The work balancing feature checks the work items (work list) for each employee and decides to which employee the work item is given.

User management

In the user management feature controls all users of the WMS. In this feature users can be assigned different roles which give them certain authorization to parts of the WMS. A manager of a department has for example more rights in the system then a regular employee. User management is a common feature which is present in all WMSs.

Security integration

This feature enables the WMS to use other, already present, authorization systems in the company. Most companies have another authentication and authorization system and some WMSs can use this external system.

Workflow Enactment Service

Basic features

WMSs always have some basic features which are required to let the system work. These basic features are always present in a WMS. Examples are the process management facility (creates workflow instances), the control flow manager (handles state changes) and the work list handler (creates work items). Since the basic features are present in all WMS they will not be addressed in this master thesis project. Only features which can be possible left out a (lightweight) WMS will be discussed.

Exception handling

Current WMSs cannot deal with unexpected exceptions well enough (Gonzalez et al., 2002) and therefore especially these unexpected exceptions will be taken into account in this research. The classification used in this research is the error perspective from Kammer et al. (2000). *Operation errors* are errors caused by mistakes in the execution of the process (e.g. a user gives a wrong input which influences further processing of that case). Flexible WMSs should take these operational errors into account to make sure that they cause no delays. The second type of exception is *design error*. These are errors which are related to the design of the WMS or the implementation of it. These types of errors are very broad in nature and usually a lot of effort is required to solve them. This is because the process model doesn't need to be changed but the system itself. The third type of error is called *dynamic organization error*. Organizations change and this may cause errors in the WMS. Due to an organization change, a process model may not represent the business process correct anymore and this can trigger an exception. If many dynamic organization related exceptions occur for a particular process, then that process needs to be changed. Some WMS have a small feature that triggers an alarm when many exceptions occur in a certain process. Below, several features on exception handling are presented. There is a considerable overlap between the features. Nevertheless they provide an interesting and practical view on exception handling.

Jump forward and backward in the process model

It may be necessary to execute a flow in a different order for a certain customer and for this it must be possible to jump back and forward in the process model. This problem is addressed by some lightweight WMSs but not all. Milano (see Agostini and De Michelis 2000a and 2000b) for example supports this interruption of the regular flow of work. By using formal models in which it is allowed to make so called linear jumps, a user can jump back and forward in the process. The solution they developed is based on Petri Nets in which tokens jump between states. They distinguish strong linear jumps and weak linear jumps. Strong linear jumps are jumps which involve one token in the process model and these jumps require no authorization. Weak jumps require authorization form the process initiator and are somewhat more complex. In weak jumps 2 or more tokens are cancelled and another token is written in the process model.

Skip task and Redo task
This feature is a bit different than the previous feature. Besides jumping forward and backward it is sometimes necessary to ship a task, so not to execute it at all. Likewise is can be necessary to execute a task multiple times, thus redoing a task.

Undoing of incomplete workflow instances
This feature is best illustrated with an example. A process model consists of several steps, e.g. a registration, a check and an action. When the registration and the check are done it might be necessary for some reason to not proceed with that process for a certain case. An easy way to support this cancellation is to allow an undo (or rollback) of this case/task. If this is not supported then a human intervention is needed for the last step, action, since this cannot be executed in the regular way.

Ad-hoc Processes
Ad-hoc processes are processes which are not predefined. This will be explained by the following example. It can be necessary for a certain case to differ from the normal process flow. This differentiation form the normal flow can possibly be solved by the earlier mentioned features on exception handling. On the other hand, on occasion a different approach is needed. In a WMS like Adept (www.adepttechnologies.com), a user can define a different flow for a certain case. When an exception occurs, it is possible to model an alternative path for this case. In this path a user can indicate which activities should be executed next and in which order. This feature in e.g. Adept is allied to the principle of starting an exceptional path or process and is called *support of ad-hoc processes* in this thesis. There are also other situations in which ad-hoc processes are needed. For these, often dynamic, processes with many uncertainties the usage of ad-hoc processes is required because not every path can be specified in advance.

Error detection and failure handling

A running WMS consists of several software processes which are always active. When one of these processes does not function anymore, the WMS can fail. Workflow vendors like TIBCO often have a feature that helps preventing failures. All software processes are monitored constantly and when one process has an error, it is automatically restarted to prevent failure of the WMS. This can be seen as a self-healing mechanism. TIBCO Hawk, an administrative monitor program, also has this feature. It furthermore supports extra failure preventive methods such as CPU monitoring. When the CPU usage gets an unusual high percentage a signal is sent to the system administrator.

Load balancing between workflow engines

A related feature of distributed workflow process execution (interface 4) is load balancing of the workflow engines. Some modern WMS (for example MQ Workflow) support this feature using a three-ball architecture (Zur Muehlen, 2002). When load balancing is used a better usage of the workflow engine is possible and this improves the performance of the WMS.

Data integration

Regular WMSs often support data exchange with other systems in the company. Connection with these systems is a common feature of regular WMSs. It has several disadvantages like performance problems and a longer implementation time. In the figure 8.1, a general architecture of an information system is presented. Several layers can be distinguished, i.e. the application layer, the WMS layer (WFMS), the User interface layer (UIMS), the Database layer (DBMS) and the Operating system layer. The arrows in the figure depict the data integration from the WMS to

the other layers. Lightweight WMS could, like regular WMS, support data integration to the UI layer, the Application layer and the Database integration layer. The question however, is to which extent they should support this.

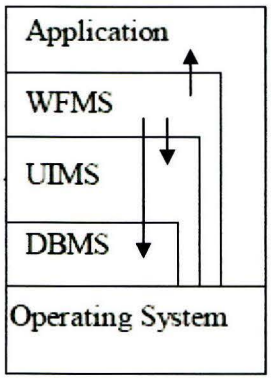

Figure 8.1: Decomposition of generic functionality (Van der Aalst and Van Hee, 2004)

Overview features related to the reference model

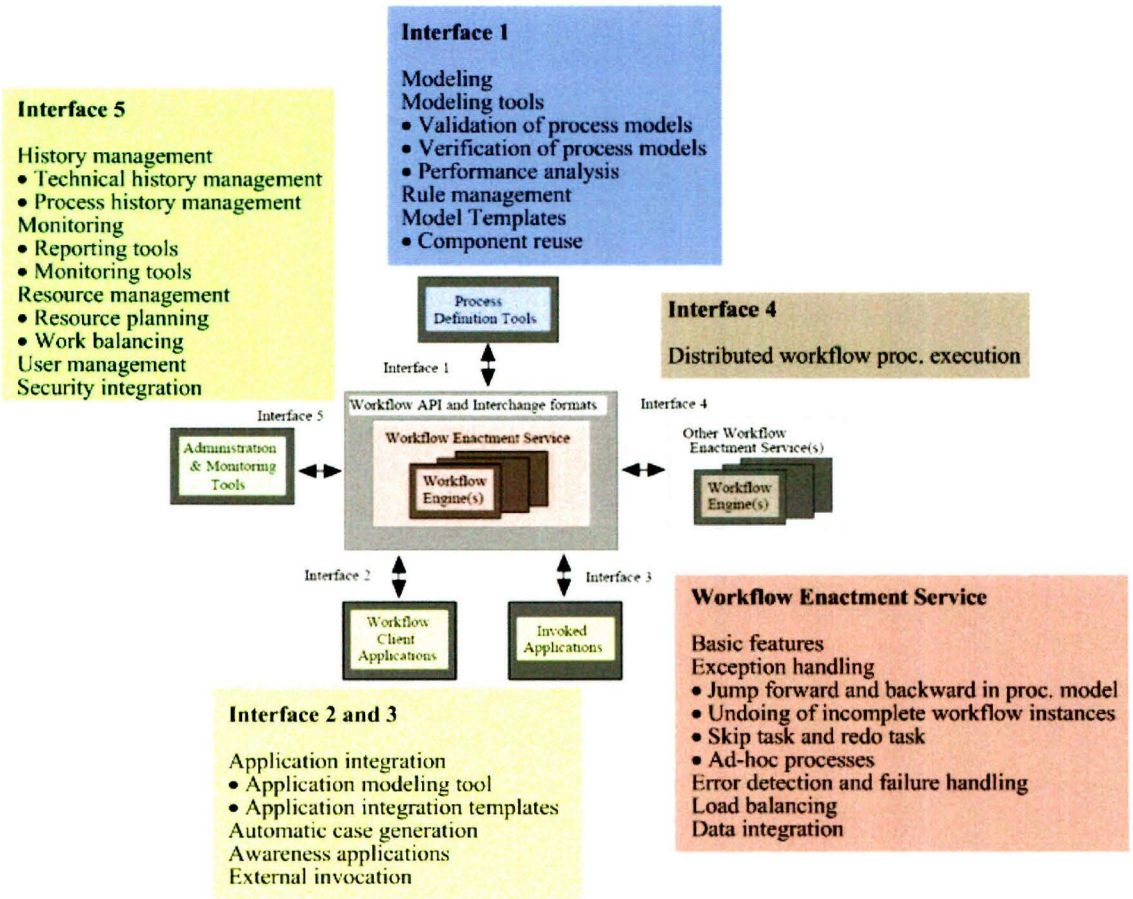

Figure 8.2: Features positioned according to the workflow reference model

Appendix 9 Classification of features

This appendix discusses all features on two issues. (1) Should they be used in a regular (R) WMS, a lightweight WMS (L) or in both (R + L)? (2) Should they be included in lightweight WMS as an essential feature, a desirable feature or and optional feature. Essential features are the core and always needed. Desirable features are needed most of the time. Optional features are needed on certain situations. The appendix builds on appendix 8 where feature of WMSs are discussed.

Interface I

Modeling (R + L) Essential
Modeling is an essential feature for both regular and lightweight WMS. Without modeling the required process models in the WMS cannot be made.

Modeling tools (R + L) Desirable
Modeling tools are often included in both lightweight and regular WMS since making process models is very important for a WMS. Another possibility is the use of an external modeling tool, sometimes provided by another software manufacturer, which can be used for making the process models. Lightweight WMS should contain modeling tools which are easier to use so that business users themselves can also model business processes. Regular WMS can contain modeling tools which are more advanced as consultants can model the business processes for the business users.

Validation of process models (R+L) Desirable
Validation, or basic simulation, is a useful feature but is not essential in a WMS. It is however desirable to have a basic simulation feature, since the quality of the process models can be improved by it. Then again, the most common way to validate is discussing the process models with the involved business users. They know the business processes very well and can indicate whether the process models represents the real life situation as it is.

Verification of process models (R + L) Desirable
Verification of process models is a very useful feature since it performs a couple of checks on a process model. This verification can eliminate errors in a very early stadium and helps to avoid common mistakes. It is a feature for both regular and lightweight WMS.

Performance analysis (R + L) Optional
This feature is also called advanced simulation and helps to determine e.g. future bottlenecks. It also helps in verification of the process models since simulating many cases can pinpoint errors in the model. Performance analysis is very useful in bigger and complex business processes. The feature is an extension on the modeling tool and can be complex to use for regular business users. Therefore it might be less suitable for lightweight WMS. On the other hand, performance analysis is a feature which can be easily used on process models because it can be used as an additional tool. It does not have to be included in the WMS itself and therefore can be used in lightweight situations as well.

Rule management (R)
Process models should support rules in general since the usage of rules is very common. However, a system for rule management which checks on e.g. conflicting rules is too much for a lightweight WMS. In situations where many complex rules need to be made a rule management

feature can help in making these rules. Moreover, when a business process contains many rules that can possibly conflict with each other, a system for maintaining these rules is quite useful. Lightweight workflow management, however, deals in general with less complex processes which do not require advanced rule management. Hence this feature should not be included in a lightweight WMS.

Model templates (R + L) Desirable
A database with model templates is a useful feature for both regular as lightweight WMS. The database can be used separately from the WMS itself and for this reason is does not give much damaging ballast to the WMS system. This feature is not essential in a lightweight WMS but since it is very useful and can speed up the implementation process it is a desirable feature.

Component reuse (R + L) Optional
Like modeling templates, the component reuse can be used for both regular as lightweight WMS. This feature is very useful in situations where a part of a process is used more than once. In lightweight WMS, this feature is optional for situations where processes are used repeatedly.

Interface 2 and 3

Application integration (R + L) Essential
A WMS, both lightweight and regular, is almost always used in a situation where application integration is needed. A WMS is mostly not used stand-alone. The extent to which application integration is supported differs between regular and lightweight WMSs but nevertheless, basic application integration is essential.

Application modeling tool (R + L) Optional
The application modeling tool is useful in situations where e.g. a lot of forms have to be made. The application modeling tool can make the creation of these forms easier so that business users themselves can make the forms. For this reason it should be supported in lightweight workflow as an optional feature, especially in situations where a lot of forms are required.

Application integration templates (R + L) Optional
Like modeling templates, templates for application integration also exist. Common forms can be stored in a database so that they can be used in later processes. The application integration templates are optional in lightweight WMS in situations where many forms are needed.

Automatic case generation (R + L) Desirable
Automatic case generation is used quite often, also in cases of lightweight WMS (e.g. in the case of the computer manufacturer). When no automatic case generation is supported this can be solved by a workaround most of the time. Workarounds should be avoided and for this reason automatic case generation is a desirable feature of lightweight WMS.

External invocation (R)
It is not expected that the external invocation feature is used in situations where a lightweight WMS can be used. External invocation is more likely to be used in complex and bigger environments and for these reasons external invocation is not a feature for lightweight WMS.

Awareness applications (R + L) Desirable
Awareness applications are useful for both regular as lightweight WMS. The feature is useful when important information needs the attention of a user such as a deadline violation of a task. For both regular and lightweight situations, an e-mail notification to a user about e.g. a deadline violation is very useful and hence this feature is classified as a desirable feature.

Interface 4

Distribution of workflow process execution (R)
This feature is quite technical and complex and is used in situations with high volume workflow processes. For this reason it is a feature of regular WMS.

Interface 5

History management (R+L) Essential
This feature is divided in technical history management and process history management.

> Technical history management (R + L) essential
> Technical history management is an essential feature for both regular and lightweight WMS and keeps track of e.g. user log-on and log-off
>
> Process history management (R + L) essential
> Process history management keeps a log on a case on aspects like activity duration and deadline violation. Since process history management is very important for generating management information, it is an essential feature.

Monitoring tools (R + L) Essential
This feature is divided in monitoring tools and reporting tools.

> Monitoring tools (R + L) Essential
> Monitoring tools are necessary to monitor operational processes real-time. They are very useful for both regular as lightweight WMS. This is an essential feature.
>
> Reporting tools (R + L) Essential
> Reporting tools differ from monitoring tools in that they provide information on a higher level such as management information. This information is essential for both lightweight as regular WMS.

Resource management (R + L) Essential
Basic resource management should be supported in both lightweight and regular WMS. Without resource management work items cannot be allocated to specific users. For this reason resource management is an essential feature of WMS.

> Resource planning (R + L) Optional
> Resource planning encompasses options like work prediction so that the amount of work for the next day can be estimated. This feature is useful in larger, high volume, processes and is therefore not required for lightweight workflow. It is however useful in situations where a lightweight WMS is used in a high volume environment. Therefore this feature is not only a regular WMS feature but also is an optional feature for lightweight workflow management.

Work balancing (R)
A work balancing feature for determining the amount of work for each employee is useful in processes where more employees execute similar tasks. Lightweight WMS are usually implemental in less complex situations where no work balancing feature is needed. A lightweight WMS like Protos Activate, for example, has one task list for a certain role. All users with that role can pick tasks out of that task list and thus no work balancing feature is needed in that case. Work balancing is only needed in complex processes where many users with similar roles are present. Work balancing is therefore a feature for regular WMS, not for lightweight WMS.

User management (R + L) Essential
User management is an important feature for both lightweight and regular WMS. Authorization of users is crucial and hence this is an essential feature.

Security integration (R)
Large companies often have sophisticated security systems. The security integration feature allows the WMS to use the already present security system. Since lightweight WM is not suitable for environments with many users and sophisticated security systems are likely to be found in large companies, this feature is not useful for lightweight WMS.

Workflow enactment service

Basic enactment features (R + L) Essential
Basic feature are essential for any WMS and should therefore be supported by both regular and lightweight WMS.

Exception handling (R + L) Desirable
Exception handling is subdivided into four features, listed below. Exception handling is needed quite often in both regular as lightweight WMS. The four features are needed to provide flexibility in executing the flow of work and to deal with exceptions. All features are useful in many situations and thus can be used on a regular basis. Since exception handling is needed in both regular as lightweight WMS, it is important to support these features in every WMS. For this reason all four features are classified as desirable features.

Jump forward and backward in the process model (R + L) Desirable
Skip and redo task (R + L) Desirable
Undoing of workflow instances (R + L) Desirable
Ad-hoc processes (R + L) Desirable

Error detection and failure handling (R + L) Desirable
This feature helps maintaining the WMS by e.g. notifying the IT-department when a certain essential service of the WMS is not running anymore. Since lightweight WMS is used often in small organizations which do not have the capacity to monitor the WMS constantly, this feature can be very useful for lightweight WMS. Yet, also in regular WMS this is a very useful feature. This feature is therefore classified as desirable and can be used in both regular and lightweight WMS.

Load balancing between workflow engines (R)
This feature can be used when multiple workflow engines are present, sometimes seen in high volume workflow processes. Since these situations are not encountered in lightweight WMS this feature should not be included in lightweight WMS.

Data integration (R + L) Desirable
Data integration can be supported to a vary amount. Regular WMS often need more connections to other systems then lightweight WMS and therefore should support data integration to a greater extent. Lightweight WMS, however, also need some support of data integration since they are frequently used in combination with other systems. Examples of these can be found in the cases of the computer manufacturer and the mortgager. Basic data integration is therefore a desirable feature in lightweight workflow management.

Appendix 10 Slightly desirable features

In this appendix three features are described which are slightly desirable. They are classified as desirable in this master thesis project, yet, they can also be seen as optional or essential.

Data integration

A problem rises when categorizing features in essential, desirable and optional, i.e. some features can be considered as *slightly desirable*. For example, the data integration feature is a desirable feature for lightweight WMS according to the above classification. One can imagine, however, that *extensive* support of data integration is not very useful since this is a complex issue in WMS. Support of extensive data integration brings along a lingering implementation phase because a lot of effort is required for all connections to other systems. To make these connections, a lot of customization of the WMS is needed which takes much time. This long implementation time is an inverse of the lightweight approach of a short and easy implementation stage. In these situations it is therefore likely that a regular WMS will perform better. For some features it is therefore important to take the extent to which they are supported into account. The following to paragraphs will give two more examples of features which are not fully desirable. First the modeling tool feature is discussed, next the exception handling feature.

Modeling tool

The modeling tool feature is classified as a desirable feature. When looking at the lightweight WMS Protos Activate, the modeling tool feature is maybe more then desirable, if not essential. The strength of Protos Activate is that business users can model the business processes themselves. With an easy usable modeling tool Protos, which is provided with Protos Activate, process models can be made. Protos is also used as communication tool between consultants and business users. Consultants can design the process models with Protos while business users are watching. Consultants receive direct feedback from the business users who recognize their business processes. Consequently, modeling can be done faster because changes, resulting from remarks from business users, can be made instantly by the consultants. Fewer meetings are necessary between business users and consultants. The interesting aspect about making process models is that a modeling tool like Protos can speed up the implementation phase of a WMS. Since a fast implementation phase is the strength of lightweight WMSs (section 2.1), the modeling tool feature might be more than desirable for a lightweight WMSs, if not essential.

Exception handling

Exception handling is another important aspect in lightweight workflow. Part one of this master thesis states that lightweight workflow should support flexibility and therefore exception handling. When looking at the features for lightweight workflow, exception handling features are thus important. Part I also stated that a mayor problem of current WMS is the lack of flexibility and that WMS should support this. For this reason it might be that the exception handling features are more important than desirable and should thus be classified somewhere between desirable and essential.

Appendix 11 Relevant dimensions in workflow environments

This appendix provides an overview of the relevant dimensions in different workflow environments.

Dimension 1: Process complexity
The process complexity of an environment has impact on the features in multiple ways. (1) In complex environments process models cannot be made by the business users themselves but have to be made by experts (consultants). The easy to use modeling approach of the modeling tool feature becomes therefore irrelevant in complex process environments. (2) Complex processes are likely to have more rules. This causes increased rule complexity and hence a rule management feature as described in appendix 8 might become very useful. One should therefore consider including the rule management feature for complex processes. (3) Complex processes might bring about advanced user management and make the user structure more complex. In other words, in complex processes users often might have multiple roles or a role has multiple users. When several users execute similar tasks, work balancing between the users becomes more important. Increased process complexity might therefore require a work balancing feature.

Dimension 2: Data integration
When the WMS needs to be connected to many other systems the data integration feature is used heavily. Those environments demand more from the data integration feature. The might demand e.g. predefined templates for connection to an Oracle database or to a Microsoft Access database. Environments with much data integration are also likely to support application integration to a bigger extent. Not only connections to other systems might be needed, integration of applications might be necessary as well.

Dimension 3: Volume
In high volume workflow environments many cases need to be processed. Useful features for these situations are performance analysis and resource planning. Performance analysis, or advanced simulation, can help determining how a business process performs in reality. Especially for high volume workflow processes this is a difficult task and advanced simulation is therefore useful. The resource planning feature for high volume workflow processes is very useful because it aids in e.g. work prediction. Resource planning can be used in low volume processes as well. However, it is much more useful in high volume processes.

Dimension 4: Implementation time
When a short implementation time is priority in lightweight WMS, many features become relevant. The modeling tool feature is relevant because it makes modeling of business processes faster. The application modeling tool is important for the same reason, i.e. is makes application integration faster. This is described in appendix 8 that describes the modeling tool feature. The features modeling templates, component reuse and application integration templates are relevant because they also allow process models to be made faster by using previous implementation projects. The last feature which is relevant is data integration. Extensive use of data integration lingers the implementation time and should be avoided. This was described in appendix 8 and 9, which elaborate on the data integration feature. Yet, when basic data integration is supported for some standard or common systems, connections to these systems can be made easier. Thus, no data integration shortens the implementation phase the most. However, when it is really necessary to make connections to other systems, basic support of common or standard systems can make this integration easier and faster.

Dimension 5: Exceptions
Naturally, situations in which with much exceptions occur require extensive exception handling. The features related to exception handling are thus very important in these environments. The jumping forward and backwards in the process model, skip en redo task, undoing of workflow instance and ad-hoc processes features are therefore almost essential in those environments.

Dimension 6: Maintenance
Self-healing features and tools like TIBCO Hawk might be useful in lightweight WMSs considering maintenance. Companies who use a lightweight product often have a small IT staff. When a lightweight WMS has a feature which constantly monitors the WMS, the IT staff can spend less time on monitoring of the system. Since IT staff is relatively expensive for small companies, this feature is very interesting for lightweight systems. For the same reason self-healing mechanisms are very useful in lightweight systems.

Dimension 7: Length of cases
No features are relevant for this dimension. Supporting multiple process models is not included as a feature in this master thesis project because it is not different then supporting process models in general. This belongs to the basic features of a WMS.

Appendix 12 Implementation lightweight WMS at Pallas Athena

The author himself has also carried out an implementation of the lightweight WMS Protos Activate. This was done at the marketing department of Pallas Athena. Two business processes were supported with the lightweight WMS, i.e. the content management of the website of Pallas Athena and the failure management of the website. During 12 weeks, the author modeled the business processes in Protos, converted them to Protos Activate and made a working lightweight WMSs. The process models of the two processes are shown below.

Content management

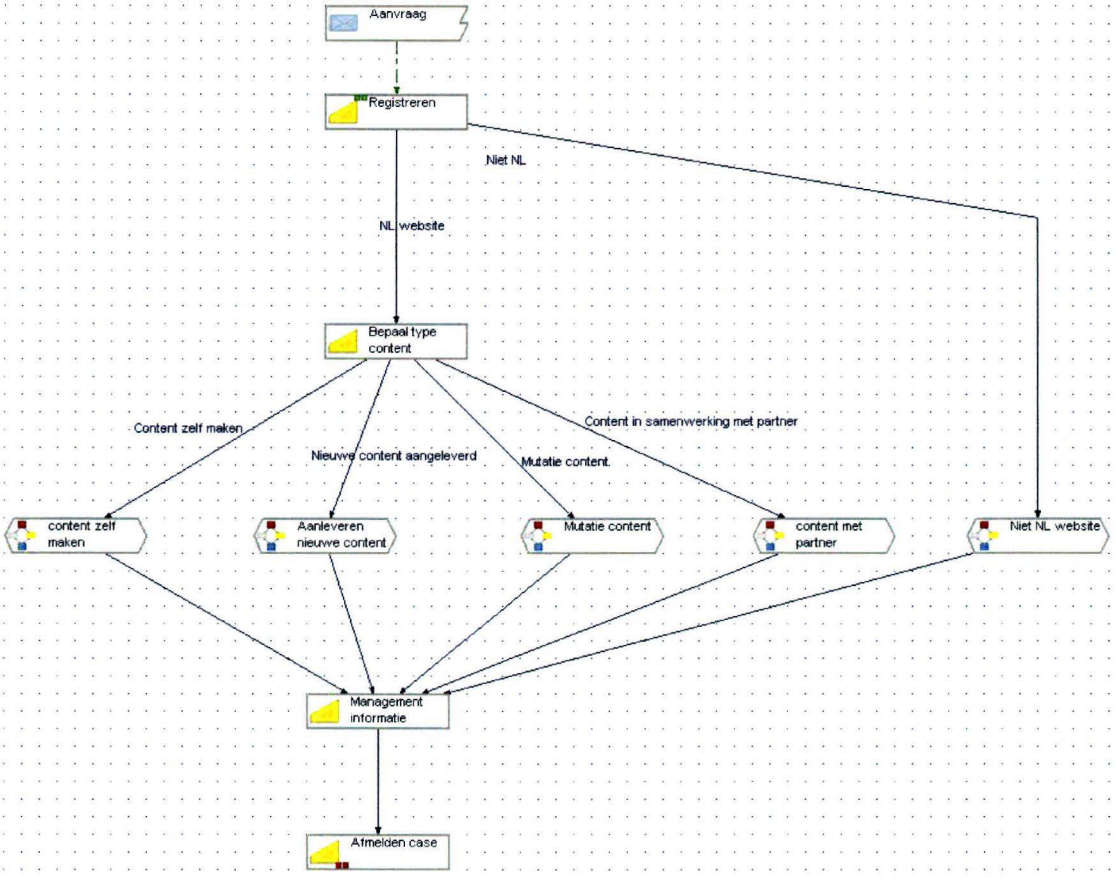

Figure 12.1: process model content management

The figure above shows the business process for content management of the website of Pallas Athena. The request for content is registered first at the activity "registreren", which means register in Dutch. Next, the request follows a certain part, depending on the nature of the content of the request. These processes are not show in the figure above because the figure only shows the main level of the process model and not the detailed levels.

Failure management

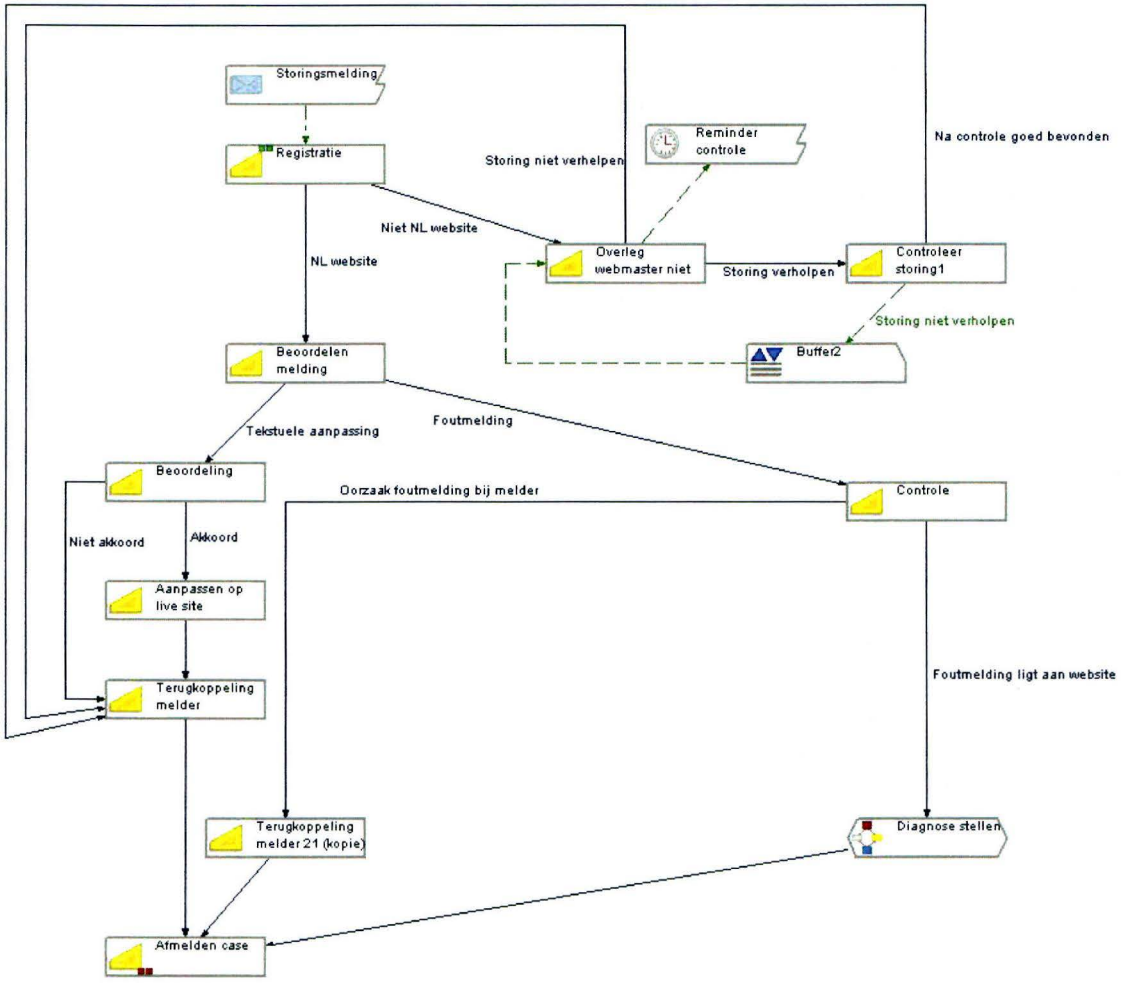

Figure 12.2: process model content management

The figure above shows the business process for failure management of the website of Pallas Athena. The failure notification is registered first at the activity "registreren", which means register in Dutch. Next, failures are divided in *adaptation of text* and *failures*. Text adaptations are corrected instantly on the website of Pallas Athena. Failures are diagnosed and solved. This *diagnose process* not show in the figure above because the figure only shows the main level of the process model and not the detailed levels.

Appendix 13 Results closed evaluation

In this appendix the results of the closed evaluation are presented. Furthermore, this appendix provides general information on the case studies.

Case 1: Computer manufacturer

This case is a description of an implementation of a lightweight WMS at a large computer manufacturer. The lightweight WMS used in this implementation is Protos Activate. The implementation concerns one department of the company, i.e. the software sales department. The scope of this implementation project was the full process of software sales. The focus in this appendix is on problems which were encountered during the implementation and the limitations of the lightweight system Protos Activate.

Problem 1: Complex workflow patterns not supported by the modeling tool of the WMS
Process modeling is easy in Protos Activate because the user friendly modeling tool Protos is used. This tool is easy to learn and is always used for modeling in Protos Activate implementations. The tool Protos is also used during the implementation to communicate about the business processes. The initial modeling is done by consultants of Pallas Athena and after this, business users can model themselves. For this they receive a basic training in Protos. The problem in this computer manufacturer case concerns the way business processes are modeled. The issue can be best described with an example.

Imagine a situation where certain software can be purchased from, say, 10 different software vendors. The first task is to obtain the software request from the customer. The second task is to select the appropriate vendor and the third task the selection of the right software that the vendor offers. The offer is made during the last task, using the previous steps. Since purchasing the software is similar for each vendor, one would model the process like this.

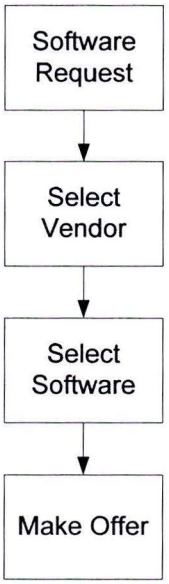

Figure 13.1: Generic way of modeling

Despite the ostensible simplicity in the above model, it is not possible to model this in Protos Activate. It is not possible to make a model like the previous figure and a work-around is needed in this case. The process needs to be modeled like 10 separate processes, depicted in figure 13.2.

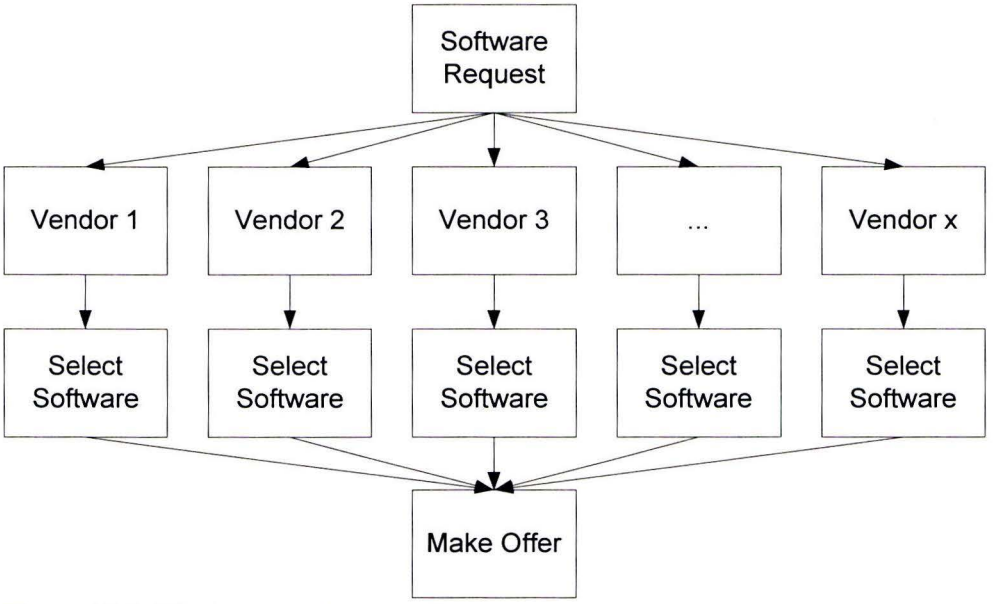

Figure 13.2: Work-around in modeling

The reason for this is that the process models used in Protos Activate need to include all possible paths that a case can follow in advance. The explanation is rather technical in nature and will not be explained in detail. The essence of this problem can be best explained using the *workflow patterns* described by Van der Aalst et al. (2003, see also www.workflowpatterns.com). In the figure above, the *Multi-Merge* pattern is used. The multi-merge is defined as: *"The convergence of two or more branches into a single subsequent branch such that each enablement of an incoming branch results in the thread of control being passed to the subsequent branch"* (Van der Aalst et al., 2003). The problem with the usage of the multi-merge is that all possible paths have to be specified in advance.
When Protos Activate would have supported the pattern *Data interaction – From multiple instance task*, it is not necessary to specify all paths in advance. The Data interaction – From multiple instance task is defined as *"The ability to pass data elements from a task which supports multiple execution instances to a subsequent task"* (Russel et al., 2006). When this pattern is used, it is possible to model the business process in a generic way, as represented in figure 13.1 on the previous page.

The case of the computer manufacturer shows that the impact of not supporting workflow patterns is twofold. (1) Business users often cannot find a work-around and therefore modeling the situation as it is in reality is not possible. This conflicts with the idea that business users should be able to model themselves. (2) The process is less clear an orderly. The advantages of implementing a WMS can not only be deduced to issues like faster processing time and less deadline violations. Another big advantage is that the business process is clear for the business users. Business users have better understanding of the business processes when they are modeled in a clear, understandable way. Hence, when a work-around is needed in modeling, the process models become less evident (compare figure 13.1 and 13.2).

Problem 2: Integration with other systems is limited supported
In the computer manufacturers' case a lot of information was stored in Microsoft Excel format. Software prices, offers and customer information were all stored in this .xls format. A disadvantage of this is that the .xls format is not supported by Protos Activate. In order to make use of .xls, scripting was necessary. Scripting can be seen as programming and thus customizing the WMS to some extent. Scripting in Protos Activate is done using Visual Basic[1] and is thus fairly easy for IT-experts. Business users, however, do not have the knowledge needed for making scripts and thus technical expertise was needed to models the business processes in the lightweight WMS. If the lightweight system Protos Activate had supported Microsoft Excel format, business users could model themselves. For this reasons it might be necessary for lightweight WMSs to support data exchange with common file formats. Examples of this are Adobe Postscript (.pdf) and Microsoft Excel (.xls).

Problem 3: Customize task lists
A WMS helps in the coordination of tasks. The system makes sure the right tasks are sent to the right persons. The tasks are sent to a certain list, called task list. Users of a WMS see these tasks in their list and can execute them. If there are many tasks in a list, it can be very useful to order these tasks by e.g. using a filter, so that a user has a better overview of all tasks. This process is called *customize task list*. In the case of the computer manufacturer this customization was an important issue. Protos Activate supports limited customization of tasks. It supports filtering on the available tasks but it does not support creation of additional task lists (which is a way of ordering a task list).
A possible danger of a long and disordered task list is that business users have no overview anymore on their tasks and this can affect proper handling of cases. Some cases can be signaled too late which can cause a deadline violation. Thus, when a lightweight system is used in an environment where many cases are used, it should support customizing task lists.

Case 2: Mortgager

Problem 1: Support of Ad-hoc processes
The first issue at the mortgager was the modeling of extensions of mortgage validations. A mortgage offer has a certain validation period, i.e. 30 days. Sometimes it is possible to extend these 30 days with another 30 days. It was not possible to model this extension in Protos because processes are executed in a certain order of activities and not in order of time. This will now be explained. When an extension of time is needed, this extension can take place in any part of the process. So it can occur in the beginning of a process but also in the end. In Protos Activate, every possibility needs to be modeled in advance (compare problem 1 in the previous case). This implies that from every activity where an extension is possible, a link needs to be made to the part of the process where an extension is made. This is depicted in figure 13.3.

[1] Visual Basic is a common used programming language made by Microsoft. It is not too complex compared to other programming languages and rather easy to learn.

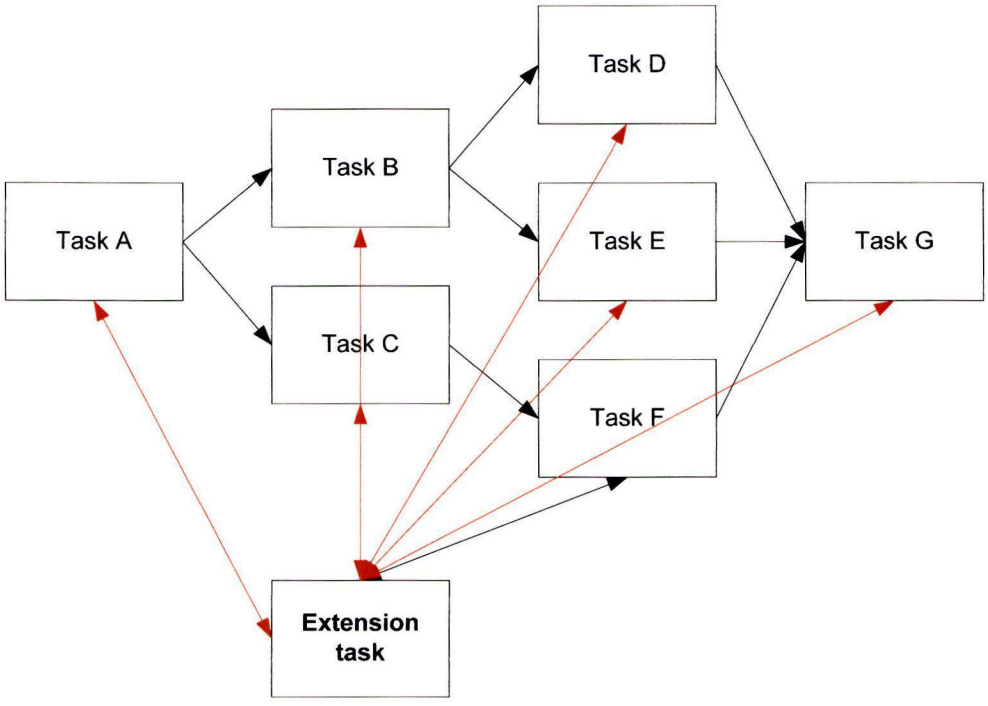

Figure 13.3: Extension task

Theoretically this is possible. However, one can imagine how the process model would look like when it has 50 tasks instead of these 7 tasks. It would be very unclear and consequently the process models are not understandable for the business users anymore. It is very important that the business users understand the process models themselves in lightweight workflow. Another issue is that business users should be able to model business processes themselves. Regular business users cannot model situations like these themselves. They cannot come up with a workaround which e.g. connects every activity where an extension is possible to the part of the process where the extension is made.

A solution for a situation like the mortgager would be to support the use of Ad-hoc processes. A regular WMS like Flower supports this. When ad-hoc processes can be started, not everything has to be modeled in advance. When a mortgage offer needs to be extended, an ad-hoc process is started for this extension after which the process returns to the right state in the original process. The ad-hoc feature is further explained in appendix 8. The problem in the mortgager case can be summarized as *not supporting ad-hoc processes*. When every possibility needs to be modeled in advanced the process model can become very unclear. Consequently it is not understandable anymore for the business users and furthermore the business users cannot make processes like these themselves because they cannot think of a workaround like connecting all activities to a certain part of the process. A lightweight WMS like Protos Activate is therefore not suitable for processes which are dynamic and therefore are not predefined. Processes with many uncertainties usually apply to this description.

Problem 2: Integration with other systems is limited supported
A limitation of Protos Activate, mentioned earlier in the case of the computer manufacturer case, is integration with other systems. In the case of the mortgager this problem also occurred. Protos Activates does not support common integration with systems like Adobe Postscript. In the case of

the mortgager, this was problem since correspondence to clients was done in Postscript format. In order to make use of postscripts, scripting was necessary. Scripting can be seen as programming and thus customization of the lightweight WMS is needed to some extent. Scripting in Protos Activate is done using Visual Basic and is thus fairly easy for IT-experts. Business users, however, do not have the knowledge needed for making scripts and thus technical expertise was needed to models the business processes in the lightweight WMS. If Protos Activate had supported the Adobe Postscript format, business users could model themselves.

Appendix 14 Explanation evaluation framework

In this appendix the evaluation framework is explained. The framework is depicted in figure 14.1 in the end of this appendix. The elements in the evaluation framework are a combination of the problems that lightweight WMSs address, the functionality lightweight WMSs offer and other relevant aspects that come up during the evaluation.

Evaluation on problems addressed by lightweight workflow
Regular WMSs have some problems (Sonnenberg 2006a). Lightweight workflow addresses these problems as described in section 2.1. When evaluating a case of lightweight workflow, it is important to take the problems lightweight workflow addresses into account. Therefore the aspects described in section 2.1.4 are used in the evaluation. The aspects are divided into three categories: implementation, footprint and organizational changes.

Implementation
The implementation part in every case will be subdivided into initial implementation and incremental implementation. Key questions are how the initial implementation would have been different when a regular WMS had been used and how the incremental implementation would have been different when a regular WMS had been used. It is not possible to measure this in a quantitative way. Elements regarding the implementation of a WMS are technical installation, configuration, acceptation by user groups, involvement user groups and duration of implementation. The technical installation incorporates the software for the workflow server, the necessary hardware and other IT infrastructure. The related stakeholder is the IT department. The initial configuration of the WMS affects 2 user groups, i.e. management and business users. It involves the business side of the installation. Acceptation and involvement by the different user groups is also taken into account and the evaluation can differ on these points for the user groups. Early involvement is excepted to have a positive influence for each user group and furthermore a positive involvement is expected to have a positive correlation with acceptation. The fourth facet in implementation is duration of the implementation and this considers the whole implementation project. The last element is effort of the implementation. The consultants are asked to estimate the effort needed in the lightweight WMS. This is again done on a 5 point scale where 1 represents high effort and 5 represents low effort. Effort is also related to the previous element, duration, and contains aspects like duration of the project vs. the number of people needed and the total amount of work needed for the implementation in comparison to similar implementation projects. This can be clarified with an example. When a medium sized implementation project lasted only 4 weeks but required 3 consultants on a full-time basis and 5 people form the customer on a full-time basis, a lot effort was needed. When the implementation would have lasted longer, say 8 weeks, but would have claimed 40 % of the capacity than in the 4 week implementation scenario, the effort needed is less. In general, fast implementations require more effort that less urgent implementations (Interview Heijink, appendix 4).

Footprint
The footprint of a lightweight WMS will be subdivided into human footprint, technical footprint, effectiveness, effort and the ratio effectiveness/effort. The first three aspects are described in the section about footprint of a WMS in chapter 2.1.2. The financial footprint will not be taken into account in this evaluation. The interviewed consultants are often unaware of the price a customer has paid for a WMS and for this reason it is left out of the evaluation. Furthermore, price is a sensitive issue because it can differ a lot among otherwise similar cases.

The first aspect in the evaluation on footprint is the human footprint. This incorporates items like impact on work procedures and is related to the user acceptance element earlier in the evaluation. WMSs are often somewhat rigid and stiff and this can have a negative impact on the work procedures. Users of the WMS sometimes have to adapt their way of working to the WMS while this is not desirable. The impact on work procedures can be negative or positive and can differ among user groups. The second element in footprint is the technical footprint. A WMS requires some components to be installed like a database management system (DBMS) or a fat client. The WMS also makes use of the available hardware like servers and the network infrastructure. The footprint with respect to these technical issues is evaluated in the light of the user group IT department. The third element in evaluating the footprint is effectiveness. For a specific case, two consultants involved in that case are asked to determine the general effectiveness of the lightweight WMS system. When the lightweight WMS system does not perform very well after the implementation it scores low on effectiveness, when the lightweight WMS does perform well after the implementation it scores high on effectiveness. The consultants are asked to estimate the effectiveness of the lightweight WMS in text, so in a qualitative way. After evaluating the lightweight WMS case on footprint issues, the consultants are also asked to make an estimation of all elements when a regular WMS would have been used. Since no actual regular WMS is implemented the consultants have to estimate how a project with a regular WMS would have looked like. For this reason it is important that the consultant has also knowledge about regular WMS cases.

Organizational changes

The aspect of supporting organizational changes is subdivided into representation of business processes, easy and fast change of process models and interruption of regular flow of work. These aspects are described in chapter 1 of this thesis. Representation of business processes considers the way in which the process models describe the actual situation. In other words; do the process models capture the business processes like they are in reality? As described in section 3.3, process modeling is an important aspect in lightweight workflow. Hence the evaluation should contain a part which determines whether the lightweight WMS describes the business processes better than a regular WMS would do this. A consultant involved in the case will be asked to evaluate this and indicate whether and to which extent lightweight workflow is better than a regular WMS in representing the business processes. Since no actual regular WMS is implemented, the consultant has to estimate how a project with a regular WMS would have looked like. The number of exception handling needed by the user groups management and business users is also an aspect in representation of process models as a high number of exceptions indicate a poor representation of the business processes.

Users of WMSs often do not know what they want in the beginning of the implementation of the system and only during the implementation this becomes clear. This fact requires fast and easy change of process models and is evaluated by asking involved consultants about the easiness of changing process models in comparison with a regular WMS. The consultants should indicate whether the lightweight WMS supports easy and fast change of process models better than when a regular WMS would have been used.

The last aspect in organizational changes is interruption of regular flow of work. This considers features like: jumping back and forward in the process model, undoing a task, skipping a task and redoing a task. For the evaluation it is interesting to know whether the lightweight WMS in the case performs better on this than a regular WMS. Two consultants involved in the case with knowledge about both regular and the lightweight WMS are asked to indicate whether the lightweight WMS is the case has performed well on these issues and to which extent it has done better or worse than a regular WMS.

Evaluation on functionality issues
Appendix 8 discussed the features of WMSs. Lightweight systems in general have less features and therefore less functionality than regular WMS. Since this can influence the effectiveness of the WMS it is important to take the functionality issue into account during the evaluation of a WMS. Questions which are asked in this part of the evaluation are: What is different in functionality in comparison to a regular WMS, What are the limitations of the lightweight WMS in the case, What features are related to these limitations, What are potential future problems (limitations) in the case, What features are related to these future problems. The questions can be related to all user groups and the information is gathered by interviewing two consultants involved in the case. The functionality part of the evaluation also contains elements about specific features. Features which are expected to be highly relevant in the light of lightweight workflow are added to the evaluation. These features are expected to play an important role in lightweight workflow and are therefore interesting in the evaluation is a lightweight WMS.

Feature related aspects
In appendix 8 the features of WMSs were discussed. Some of these features are very interesting for lightweight workflow, as described in appendix 9. These important features for lightweight workflow, modeling tool, data integration and error detection and failure handling are also used in the evaluation. A consultant is asked to evaluate the cases with respect to these three features. Important in this evaluation is the difference with respect to these features when a regular WMS would have been used. It is not possible to predetermine criteria for this part of the evaluation because it is too case specific.

Other aspects
So far, two parts of the evaluation are discussed, i.e. problems addressed by lightweight workflow and functionality aspects. The last part contains other elements which would have been different in case a regular WMS would have been used. Two consultants are asked to think about other aspects in the evaluation of the case and these will also be taken into account. The aspects are possibly specific for each case and therefore it might be difficult to compare the cases with each other on these elements.

Evaluation criteria lightweight WMS	User group	Evaluation on crit.	Comparison regular WMS
Problems addressed by lightweight WMS			
Initial implementation			
Technical installation	IT	Text	1 to 5[1]
Configuration of the system	BU, MA	Text	1 to 5[1]
Acceptation by user groups	BU, MA, IT	Text	1 to 5[1]
Involvement user groups	BU, MA, IT	Text	1 to 5[1]
Effort	BU, MA, IT	Text	1 to 5[1]
Duration initial implementation	BU, MA, IT	Text	1 to 5[1]
Incremental implementation			
(Re)Configuration of the system	BU, MA	Text	1 to 5[1]
Acceptation by user groups	BU, MA	Text	1 to 5[1]
Involvement user groups	BU, MA	Text	1 to 5[1]
Effort	BU, MA, IT	Text	1 to 5[1]
Duration incremental implementation	BU, MA	Text	1 to 5[1]
Footprint			
Human footprint	BU, MA	Text	1 to 5[1]
Technical footprint	IT	Text	1 to 5[1]
Effectiveness	BU, MA, IT	Text	1 to 5[1]
Organizational changes			
Representation business processes	BU, MA	Text	1 to 5[1]
Easy and fast change of process models	BU, MA	Text	1 to 5[1]
Interruption regular flow of work	BU, MA	Text	1 to 5[1]
Functionality			
What differs in functionality in comparison to a regular WMS?	BU, MA, IT	Text	n.a.
What are the limitations of the lightweight WMS?	BU, MA, IT	Text	n.a.
What features are related to these limitations?	BU, MA, IT	Text	n.a.
What are potential future problems (limitations)?	BU, MA, IT	Text	n.a.
What features are related to these future problems?	BU, MA, IT	Text	n.a.
Lightweight Specific features:			
Modeling tool	BU, MA	Text	1 to 5[1]
Data integration	BU, MA, IT	Text	1 to 5[1]
Error detection and failure handling	BU, MA, IT	Text	1 to 5[1]
Other aspects			
Other relevant aspects in case	BU, MA, IT	Text	n.a.

Legend

[1] 1 is poorer than regular WMS, 5 is better than regular WMS

n.a. Not applicable

BU User group: Business users

MA User group: Management

IT User group: IT department

Figure 14.1: Evaluation framework

Appendix 15 Results open evaluation case studies

In this appendix the results of the open evaluation are presented. Consultants involved in the implementation of Protos Activate were asked to evaluate the case of the computer manufacturer and the mortgager. First, an interview with one consultant was conducted on the case of the computer manufacturer. Next, a follow-up interview with the same consultant was held on the case of the mortgager. Last, one interview was carried out with a second consultant to validate the results gathered in the first two interviews. In this last interview, both cases were examined.

In this appendix, the overall results on case of the computer manufacturer are described first. The results on the mortgager case are explained second. Since the results of both cases were similar, the part on the mortgager is described briefly. Only the differences in comparison to the case of the computer manufacturer are mentioned.

Case 1: Computer manufacturer

The results are grouped according to the different parts of the evaluation. First, the results regarding the initial implementation are given and after this the results regarding the incremental implementation are shown. The third part discusses the results about the part of the footprint and the fourth part concerns the organizational changes. The last part discusses the functionality.

Initial implementation

The technical implementation is easier with Protos Activate than with Flower. Consequently, the consultants gave 4 points on this part of the evaluation. A related aspect is configuration of the system and in the case of the computer manufacturer there were 2 issues which caused a score of 2. The first issue is related to the used file formats. Protos Activate does not support Microsoft Excel by default and because these files were needed for external communication some customization was needed. The second issue wanted to customize their task list and this is also not supported in Protos Activate.

The acceptation by the business users was good in the case of the computer manufacturer. This is related to the next point in the evaluation, involvement by the users. More involvement leads in general to more acceptation of the system. It is easy with Protos Activate to let users participate during the implementation of the system. This leads to better involvement and therefore more commitment. It is harder to involve users when Flower is used. The consultants scored acceptation and evaluation with a 4.

The effort is expressed by the number of labor needed at the implementation. When a regular WMS is used, the implementation process is more sequential is nature. This means that the users are interviewed first and the functional design (i.e. first version of the process models) is made next. The consultants then have to go back to the company and activate and configure these first version models. These models are tested next, after which they are improved. This sequential process is time consuming and needs a lot of effort. When Protos Activate is used, the implementation is more iterative in nature, i.e. users are involved more often. Process models are still being made by first interviewing the users but are improved continuously by interviewing business users. The advantage of a tool like Protos is that improving the models is easier and faster then with a regular modeling tool and that modeling can be done while users are present. When users are present they can comment immediately on the new models and changes can be made on the spot. Furthermore, consultants do not have to configure and implement the new models; the business users can implement the models themselves. Moreover, configuration of the WMS is not needed, only activating the models and this takes only several minutes.

Evaluation criteria	Results interview with first consultant on case computer manufacturer	Results interview first consultant on case mortgager	Results interview with second consultant computer manufacturer	Results validation interview with second consultant on case mortgager	Results validation interview with second consultant on case
Problems addressed by lightweight WMS					
Initial implementation					
Technical installation	4	4	4	4	
Configuration of the system	2	2	2	2	
Acceptation by user groups	4	5	4	5	
Involvement user groups	4	5	4	5	
Effort	5	5	5	5	
Duration initial implementation	5	5	5	5	
Average score	**4**	**4,3**	**4**	**4,3**	
Incremental implementation					
(Re)Configuration of the system	4	4	4	4	
Acceptation by user groups	4	5	4	5	
Involvement user groups	4	5	4	5	
Effort	5	5	5	5	
Duration incremental implementation	5	5	5	5	
Average score	**4,4**	**4,8**	**4,4**	**4,8**	
Footprint					
Human footprint	3	3	3	3	
Technical footprint	2	2	2	2	
Effectiveness	2	2	2	2	
Average score	**2,3**	**2,3**	**2,3**	**2,3**	
Organizational changes					
Representation business processes	4	4	4	4	
Easy and fast change of process models	5	5	5	5	
Interruption regular flow of work	1	1	1	1	
Average score	**3,3**	**3,3**	**3,3**	**3,3**	
Functionality					
Lightweight Specific features:					
Modeling tool	5	5	5	5	
Data integration	1	1	1	1	
Error detection and failure handling	3	3	3	3	
Average score	**3**	**3**	**3**	**3**	
Total average score	**3,4**	**3,6**	**3,4**	**3,6**	

Legend
1 is poorer than regular WMS, 5 is better than regular WMS

Table 15.1: Results case study evaluation

Thus, when Flower is used the implementation process is more abstract; business users cannot see the models directly, they have to be made later. When Protos Activate is used the models can be made when business users are present and this saves effort and time. For these reasons the consultants gave 5 points on effort and duration.

Incremental implementation

For most reconfigurations, the lightweight WMS Protos Activate works the same as the regular WMS Flower. For updates to the WMS, a new WMS has to be installed which takes a little more time when Flower is used. Smaller reconfigurations like user management are easier and made faster in Protos Activate. For these reasons the consultants scored Protos Activate with a 4 with respect to reconfiguration of the system.

The next aspects are involvement and acceptation. When Protos Activate is bought by a company, a decision is made about maintenance of the system. A customer can choose whether Pallas Athena can make new or additional process models or that the business users do this themselves. Naturally, in the last situation, there is more involvement and acceptation for incremental implementations. Thus, because business users can do some incremental implementations themselves, Protos Activate performs better then Flower on involvement and acceptation. For this reason the consultants gave a score of 4 points.

The effort in incremental implementations is less with Protos Activate then with Flower. When Activate is used, a customer can load the new business processes into the WMS. When flower is used, new process models have to be loaded into the WMS by the consultant and furthermore, some configuration is needed. Issues like this illustrate that with Protos Activate less effort is needed for incremental implementations. The same applies for duration of implementations. When a consultant needs to load the new process models into the system and need to do some reconfiguration, this takes a more time. Hence duration is given 5 points.

Footprint

The consultants estimate the human footprint similar for Protos Activate and Flower. Both WMS do not have a large human footprint in general. Hence the score for human footprint is 3 points. The technical footprint differs between Activate and Flower. In the case of the computer manufacturer, offers to clients were sent in Microsoft Excel format. This file format is not supported by Activate but is supported in Flower. As a result, the Excel files are embedded in the Flower WMS, but not in the Activate WMS. In issues like file backup, the IT department has to take 2 systems into account, i.e. The Protos Activate environment and the Windows environment (where the Excel files to clients are located).

When looked at effectiveness in the case of the computer manufacturer, the lightweight WMS scored lower then the regular WMS. As described in section 4.1, many workarounds were needed to model the business processes. These workarounds lead to a lower score on effectiveness. When Flower would have been used, no workarounds would have been needed. Hence the consultants gave effectiveness a score of 2.

Organizational changes

Modeling in Protos Activate is done with the modeling tool Protos. This tool is easy in usage and understandable for business users. In addition to these advantages, Protos also represents the business processes well. Modeling power in Flower is similar to Protos, however, communication to the business users is less clear with Flower. For this reason the consultants gave a score of 4 in the evaluation. When Protos is used for modeling, business users can adept the process models themselves if they find errors. In Flower, the business users cannot models themselves and therefore a score of 5 is given in the evaluation.

Interruption of the regular flow of work is very easy with Flower. In Protos Activate it is not possible to dissent form the predefined path. Activate scores low in exception handling in general

while Flower handles exceptions very well. For this reason a score of 1 is given on interruption of regular flow of work.

Functionality
When comparing Protos Activate to Flower on features, Protos Activate has less functionality in exception handling (see section 4.4). With Protos Activate it is not possible to e.g. jump forward and backward in the process model. Skipping and redoing tasks or undoing of a workflow instance is not possible either. Last, ad-hoc processes are also not supported by Protos Activate while this is something which is often needed in dynamic or flexibly settings (see section 3.4). In the mortgager case, the use of ad-hoc processes was needed and a workaround was applied to overcome the missing of ad-hoc processes. Flower does support all these features and this makes it more suitable than Protos Activate in an environment where a lot of flexibility is needed. One can say that a limitation of Protos Activate can be found in the light of flexibility.
The consultants indicated in the evaluation that BPM suites are a future trend in workflow management. Pallas Athena is currently busy with their BPM suite and it will contain their programs Protos, Protos Activate and Flower. The integration between the lightweight WMS Protos Activate and the regular WMS Flower will be increased with the BPM suite. Issues for the future can therefore be found in this domain. Questions like 'how can a company upgrade from a lightweight WMS to a regular WMS' become very relevant. Regarding lightweight workflow, this is a very interesting topic. This trend can increase sales in lightweight systems. Organizations might purchase a WMS easier if they know that they can extend it later to a regular WMS. The lightweight WMS can become the predecessor for the regular WMS. It might even take away barriers for purchasing a regular WMS. Companies can try a WMS in an inexpensive and easy way by using a lightweight WMS and when they desire, they can upgrade later to the regular WMS.
Another issue which might become relevant in the near future is switching form workflow vendor. Currently switching form WMS vendor is not easy and often takes much time since the implementation has to be done again form scratch. When WMSs are standardized, switching becomes easier. For this reason, standardization of WMSs is also a future topic for lightweight workflow. Standardization can also put in the light of the upgrade perspective. When a lightweight WMS complies with the future workflow standard, it might be possible to upgrade from that *lightweight* WMS to any other *regular* WMS. The benefit for lightweight workflow vendors is that they can market their product as a predecessor for any other WMS. Thus, when a company buys their lightweight WMS, it can try out workflow management in general relatively cheap and easy. When the company outgrows the lightweight WMS, it is then free in the choice for any other regular WMS. Applying to a workflow standard is therefore a worthy consideration and can provide large advantages. The workflow management coalition plays (www.WfMC.org) a very important role in this standardization.

The author expects that some of the features described in appendix 8 play a very important role in lightweight workflow. These features are modeling tool, data integration and error detection and failure handling. The modeling tool feature is very important in Protos Activate. Modeling is done in a separate tool called Protos. The models in Protos are very easy to read and to draw and hence the tool can be used by the business users. This is an important difference compared to the regular WMS Flower as business users cannot models themselves in Flower. Modeling is more abstract in Flower because business users cannot see the models immediately; they have to be designed by consultants first. With Protos Activate however, models can be made in the presence of the business users. Protos is thus also used as a communication tool and users see the processes which improves their understanding of the models. The modeling tool is an important feature in lightweight WMS.

The data integration feature in WMS is important because it often involves customizing the WMS. Integrating with common IT-systems is often supported by default though this data integration often takes a considerable amount of time and causes more complexity. Integration with less common IT-systems is even harder and in these situations customization of the WMS is needed. The extra implementation time and increased complexity conflict with important advantages of lightweight workflow of a fast implementation phase. The strengths of lightweight WMS are fast implementation time and easy to model workflow processes and this requires standardization. For these reasons one can argue that lightweight WMS should not support too much data integration, despite entailed the limitations.
The third feature evaluated is *error detection and failure handling*. Flower and Protos Activate have similar functionalities in this. The consultants state, however, that error detection in modeling should be investigated when business users model the business processes themselves. Because business users are no modeling experts, their way of modeling might be inefficient or even wrong. When a modeling tool provides a mechanism which can aid business users in modeling and helps preventing modeling mistakes, better models can be obtained. Currently, some of those checks are made on the models when using Protos of Flower and hence this feature is supported already to some extend. Nevertheless it can be useful to extent this feature further so that process models become better in the future.

The overall score in comparing Protos Activate with Flower is 3,6. This score is a quick indication that Protos Activate performs somewhat better on the evaluated aspects in comparison to Flower. Protos Activate scored very poor on interruption of regular flow of work (e.g. exception handling) and data integration. The consultants gave a score of 1 on these points in the evaluation. Protos Activate scored very well on effort and duration of the implementation as well as on easy and fast change of process models. The consultants gave a score of 5 on these points.

Case 2: Mortgager
The result of the mortgager case is similar to the case of the computer manufacturer. Nevertheless, there were some differences, i.e. the involvement and acceptation of user groups was higher in the case of the mortgager.

The involvement of the business users was higher in the mortgager case then in the computer manufacturer case. The maintenance in the case of the computer manufacturer was done by Pallas Athena, in the mortgager case by the company itself. Maintenance can be technical in nature but changes in process models can also considered as maintenance. In order to do this maintenance, the company has to be involved more during the implementation to get more knowledge about Protos Activate. Therefore involvement of all user groups was higher.

Another difference between the cases is that the mortgager does not use the lightweight WMS yet. Therefore, the acceptation can only be estimated, not measured. Nevertheless the consultants expect high acceptation by all user groups because they have been involved a lot and moreover, they have tested the prototype of the lightweight WMS. Since user groups were very content with the prototype, a high acceptance of the WMS is expected. Because the WMS is not in use yet, some other elements are estimated as well. Despite this, the case of the mortgager is still valuable since the major part of the implementation is already done.

Appendix 16 How light is Protos Activate?

This appendix compares Protos Activate with the most important characteristics and features of lightweight workflow. Protos Activate is first compared with relevant characteristics presented in chapter 2. Second, Protos Activate is compared with the features of lightweight WMSs presented in chapter 3.

Characteristics

Lightweight WMSs have several characteristics. These are shown in the table on the next page. The main characteristic, stated in chapter 1, is less functionality then regular WMSs. When comparing Protos Activate to its counterpart Flower, Protos Activate has less functionality. Since lightweight WMSs have less features then regular WMSs, Protos Activate is light with respect to functionality.

The second characteristic of lightweight WMS is a fast implementation. Regular WMSs often have implementations which last 3 to 4 months. Lightweight systems should have an implementation phase which is much shorter, e.g. 1 month. Protos Activate has an implementation phase of approximately 1,5 months (interview Heijink, appendix 4) which makes it lightweight regarding the implementation.

Regular WMSs have a footprint with respect to financial, technical and human matters. The evaluation of Protos Activate in chapter 3 demonstrated that the technical footprint was bigger for Protos Activate then for Flower. Regarding the human footprint, Protos Activate and Flower were considered to be the same. It seems that Protos Activate has a smaller financial footprint then Flower because it supports fewer features and is priced lower. Therefore, it is likely to conclude that Protos Activate is nor regular nor lightweight on footprint.

A fourth characteristic of lightweight workflow is support of flexibility with respect to work procedures. Chapter 1 states that lightweight WMSs should not be rigid and stiff like many regular WMSs are, lightweight WMSs should be flexible for business users. This can be achieved by e.g. the exception handling features presented in appendix 8. Protos Activate does not support these features and therefore scores low on flexibility. Regarding flexibility it can be concluded that Protos Activate is not lightweight.

The fifth characteristic concerns the modeling of business processes. Chapter 1 states that business users should be able to model and adapt the business processes themselves. This is necessary because organizational changes cause changes in the process models. Protos Activate uses the program *Protos* for modeling. This modeling tool is easy in usage and understandable for the business users. Protos Activate, thus, complies with characteristic that business users should be able to model and change business process themselves. Hence Protos Activate is lightweight in this aspect.

The last characteristic is the bottom-up approach instead of the top-down approach. It is not clear how implementations of Protos Activate are initiated and therefore this characteristic is not taken into account.

Number	Characteristic	Supported in Protos Activate
1	Less functionality than regular WMSs	Yes
2	Faster implementation phases than regular WMSs	Yes
3	Smaller footprint than regular WMSs	Neutral
4	Support of flexibility with respect to work procedures	No
5	Easy and fast support of organizational changes	Yes
6	Bottom-up implementation instead of the regular WMSs' top-down approach	No information

Table 16.1: How light is Protos Activate?

Features

In section 3.2 an overview of lightweight features was presented. These features were classified in essential, optional and desirable. The table below shows which of these features are supported by Protos Activate.

WF Reference model Interface[1]	Features	Classification	Supported in Protos Activate
1	Modeling	Desirable	Yes
	Modeling tools	Desirable	Yes
	Validation of process models	Desirable	Yes
	Verification of process models	Desirable	Yes
	Performance analysis	Optional	No
	Modeling templates	Desirable	Yes
	Component reuse	Optional	No
2 & 3	Application integration	Essential	Yes
	Application modeling tool	Optional	No
	Application integration templates	Optional	No
	Automatic case generation	Desirable	Yes
	Awareness application	Desirable	No
5	Technical history management	Essential	Yes
	Process history management	Essential	Yes
	Monitoring tools	Essential	Yes
	Reporting tools	Essential	Yes
	Resource management	Essential	Yes
	Resource planning	Optional	No
	User management	Essential	Yes
Workflow enactment service	Basic features	Essential	Yes
	Exception handling	Desirable	No
	Jump forward and backward in process model	Desirable	No
	Skip and redo task	Desirable	No
	Undoing workflow instances	Desirable	No
	Ad-hoc processes	Desirable	No
	Error detection and failure handling	Desirable	Yes
	Data integration	Desirable	Yes

Legend

[1] Position according to the Workflow Reference Model (Hollingsworth, 1995)

Table 16.2: Features supported by Protos Activate

All essential features are supported by Protos Activate. When looking at the desirable features, all features are supported except for the awareness application feature and the exception handling features. Especially the last group of features, exception handling, are very important for lightweight WMS. When a WMS does not support these features, it is likely that the WMS does not support enough flexibility. Regarding flexibility, Protos Activate is thus not lightweight.
Table xx also indicates that optional features are not supported by Protos Activate. Since optional features do not have to be included in a lightweight WMS in any case, Protos Activate is not less lightweight because it does not have these optional features. One can even argue that Protos Activate is particularly lightweight because it only supports the fundamental and desired features and no extra unnecessary features.

Conclusion

The overall conclusion is that Protos Activate is a lightweight WMS on most issues. It supports the necessary features and has the important characteristics of a lightweight WMS.

However, Protos Activate is not lightweight on the aspect of flexibility. It does not support e.g. exception handling features and consequently, is not lightweight with respect to flexibility. In addition, the characteristics of lightweight WMSs on the previous page indicated that Protos Activate is not lightweight with respect to flexibility in work procedures. Protos Activate therefore seems as rigid as regular WMSs.

Thus, most features and characteristics indicate that Protos Activate is a lightweight WMS. Only with respect to flexibility Protos Activate is not classified as lightweight.