# Workflow management systems

Master's thesis

Geir Inge S. Imsland
Department of Informatics
University of Bergen

1 December 2007

# Preface

This master's thesis gives an insight to workflow technologies used to improve efficiency of business processes. Ways to use such technologies in order to help users through tasks in MIPS (Material Integrated Production System) are discussed. Description on how to implement a *workflow management system* for MIPS is provided. *Workflow engine, workflow importer* and *user interface* are the three most important parts discussed.

## Acknowledgements

First of all I would like to thank Carsten Helgesen for being my counsellor. He has been highly available and very supportive throughout the whole process of writing this thesis. I would also like to thank Ole Dramdal and Einar Dag Digernes from Aker Kvaerner for giving me the opportunity to work with workflow technology and MIPS. I also thank my fiancée Kristine for her patience, understanding and encouragement. I would also like to thank my parents Theodor and Olaug for supporting me through the years.

## Table of contents

# 1. Introduction

Aker Kvaerner has many different computer systems that users work with. This thesis focuses on those systems being used in project management, primarily MIPS (Material Integrated Production System).

MIPS is used to control material, production and completion in large building projects, and has been developed since 1976. It includes tools for material warehousing, like purchase orders, requisitions, tracking shipment, budgeting and tools management. It also has tools for project management, like completion of tasks, commissioning, preservation, technical information, cost control and document control.

Figure 1 shows other project management systems that MIPS is closely related to.

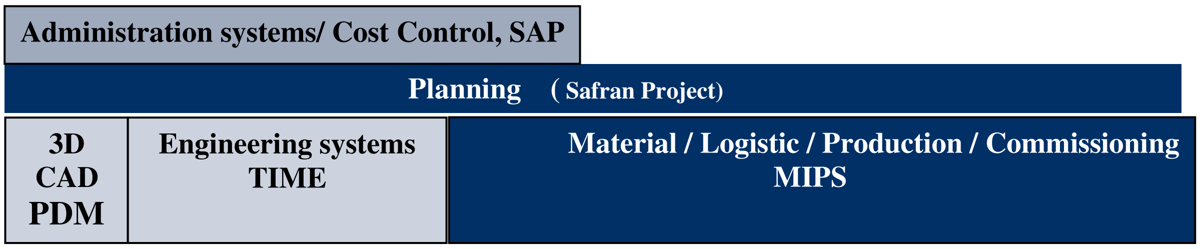

Administration systems/ Cost Control, SAP
Planning ( Safran Project)
3D CAD PDM
Engineering systems TIME
Material / Logistic / Production / Commissioning MIPS

**Figure 1: MIPS amongst other project management systems.**

MIPS has over 4000 users and is used in more than 400 projects all over the world. It has about 1000 different views, and its database has more than 500 tables.

In figure 2 below the MIPS main menu can be seen. The main menu starts one module which has its own views.

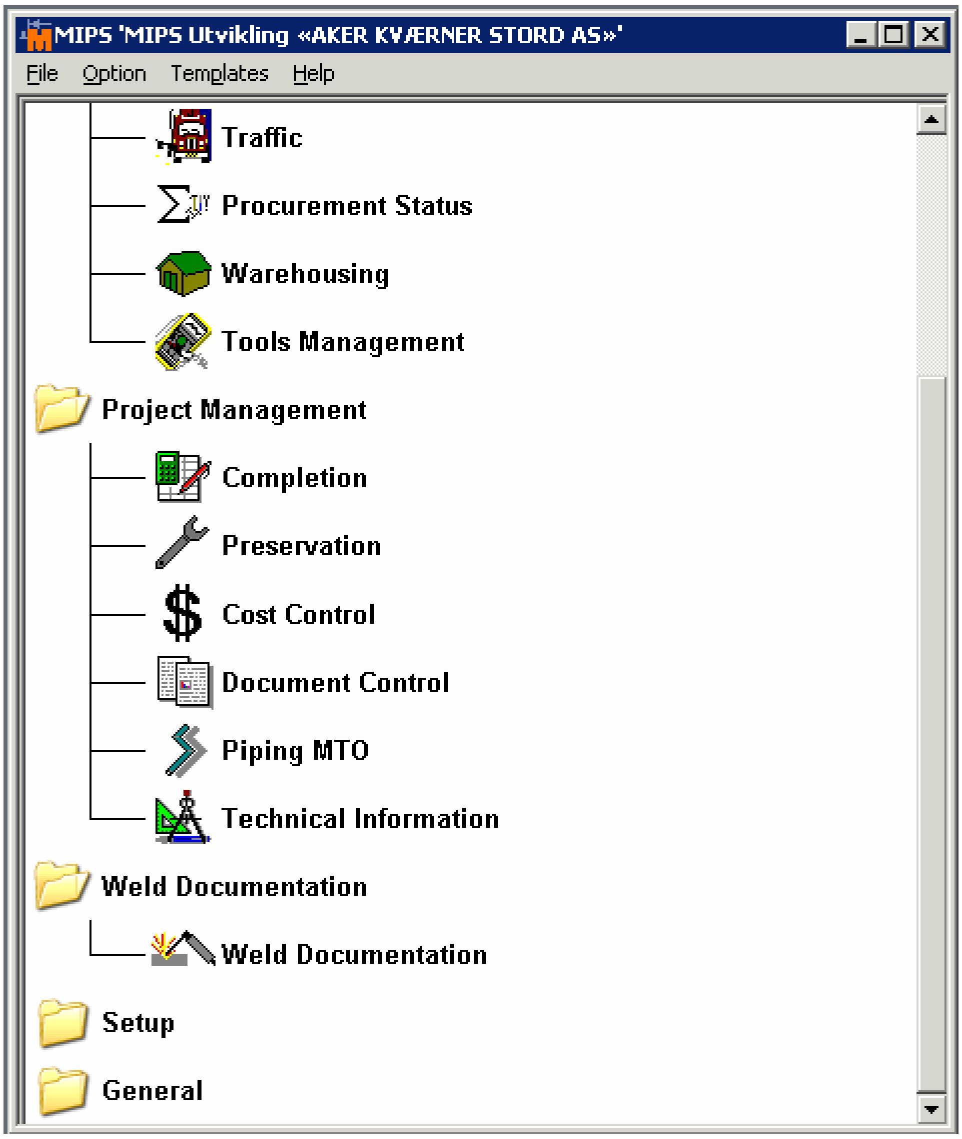


Figure 2: MIPS main menu.

The main user interface shows up after clicking on a menu item. The two rows of buttons on top of the user interface represents all functions associated with a view. These rows can be seen in figure 3. Notice that there are a lot of functions involved.

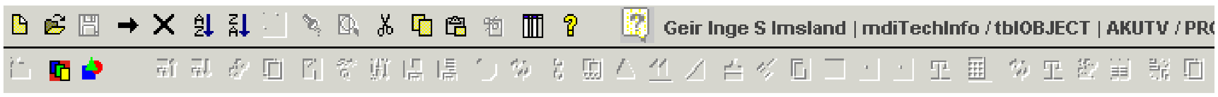


**Figure 3: Function buttons for a view in MIPS.**

With all of MIPS' views and all of the functions involved it is easy to see that it can be hard for new users working in MIPS. This thesis covers solutions making it easier to work with.

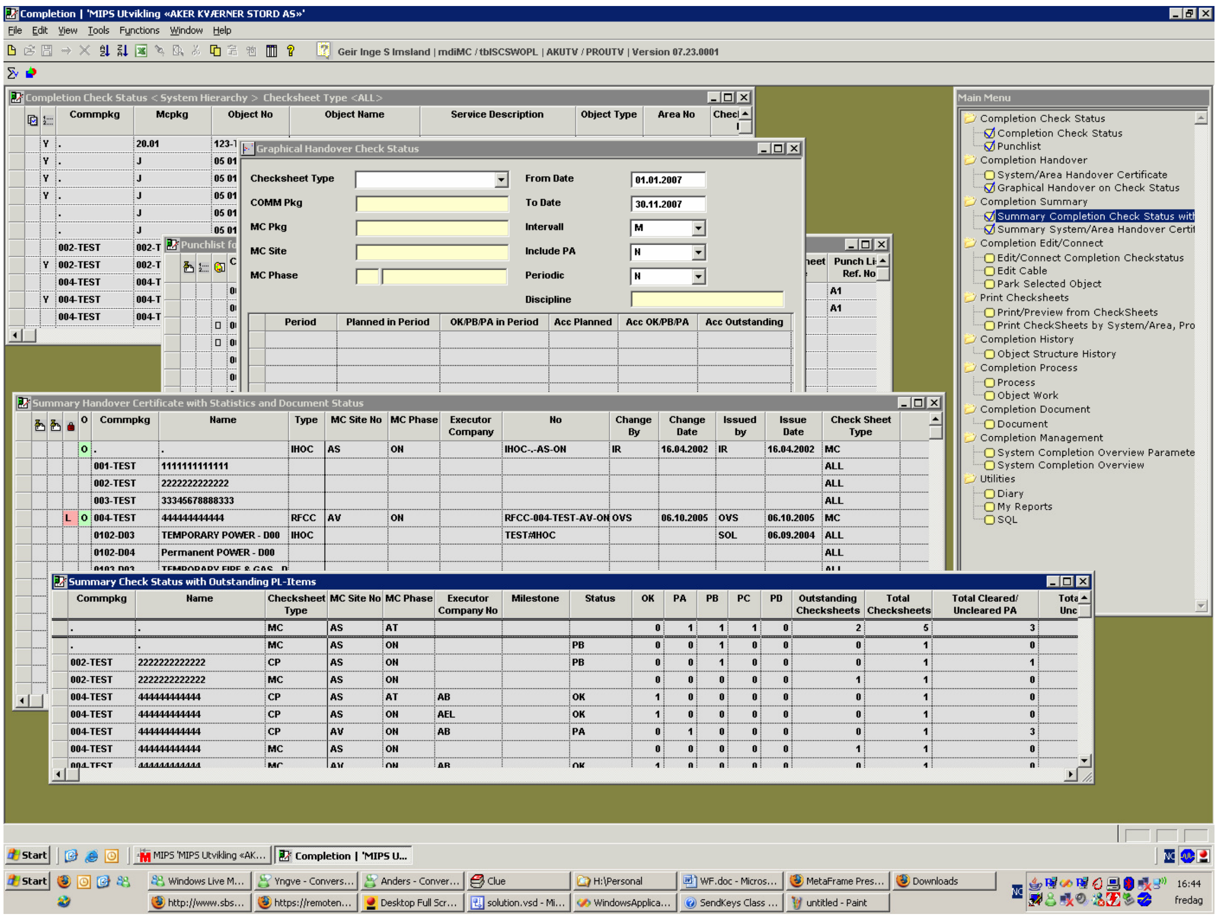
## *1.1. Why this system is being made*

Users planning projects in Aker Kvaerner have many different systems to work with. This can at times be very messy and difficult, especially for new users. Many views are open in MIPS when users do fairly simple tasks. For new users it will be a lot easier if they are somehow guided through all subtasks in project systems.

A workflow system will guide users from subtask to subtask in order to complete the whole task. New users will then learn one correct way of doing tasks. There can of course be several ways to complete tasks, but the workflow system will show users one of them. This system will also help experienced users. With this system they can work faster and more goal oriented.

## 1.2. Today's situation

Today there are many complex tasks in many different data systems. MIPS will eventually end up with many open views which in turn make tasks even more complex. An example can be seen in figure 4 below.


Completion | 'MIPS Utvikling «AKER KVÆRNER STORD AS»'

Geir Inge S Imsland | mdiMC / tblSCSWOPL | AKUTV / PROUTV | Version 07.23.0001

Completion Check Status < System Hierarchy > Checksheet Type <ALL>

| | Commpkg | Mcpkg | Object No |
|---|---|---|---|
| Y | . | 20.01 | 123-1 |
| Y | . | J | 05 01 |
| Y | . | J | 05 01 |
| Y | . | J | 05 01 |
| | . | J | 05 01 |
| | . | J | 05 01 |
| | 002-TEST | 002-T | |
| Y | 002-TEST | 002-T | |
| | 004-TEST | 004-T | |
| Y | 004-TEST | 004-T | |
| | 004-TEST | 004-T | |

Graphical Handover Check Status

| | | | |
|---|---|---|---|
| Checksheet Type | | From Date | 01.01.2007 |
| COMM Pkg | | To Date | 30.11.2007 |
| MC Pkg | | Intervall | M |
| MC Site | | Include PA | N |
| MC Phase | | Periodic | N |
| | | Discipline | |

| Period | Planned in Period | OK/PB/PA in Period | Acc Planned | Acc OK/PB/PA | Acc Outstanding |
|---|---|---|---|---|---|

Summary Handover Certificate with Statistics and Document Status

| | O | Commpkg | Name | Type | MC Site No | MC Phase | Executor Company | No | Change By | Change Date | Issued by | Issue Date | Check Sheet Type |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | O | . | . | IHOC | AS | ON | | IHOC-.-AS-ON | IR | 16.04.2002 | IR | 16.04.2002 | MC |
| | | 001-TEST | 1111111111111 | | | | | | | | | | ALL |
| | | 002-TEST | 2222222222222 | | | | | | | | | | ALL |
| | | 003-TEST | 33345678888333 | | | | | | | | | | ALL |
| L | O | 004-TEST | 444444444444 | RFCC | AV | ON | | RFCC-004-TEST-AV-ON | OVS | 06.10.2005 | OVS | 06.10.2005 | MC |
| | | 0102-D03 | TEMPORARY POWER - D00 | IHOC | | | | TEST#IHOC | | | SOL | 06.09.2004 | ALL |
| | | 0102-D04 | Permanent POWER - D00 | | | | | | | | | | ALL |

Summary Check Status with Outstanding PL-Items

| Commpkg | Name | Checksheet Type | MC Site No | MC Phase | Executor Company No | Milestone | Status | OK | PA | PB | PC | PD | Outstanding Checksheets | Total Checksheets | Total Cleared/ Uncleared PA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| . | . | MC | AS | AT | | | | 0 | 1 | 1 | 1 | 0 | 2 | 5 | 3 |
| . | . | MC | AS | ON | | | PB | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 002-TEST | 2222222222222 | CP | AS | ON | | | PB | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 002-TEST | 2222222222222 | MC | AS | ON | | | | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 004-TEST | 444444444444 | CP | AS | AT | AB | | OK | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 004-TEST | 444444444444 | CP | AS | ON | AEL | | OK | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 004-TEST | 444444444444 | CP | AV | ON | AB | | PA | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 3 |
| 004-TEST | 444444444444 | MC | AS | ON | | | | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

**Figure 4: Screenshot of MIPS with many open views.**

Users do tasks that include one or more applications, and that makes it hard for them to keep track of what has been done.

Read more on today's situation in chapter 3.1.

## 1.3. The goal

The workflow system should be able to guide the user from task to task, no matter which application the task should be done in. It should be wizard-like, providing users help at each task. The visual design should be intuitive, with minimal choices.

Experienced users should also want to work with the workflow system because they find it faster to work with.

## *1.4. Reader's manual*

To get an overview of the situation and get under the skin of workflow technology one should read the thesis from beginning to end. If one is confident with the term workflow, one could skip chapter 2.

Chapter 1 through 3 covers all one need to know about the problem and all the theory involved. Chapter 4 gives an understanding of what has to be done along with suited technologies. Chapter 5 through 7 covers the work that has been done, and finally chapter 8 gives the conclusion.

# 2. Background

## 2.1. Technical background

Lately there have been some changes in Aker Kvaerner's policy regarding ICT. They have decided to only use data systems that run off the shelf. Microsoft has been chosen as their main supplier. That means that MIPS had to change its database server to Microsoft SQL server from Sybase.

The programming environment has to be Microsoft Visual Studio, and Aker Kvaerner's preferred programming language is C#, although I could choose between all technologies in Microsoft's .NET framework.

For database schemas, data models and such Microsoft Visio will be used.

## 2.2. Workflow theory

The presentation of workflow theory is built upon material found in [3] and [4].

A workflow can perhaps best be described as a cooking recipe with ingredients. The recipe describes each of the steps necessary to complete the cooking. In computer systems, the recipe describes each step necessary to complete the main task. The ingredients represent the various programs used in each step.

In business situations in the old days, some managers managed what we call workflows. They assigned tasks to resources, perhaps based on experience, training and so forth. In early days those resources were only people. People could i.e. write invoices by hand. Later some tasks were automated. Computers could sum up the totals and print it on a connected printer. People just had to enter the invoice data. Basically some tasks were automated, but the workflow management had changed little. Supervisors assigned work to resources, and clerks passed the work from one resource to another. Lists were made to keep track of the work so that nothing could get away, or to measure performance.

In the last 20 years technology has advanced, and we now have the technology to manage workflows automatically. The workflow process is now managed by a computer. This computer assigns tasks, passes these tasks on, and keeps track of them.

With the automated workflow system:

- Work doesn't get stalled or misplaced.
- Managers can focus on staff and business issues, instead of assigning tasks to resources
- The workflow process and its tasks are formally documented, and the tasks are then followed exactly as in the documentation.
- Tasks are assigned to the best resource (person or computer), and the most important tasks get assigned first.
- Tasks can be performed concurrently.

With this list we can assume that all the resources does work suited for them following the correct procedures. This ensures effectiveness, lower costs, and better service to customers.

Staff should also be happier because tasks are evenly distributed, and they are always working on the "right" task.

### 2.2.1. Work is done by the best available resource

Most workflow systems today use some algorithm to pick the best available resource based on some criteria. Storing user profiles in the system helps the system decide which resource to use. The criteria can be how qualified a person is to perform certain tasks, and how good this person is to do these tasks. Also, the supervisor could have a say in the selection.

Persons may be sick, and therefore have less capacity than normal. Then the system could perhaps assign easier tasks to this person. Persons that will be taking a vacation may want to finish tasks that are "in progress", so the workflow system should stop assigning tasks to this person.

There is another way of assigning tasks that will be more effective. In Aker Kvaerner users work in departments and have more or less the same qualifications and the tasks within the department are somewhat equal in the sense of what has to be done. Since we have these constraints, we would be allowing users to pick tasks from an inbox. This inbox will have tasks assigned to the user itself, or to the user's department. The user may select any task that lays open for the department. In the user's personal section of the inbox, the user will mostly find started but not finished tasks that the user has selected earlier. Inbox functionality is introduced as future work in section 8.3.2

### 2.2.2. Types of workflow

There are two basic types of workflow: "Assembly line" and "Once and done".

**Assembly line**
In this type there may be many people involved in a workflow. Some may gather information, some process the information and some make decisions based on the processed information. Using this type makes tasks in a workflow easier and do not require special skills for the people involved.

**Once and done**
This type requires skilled people to take care of one workflow. In other words, there is only one person involved in the workflow from start to end. This type minimizes the overhead, and will also minimize waiting time between tasks, as the same person is doing all tasks.

The best thing to do is to blend these two types. A large workflow in Aker Kvaerner will include work needed to be done by different departments. Then the best way will be to have the same person doing the tasks in the workflow that belongs to that person's department. The assignment should only change when tasks in the workflow actually changes department.

### 2.2.3. Interface to data systems

There are many levels of interfacing workflow systems to data systems. Workflow can be used without any interaction with business systems. This way a person gets a task that the person must handle without any help from the workflow system. The task may require the use of one or more business systems, but the user must tell the workflow system to move on when the user is finished with his assigned tasks. This is often referred to as a *side-by-side workflow system*.

In side-by-side situations users might get tired of opening the same business system over and over. The solution to this is to make a minimal interface that invokes this business system.

Some vendors may build workflow into their application to help users do their tasks within the application. This is called an *embedded workflow*. In embedded workflows, it is the application that drives the workflow further.

Today users often have to deal with many different business systems. The side-by-side or embedded workflow may then prove insufficient alone. In these situations it may be more practical to have the *independent workflow system* invoke all needed business systems. We say that the workflow drives the applications towards finish.

In this thesis the focus will be on a combination of independent and embedded workflows. The workflow system should invoke business systems used in the workflows, and also guide the user through the tasks within each business system.

### 2.2.4. Logging and tracking

Workflow systems typically log the processing history. The history contains records of when each task was done, and who the task was done by. It may also have a field for user comments. A user may explain why a task took so long, or that the user needs more information.

### 2.2.5. Workflow standards

In this thesis the standards made by The Workflow Management Coalition [1](WfMC) will be the fundament. Their workflow standards exist at three levels. They have a *Reference Model* which explains the big picture of how the standards fit together. The *Abstract Specifications* identifies each of the functions required and what data is involved. The *Bindings* are the details on how the specification is implemented with a basic set of tools, formats and protocols. We will now look further into the Reference Model.

The Reference Model has five identified interfaces to the workflow engine:

1. Process definition interchange. This definition includes the procedures that are included in a workflow and the resources (people, computers, groups) that perform the tasks. The workflow is usually designed in a separate modelling system, and then loaded into the workflow engine through interface 1. WfMC has made XPDL (Xml Process Definition Language) to accomplish this. This thesis will cover the use of XPDL as process definition language.
2. Workflow Client Application Programming Interface. This interface describes an API (Application Programming Interface) which handles the way processing of tasks should be done. This thesis will not cover the use of this interface.
3. Invoked applications. This interface covers workflow invoking applications, and applications invoking workflows. This thesis will only use ideas from this interface, and not use it as it is.

[1] WfMC's website http://www.wfmc.org contains lots of information about their work.

4. Interoperability. Describes the way different independent workflow systems work together, sharing information and such. This thesis focuses on one single independent workflow system, so this is not included further in the thesis.
5. Administration & Monitoring Tools. Aims to provide a common interface which enables several workflow services to share a range of common administration and system monitoring functions. It involves keeping history in workflows, and monitoring the total work performed. This thesis will not use this interface.

These interfaces are put together as shown in figure 5.

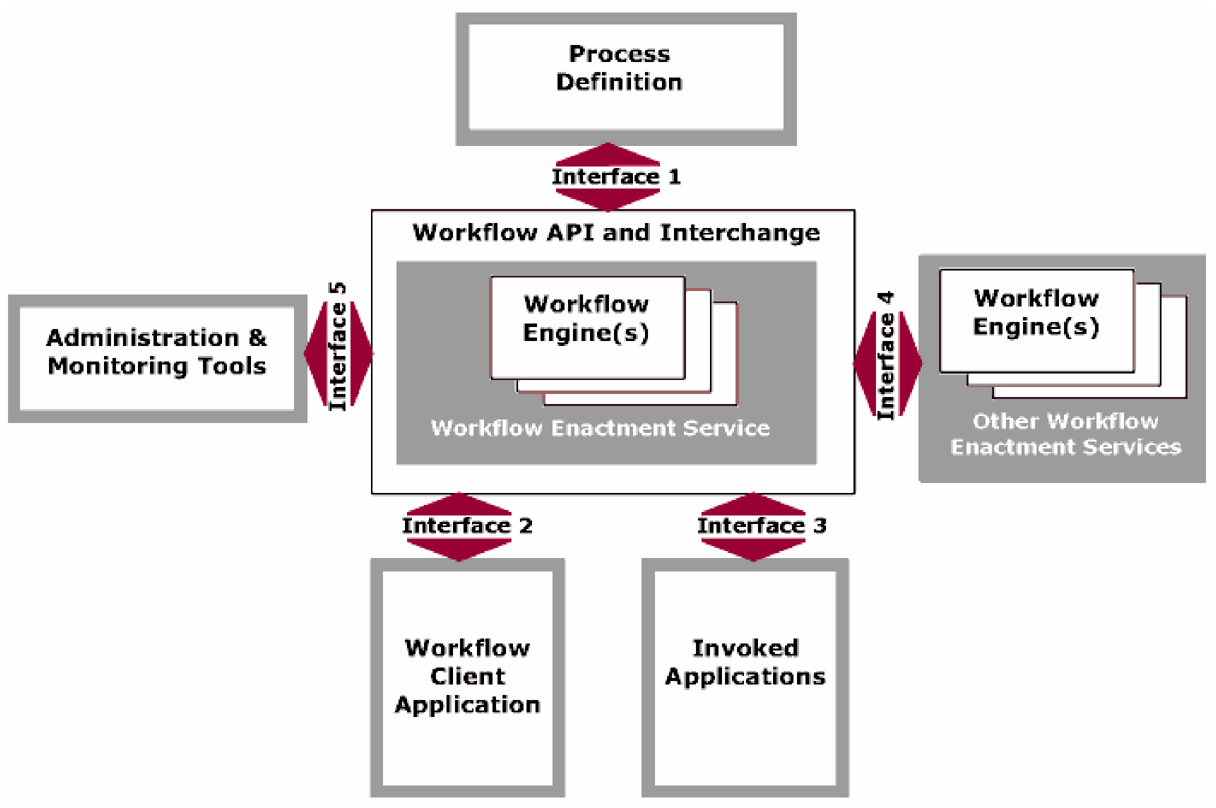

Process Definition
Interface 1
Workflow API and Interchange
Workflow Engine(s)
Workflow Enactment Service
Interface 5
Administration & Monitoring Tools
Interface 4
Workflow Engine(s)
Other Workflow Enactment Services
Interface 2
Interface 3
Workflow Client Application
Invoked Applications

**Figure 5: Five interfaces described by the reference model from WfMC. Taken from http://wfmc.org/graphics/processmodel_large.gif.**

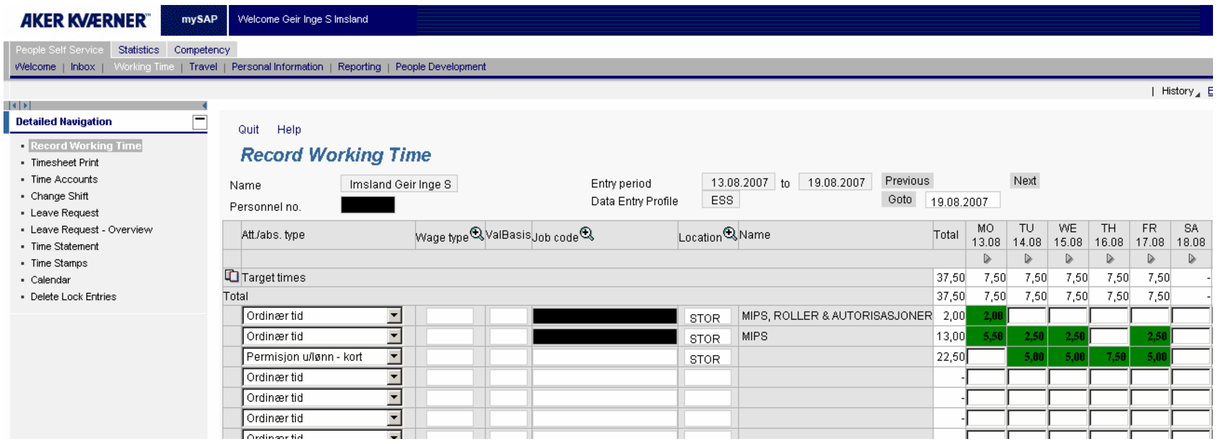
The thesis will introduce the desired use of these interfaces later on.

## 2.3. *Some systems using workflow*

There are actually a lot of systems using the concept of workflow today.

SAP is perhaps the most known system using workflow technology and it is of the embedded workflow type. In Aker Kvaerner every employee has knowledge about SAP. Every employee must record their working hours in SAP, and then send the timesheet to their supervisor for approval. If it is disapproved, the timesheet is sent back to the employee along with the reason for why it was rejected. If the working hours are approved, the supervisor sends the timesheet to the payroll office. The payroll office then schedules the payment on payment day.

In figure 6 one can the form used to register working hours.



AKER KVÆRNER mySAP Welcome Geir Inge S Imsland

People Self Service | Statistics | Competency

Welcome | Inbox | Working Time | Travel | Personal Information | Reporting | People Development

| History

Detailed Navigation

- Record Working Time
- Timesheet Print
- Time Accounts
- Change Shift
- Leave Request
- Leave Request - Overview
- Time Statement
- Time Stamps
- Calendar
- Delete Lock Entries

Quit Help

Record Working Time

| | | | |
|---|---|---|---|
| Name | Imsland Geir Inge S | Entry period | 13.08.2007 to 19.08.2007 |
| Personnel no. | [redacted] | Data Entry Profile | ESS |

Previous Next Goto 19.08.2007

| Att./abs. type | Wage type | ValBasis | Job code | Location | Name | Total | MO 13.08 | TU 14.08 | WE 15.08 | TH 16.08 | FR 17.08 | SA 18.08 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Target times | | | | | | 37,50 | 7,50 | 7,50 | 7,50 | 7,50 | 7,50 | - |
| Total | | | | | | 37,50 | 7,50 | 7,50 | 7,50 | 7,50 | 7,50 | - |
| Ordinær tid | | | [redacted] | STOR | MIPS, ROLLER & AUTORISASJONER | 2,00 | 2,00 | | | | | |
| Ordinær tid | | | [redacted] | STOR | MIPS | 13,00 | 5,50 | 2,50 | 2,50 | | 2,50 | |
| Permisjon u/lønn - kort | | | | STOR | | 22,50 | | 5,00 | 5,00 | 7,50 | 5,00 | |
| Ordinær tid | | | | | | - | | | | | | |
| Ordinær tid | | | | | | - | | | | | | |
| Ordinær tid | | | | | | - | | | | | | |
| Ordinær tid | | | | | | | | | | | | |

**Figure 6: Screenshot of mySAP.**

SAP is not only used for recording working time, it has in fact a large variety of uses. For instance, Komplett.no in Norway uses SAP as the backend for their web shop [1].

There are also several different workflow systems in the field of bioinformatics:

| | |
|---|---|
| **BioBike** | is a biocomputing platform based upon the KnowOS (Knowledge Operating System) e-science technology. Written entirely in Lisp, KnowOS's main distinguishing feature is "through-the-browser" programmability. |
| **DiscoveryNet** | is a £2m Engineering and Physical Sciences Research Council (EPSRC) - funded project to an e-Science platform for scientific discovery from the data generated by a wide variety of high throughput devices at Imperial College London. |
| **Geodise** | Grid Enabled Optimisation and Design Search for Engineering (GeoDise)] developed at the University of Southampton. |
| **Kepler** | enables scientists in a variety of disciplines like biology, ecology and astronomy to compose and execute workflows. Kepler is based on the Ptolemy II system for heterogeneous, concurrent modelling and design. Ptolemy II was developed by the members of the Ptolemy project at University of California Berkeley. Although not originally intended for scientific workflows, it provides a mature platform for building and executing workflows, and supports multiple models of computation. |
| **Pegasus** | is a flexible framework that enables the mapping of complex scientific workflows onto the grid developed at the Information Sciences Institute at the University of Southern California. |
| **Pegasys** | is a software for executing and integrating analyses of biological sequences, developed by the University of British Columbia. |
| **Taverna Workbench** | is an open source workflow system that enables scientists (typically, though not exclusively, in bioinformatics) to compose and execute scientific workflows. It has been developed as part of a £5.5m EPSRC project called myGrid based at the University of Manchester. Independently, other researchers have created Programming by example workflow development tools that are interoperable with Taverna. |
| **Triana** | is an open source problem solving environment developed at Cardiff University that combines an intuitive visual interface with powerful data analysis tools. |
| **Wildfire** | is a distributed, Grid-enabled workflow construction and execution |

| | |
|---|---|
| | environment. It has a graphical user interface for constructing and running workflows. Wildfire borrows user interface features from Jemboss and adds a drag-and-drop interface allowing the user to compose European Molecular Biology Open Software Suite (EMBOSS) (and other) programs into workflows. For execution, Wildfire uses GEL, the underlying workflow execution engine, which can exploit available parallelism on multiple CPU machines including Beowulf-class clusters and Grids. |

**Table 1: List of workflow based system used in the field of bioinformatics. Information is collected from [2].**

# 3. Problem description

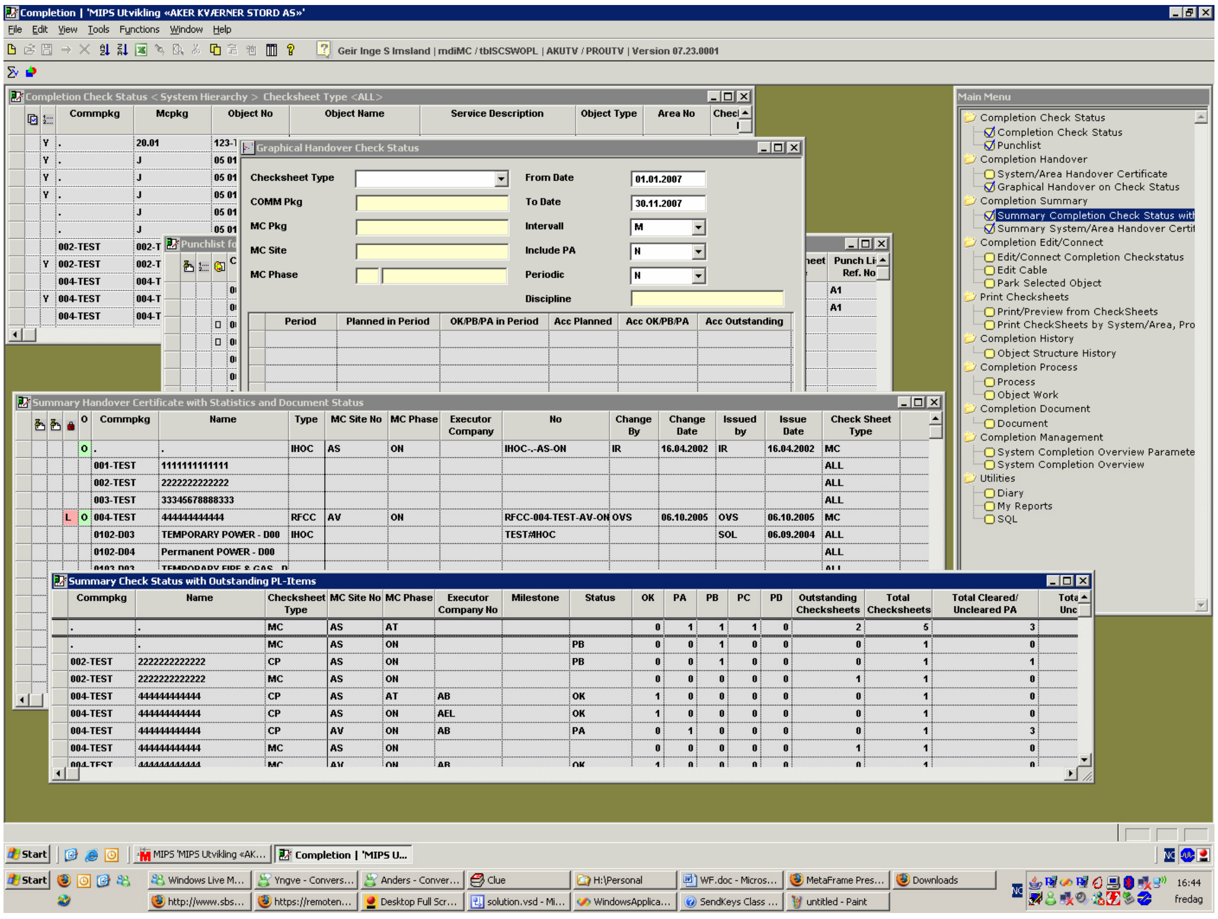
This chapter gives a short description of the problem.

## 3.1. *Today's situation*

A short overview of today's situation was presented in chapter 1.2.

As users do their tasks there are many different business systems involved. A user may work with a purchase order which must be entered in MIPS. In between operations in MIPS, the user has to use SAP to register the purchase order or check if a vendor exists. There could also be cases where emails must be sent. All in all this leads to a confusing process, where it is hard to keep track of what has been done. One can just imagine how hard this must be for new users.

What may be even more confusing are MIPS itself. When users do tasks in MIPS, they may be ending up with many open views. This may lead to confusion as the users loose their overview over the process. See screenshot of MIPS in figure 7 below for an example.



Completion Check Status < System Hierarchy > Checksheet Type <ALL>

| | Commpkg | Mcpkg | Object No | Object Name | Service Description | Object Type | Area No | Chec |
|---|---|---|---|---|---|---|---|---|
| Y | . | 20.01 | 123-1 | | | | | |
| Y | . | J | 05 01 | | | | | |
| Y | . | J | 05 01 | | | | | |
| Y | . | J | 05 01 | | | | | |
| | . | J | 05 01 | | | | | |
| | . | J | 05 01 | | | | | |
| | 002-TEST | 002-T | | | | | | |
| Y | 002-TEST | 002-T | | | | | | |
| | 004-TEST | 004-T | | | | | | |
| Y | 004-TEST | 004-T | | | | | | |
| | 004-TEST | 004-T | | | | | | |

Graphical Handover Check Status

| | | | |
|---|---|---|---|
| Checksheet Type | | From Date | 01.01.2007 |
| COMM Pkg | | To Date | 30.11.2007 |
| MC Pkg | | Intervall | M |
| MC Site | | Include PA | N |
| MC Phase | | Periodic | N |
| | | Discipline | |

| Period | Planned in Period | OK/PB/PA in Period | Acc Planned | Acc OK/PB/PA | Acc Outstanding |
|---|---|---|---|---|---|

| heet | Punch Li Ref. No |
|---|---|
| | A1 |
| | A1 |

Summary Handover Certificate with Statistics and Document Status

| | O | Commpkg | Name | Type | MC Site No | MC Phase | Executor Company | No | Change By | Change Date | Issued by | Issue Date | Check Sheet Type |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | O | . | . | IHOC | AS | ON | | IHOC-.-AS-ON | IR | 16.04.2002 | IR | 16.04.2002 | MC |
| | | 001-TEST | 1111111111111 | | | | | | | | | | ALL |
| | | 002-TEST | 2222222222222 | | | | | | | | | | ALL |
| | | 003-TEST | 33345678888333 | | | | | | | | | | ALL |
| L | O | 004-TEST | 444444444444 | RFCC | AV | ON | | RFCC-004-TEST-AV-ON | OVS | 06.10.2005 | OVS | 06.10.2005 | MC |
| | | 0102-D03 | TEMPORARY POWER - D00 | IHOC | | | | TEST#IHOC | | | SOL | 06.09.2004 | ALL |
| | | 0102-D04 | Permanent POWER - D00 | | | | | | | | | | ALL |

Summary Check Status with Outstanding PL-Items

| Commpkg | Name | Checksheet Type | MC Site No | MC Phase | Executor Company No | Milestone | Status | OK | PA | PB | PC | PD | Outstanding Checksheets | Total Checksheets | Total Cleared/ Uncleared PA | Tota Unc |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| . | . | MC | AS | AT | | | | 0 | 1 | 1 | 1 | 0 | 2 | 5 | 3 | |
| . | . | MC | AS | ON | | | PB | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | |
| 002-TEST | 2222222222222 | CP | AS | ON | | | PB | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | |
| 002-TEST | 2222222222222 | MC | AS | ON | | | | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | |
| 004-TEST | 444444444444 | CP | AS | AT | AB | | OK | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | |
| 004-TEST | 444444444444 | CP | AS | ON | AEL | | OK | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | |
| 004-TEST | 444444444444 | CP | AV | ON | AB | | PA | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 3 | |
| 004-TEST | 444444444444 | MC | AS | ON | | | | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | |

**Figure 7: Screenshot of MIPS with many open views.**

## 3.2. *Why today's situation is not satisfactory*

Based on what was covered in chapter 1.2 and chapter 3.1 one can easily see that the way users work may be improved radically. The way users work today has these problems:

- Each user has its own way of completing tasks. Which way are most correct?
- Speed. Users may not work in an optimal way.
- Hard to keep track of what has been done.
- Some tasks may end up not being done properly.
- New users will have a hard time learning all the business systems, and each task that involves these business systems.

It is fairly easy to see that every user may have its own ways of doing tasks, and in many cases it is not possible to say that a specific way is not right.

Some users may be working faster than others. This may not be only because of skills or training - users working faster may have found a more optimal way of doing their tasks. This leads us to the fact that all users should follow the most optimal way to optimize their total work.

It is also fairly easy to see that users can loose track of what has been done. If a task is not properly done, or not done at all, this may never be discovered. It will eventually lead to errors. Perhaps a customer ends up with an invoice that shows a lower total than it was supposed to. We see that a task not properly done could have negative effects on Aker Kvaerner's economy.

How can new users feel confident in themselves, when all the other experienced users do their tasks in different ways? They would probably not feel confident at all. When a new user asks for help, the information he gets could have differences. The new user could end up being even more confused. And this is all in addition to the difficulty of learning all involved business systems. Training new users could end up taking much longer time than necessary.

## 3.3. *Definition of the assignment*

The task in this thesis work is to make a system that takes care of all issues which has been explained in chapter 3.2.

The workflow system must:

- Guide users through their tasks in the most optimal way, and have intuitive design.
- Keep track of what has been done, so that no tasks are left undone or unfinished.
- Do its work without interfering with the behaviour of business applications included in its workflow.
- Be able to invoke all business applications needed to complete a workflow.

As said in chapter 0, the main focus of this thesis will be on MIPS. The workflow system must be able to guide users from task to task in MIPS. It must also keep track of what has been done in order to ensure that all tasks are properly done.

That being said, one must also plan ahead. The workflow system will include other applications than MIPS some time in the future. In order to accomplish that, the planning will show that other applications can be used.

In chapter 2.2.3 we saw that some applications have embedded workflows. SAP is one business application that has this feature. In cases where the workflow includes such business applications, the business application's embedded workflow should be triggered. There is no need to make the workflow system follow users from task to task inside a business application if there are embedded workflows doing that.

# 4. Problem analysis

This chapter covers the problem definition, dividing the main problem into sub problems and discusses available technologies that can be used.

## *4.1. Problem definition*

A system capable of handling workflows must be made. From now on this system will be called the Workflow Management System (WMS).

Keeping in mind that this thesis focuses on workflows in MIPS the WMS must:

- Be able to import workflows made by some workflow modelling software that can save the workflow in XPDL format.
- Run workflows suited for MIPS.
- Invoke applications included in workflows (At least MIPS).
- Show and hide views inside MIPS as needed in each task in a workflow.
- Keep track of each task in the workflow.
- Have easy and intuitive design that users understand.
- Wizard-like behaviour

There are also more concerns. For instance, there is no way that existing applications in Aker Kvaerner can be tampered with. This fact leads us to that the only way of controlling work units inside applications in a workflow is to simulate user control.

For MIPS, it is possible to start its different modules from command line. Once a module is running, simulation of key presses and mouse clicks can be used for controlling the views and menus inside the module.

### 4.1.1. Standalone or server/client

The WMS can be built either as a standalone application or a server/client-based application. WfMC's reference model gives four different types of server/client-based applications. These can be seen in the figure 8 below.

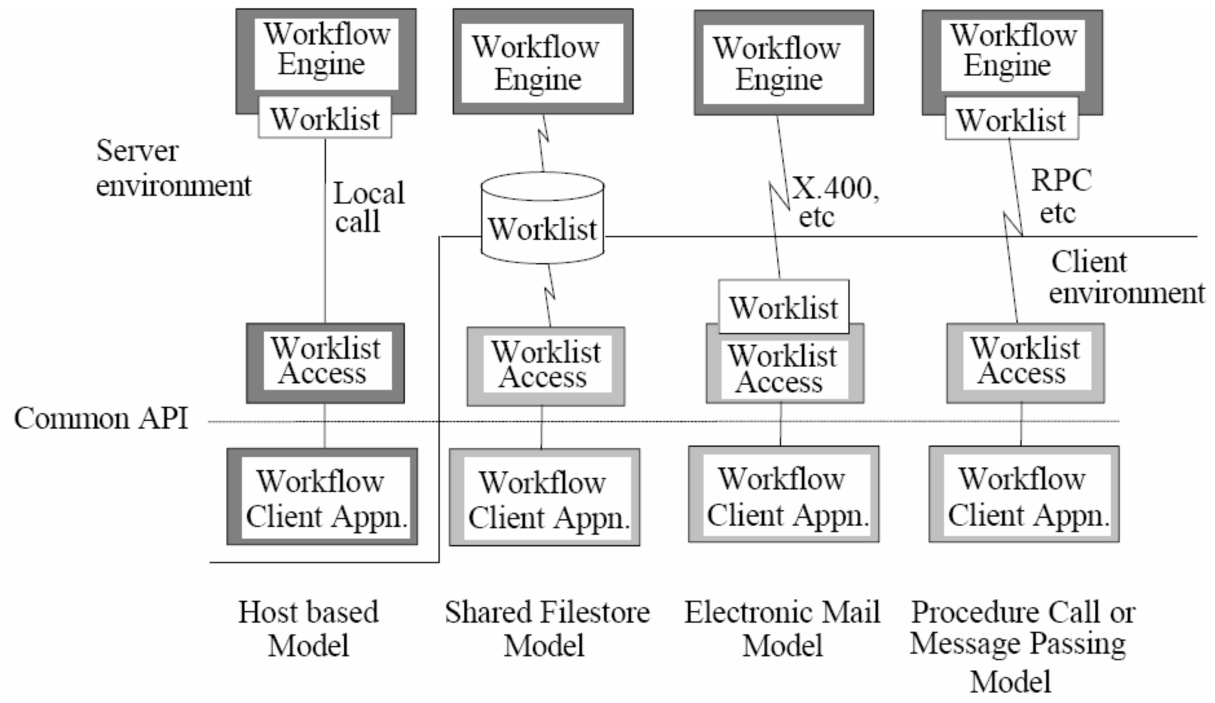


**Figure 8: Alternative client work list handler implementations. Figure extracted from the WfMC Workflow reference model [8].**

The WfMC reference model mentions only server/client based workflow systems. The WMS covered in this thesis will only use local workflows, meaning there will be no shared workflows amongst users. In other words, a server/client based system is not necessary. The WMS can be deployed on users workstations and execute workflows locally.

## *4.2. Sub problems*

Based on what was covered in chapter 4.1 the problem can be divided into these parts:

- Workflow importer for XPDL
- Workflow engine
- Visual design for the WMS.

### 4.2.1. Workflow importer

The workflow importer for the WMS will be responsible for importing workflows in XPDL format. The importer will convert the specifications from the XPDL format into objects that will be used by the workflow engine to process the workflow.

### 4.2.2. Workflow engine

The workflow engine will process workflows. It will be able to keep track of what has been done in each task inside a workflow. The engine will also provide the user interface.

### 4.2.3. WMS visual design

As stated in chapter 4.1 the visual design of the WMS must be intuitive and easy to understand for all users. If the design is not good enough, the WMS will just be another burden.

A graphical user interface (GUI) is poor if: [11]

- The user is forgotten. Developers often design for what they know, not what the user knows.
- The software controls the user. The user should be controlling the software, not the other way around.
- The GUI provides too many features at top level. Only common features should be at top level. Less used features can be “hidden” under menus.

Good GUI design principles: [11]

- Allow the user to choose items instead of writing them.
- Use standard icons as far as possible.
- Use terms that users are familiar with.
- Do not mix designs. Be consistent!
- Provide information. Use progress indicators.
- Do not use sounds. It will become annoying to users.
- Concise text. Long messages can be unclear.
- Let the user know how to navigate in the GUI.
- Use shortcut keys.
- Consistent look and feel.
- Be sure when to use dialog boxes or windows.
- Minimize the use of control components.

An example of really poor GUI design can be seen in figure 9.

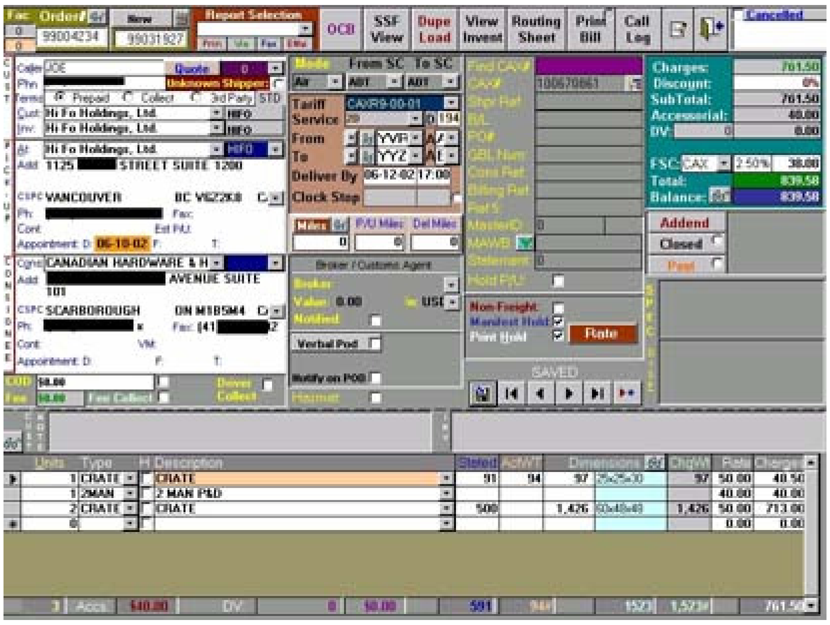


**Figure 9: Bad GUI design. Collected from http://www.ssw.com.au/ssw/Standards/Rules/Images/badui2.jpg.**

## *4.3. Available technologies*

This section covers the available technologies for the WMS. There are two main competitors in workflow technology: Workflow Management Coalition (WfMC) and Windows Workflow Foundation.

### 4.3.1. Workflow Management Coalition

The WfMC does not actually provide any programming APIs. Instead the WfMC creates and contributes to process related standards, educates the market on related issues. The WfMC is the only standards organization that concentrates purely on process. They have created Wf-XML and XPDL. XPDL is the leading process definition used in over 50 solutions to store and exchange process models. [5]

The Wf-XML is a XML based protocol for run-time integration of process engines. It provides ways to communicate with process engines, and can get information about activities that currently are being waited for and who or what an activity is waiting for. It can also retrieve process definitions, as well as add new ones. The Wf-XML is currently on a draft stage, and requires a lot of extra work if it is to be used. [6]

XPDL is a process definition language in XML. The goal of XPDL is to store and exchange process definition models, and let other tools read and edit them. Also to let tools run the process. XPDL is currently in version 2. XPDL can store coordinates of visual objects, lines, pointers etc. This is all in addition to the more technical definition of the process, like which applications to invoke, description of all activities and so on. [7]

The WfMC has also a workflow reference model that identifies workflow management systems' characteristics, terminology and components. The reference model allows individual specifications to be developed within the context of an overall model for workflow management systems. [8]

### 4.3.2. Windows Workflow Foundation

This presentation of Windows Workflow Foundation is built upon material found in [9].

Windows Workflow Foundation (WWF) provides a common workflow technology for Windows. It also offers a workflow framework for Windows applications, and it unifies system and human workflow.

There are several products for Windows that have embedded workflows. In Microsoft's own products, workflow technology is available in Microsoft BizTalk Server, Microsoft Exchange Server and others.

WWF's goal is to incorporate workflow support into the Windows platform. It is part of the .NET framework 3.0, and is also native in Windows Vista. In the future all Microsoft products will use this technology.

WWF provides a general framework for creating and executing workflows in Windows applications. Workflows can be designed using the provided Workflow designer, or they can be written in code directly. The Workflow designer hosted in Microsoft Visual Studio 2005 can be seen below in figure 10.

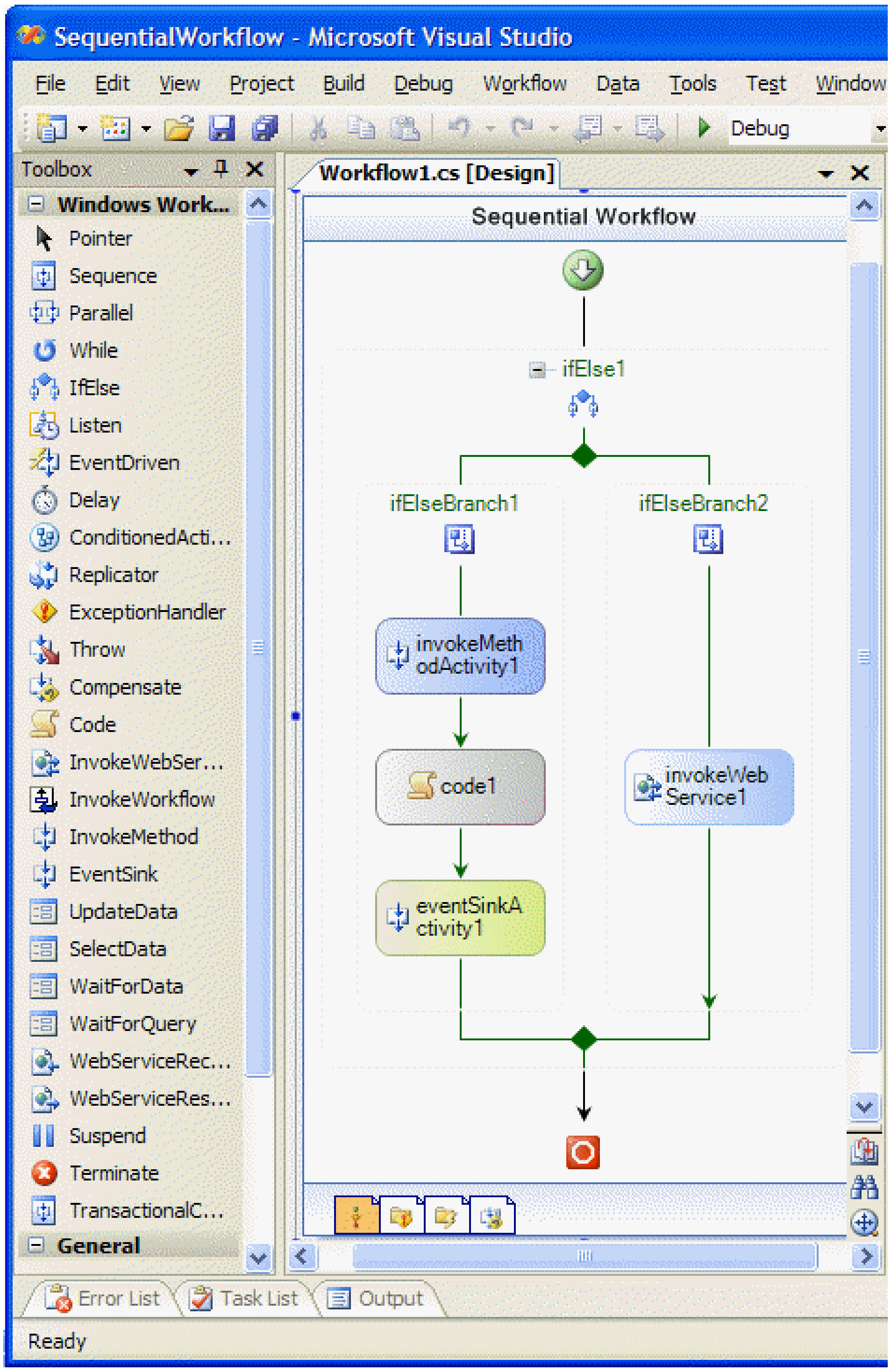


**Figure 10: Workflow Designer hosted in Microsoft Visual Studio 2005. Image is taken from [9].**

The WWF has these main components:

- Activity
- Workflow
- WWF designers
- WWF base activity library
- WWF runtime engine

- Host process

The activity represents a unit of work. The complexity of the unit of work can range from very simple to very complex. The workflow represents a group of activities that implements all or part of a business process. The WWF designers are graphical tools that can be used to create and modify workflows and activities. The WWF base activity library is a fundamental group of activities that can be used for creating workflows. The WWF runtime engine is a library that executes the workflows. It also provides services for communicating with software outside the workflow. The host process hosts the WWF runtime engine and any workflow it executes. The host process also keeps track of a workflows state, handling transactions and other functions.

Based on what was covered in chapter 2.2, it is easy to see that the WWF has all the components needed in a workflow management system.

### 4.3.3. Other technologies

There are other technologies, but they are mainly built for special purposes. There are also some workflow engines that are open source and could be tailored to Aker Kvaerner's needs. However, these systems tend to be written in other languages than those included in Microsoft's .NET framework, and could therefore not be considered.

Some examples of other technologies:

| | | |
|---|---|---|
| Skelta BPM.NET 2007 Enterprise Edition | http://www.skelta.com/ | Skelta BPM.NET 2007 Enterprise Edition enables business users, analysts, managers and other stakeholders to collaborate with developers in process design, implementation, integration and administration of an enterprise's BPM environment. |
| jBoss jBPM | http://www.jboss.com/products/jbpm | With jBoss jBPM you can create business processes that coordinate people, applications, and services. Designed for SMB and large enterprise applications alike, JBoss jBPM brings process automation to a much wider set of business problems, from embedded workflow to enterprise business process orchestration and BPM. |

| | | |
|---|---|---|
| Galaxia | http://workflow.tikiwiki.org/ | Galaxia is the ideal tool for implementing business processes. It is well-suited for use in corporate environments, where it can be used for tracking requests, product QA, and any other situation where people from many different areas must collaborate. |
| Bossa Workflow System | http://www.bigbross.com/bossa/ | Bossa is a workflow engine written in Java. The engine is very fast and lightweight, uses a very expressive Petri net notation to define workflows, does not require a RDBMS and is very simple to use and to integrate with java applications.<br><br>Actually, it was **designed** to be embedded. Therefore, Bossa is organized as a library to be used by server side applications (especially web oriented ones) that need workflow services. We plan, in the future, to implement a shell over the current library to offer workflow services as a stand alone server. |

**Table 2: Some samples of other workflow technologies.**
:

### 4.3.4. Choosing the technology

While WWF clearly has most of the components needed in a workflow management system, it cannot be used in Aker Kvaerner. Most applications used in project planning in Aker Kvaerner are not built on the .NET platform. MIPS is programmed in Centura.

Due to the fact that most of the applications used in workflows in Aker Kvaerner are not built on .NET technology, the only choice is to use the technology provided by WfMC.

## 4.4. The chosen technology

The technology provided by WfMC will be used. XPDL will be used to store and exchange workflow models. The WMS will be built within the context of WfMC's workflow reference model, although some parts will be left out of the solution.

As stated in chapter 2.1 there really is no choice regarding programming environment or tools. Microsoft's products will be used according to Aker Kvaerner's ICT policy. There is also no reason not to use Aker Kvaerner's preferred programming language. To sum up, the following tools and technologies will be used:

- Visual Studio 2005 with C# as programming language
- Microsoft SQL server 2005
- Microsoft Visio 2003
- XPDL
- Concepts found in WfMC's workflow reference model

# 5. Solution

This chapter covers the overall solution.

## *5.1. System model*

### 5.1.1. The starting point for the workflow engine

This presentation of the starting point is based on material found in [8] and [10].

There is a very good meta model that describes an overview of the different parts in a workflow management system. It actually belongs to the WfMC interface 1, about process definition. This model will be the starting point for the WMS. The model can be viewed in figure 11.

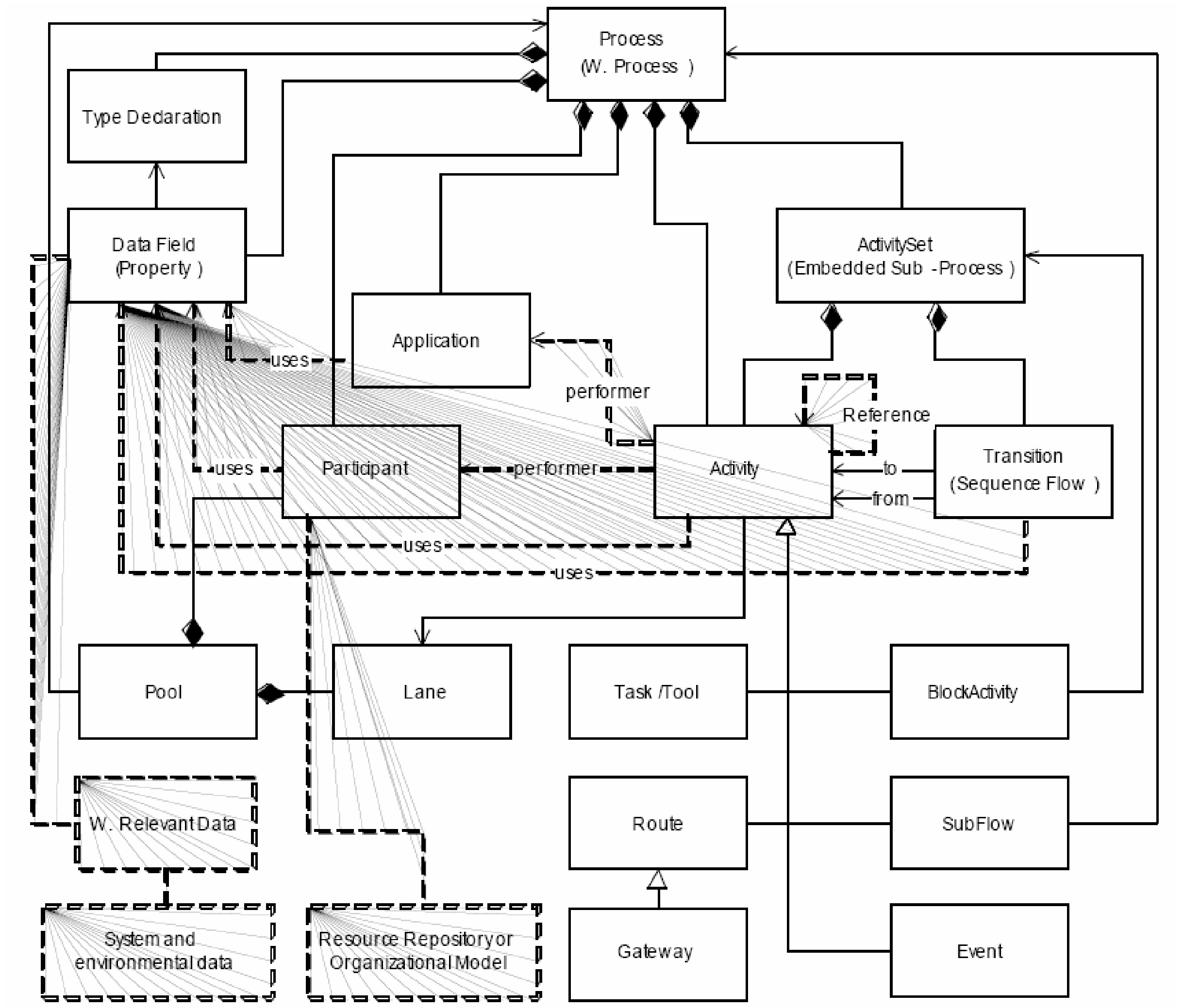


**Figure 11: Process definition meta model. The model comes from [10].**

The following text will explain the most important entities in the process definition meta model.

The process (workflow process) entity provides contextual information that applies to other entities within the process. The process entity consists of one or more activities that contain a work unit within the process. These activities belong to a lane. Lanes subdivides a pool, which act as a container for flow objects (activities) and sequence flow (transitions) between them. Activities can cross the borders of a lane, but not the borders of a pool. A typical use of lanes is to give them role names, and then have the activities inherit these role names. An activity may also be an activity set, which is a sub workflow.

To get a better understanding of pools, lanes and activities, a workflow diagram is shown in figure 12.

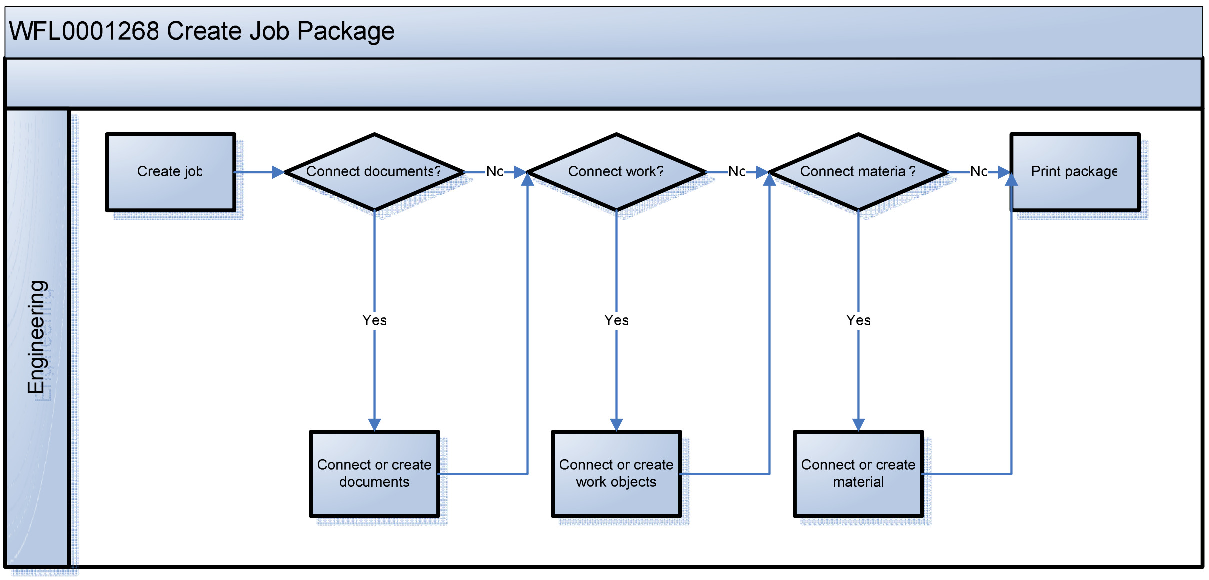

WFL0001268 Create Job Package
Engineering
Create job
Connect documents?
No
Connect work?
No
Connect material?
No
Print package
Yes
Yes
Yes
Connect or create documents
Connect or create work objects
Connect or create material

**Figure 12: A simple workflow for creating a job package in MIPS.**

Figure 4 shows a pool with only one lane. The lane has the name of the department it belongs to. It consists of two types of activities. The square ones are ordinary MIPS activities. The others are a special type of activity, called decision activities. These provide the user with a question with answer alternatives. Based on what the user responds, the user will be guided to the next task, or question.

### 5.1.2. Simplified model for the workflow engine

Some of the entities in the original model are not valuable for workflows concerning MIPS. For that reason, a new somewhat simpler model has been developed. The simplified model can be seen in figure 13.

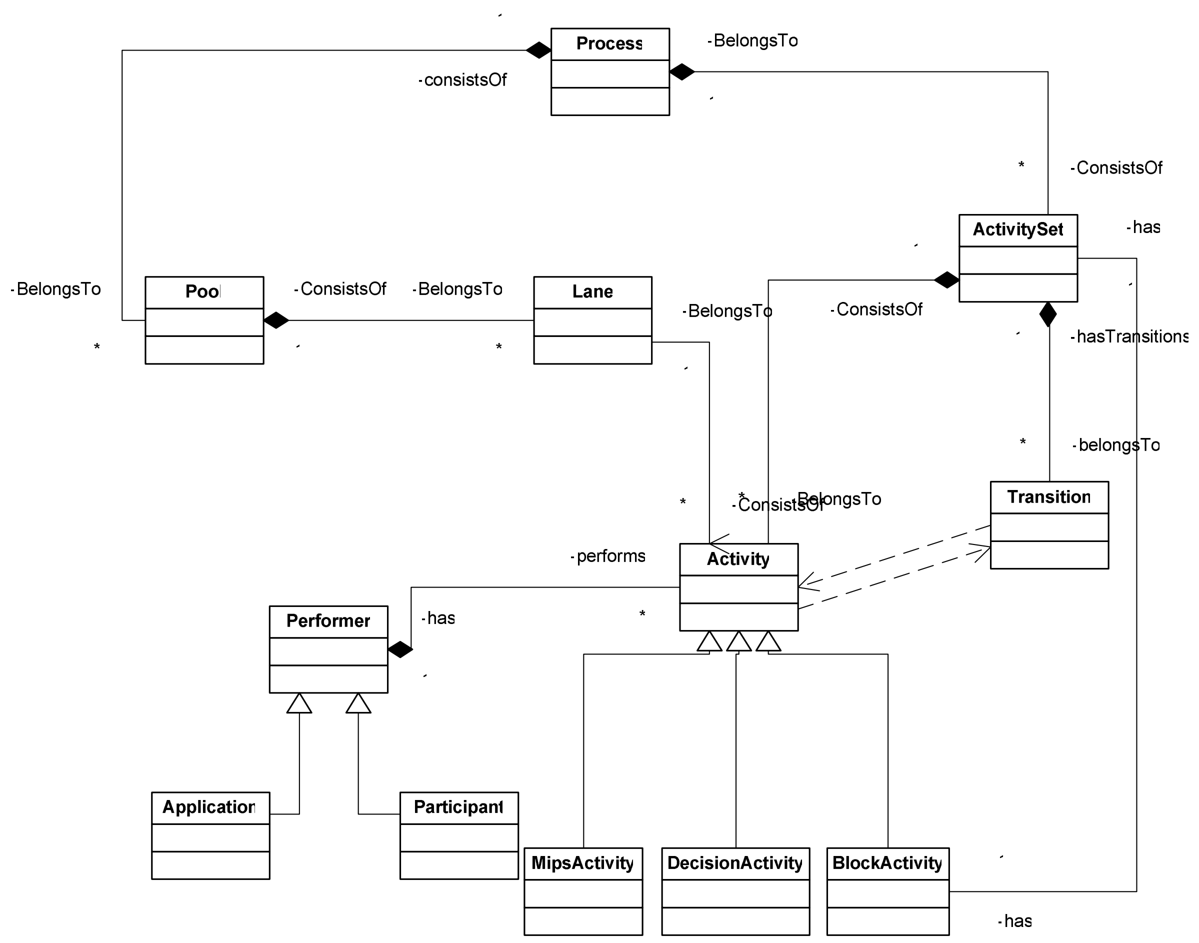


**Figure 13: Simplified model for the workflow engine.**

The main differences are that the two performers, Application and Participant, can be generalized into one class, Performer. The Performer performs the Activity, which can be a MipsActivity, DecisionActivity or BlockActivity. The BlockActivity is an activity which executes an ActivitySet (sub workflow). Activities have sequence flows (transitions) between them. The Pool and Lane entities have the same properties seen in figure 11.

## *5.2. System architecture*

### 5.2.1. Solution overview

Figure 14 shows how the system is put together. Both the user interface and workflow importer are strongly connected to the workflow engine.

The workflow importer is responsible for importing XPDL formatted workflows. It will convert objects in the XPDL formatted file to objects in the workflow engine.

The user interface is responsible for communication with the user and the workflow engine.

The workflow engine is responsible for holding all objects needed in a workflow. It will start all invoked applications specified by a workflow definition. It will also execute and end activities on signal from the user interface.

The performer is a user communicating with the WMS' user interface and the invoked applications.

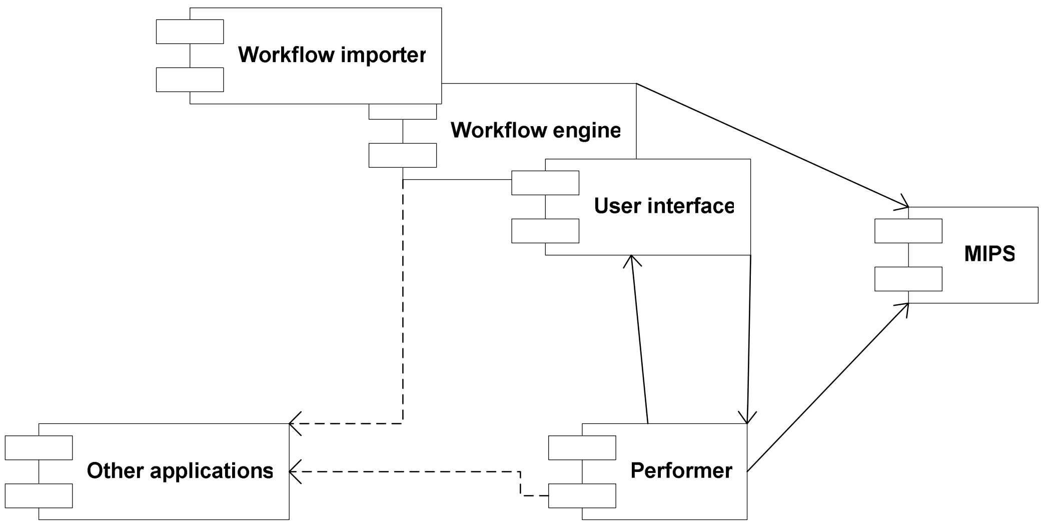


**Figure 14: WMS solution overview.**

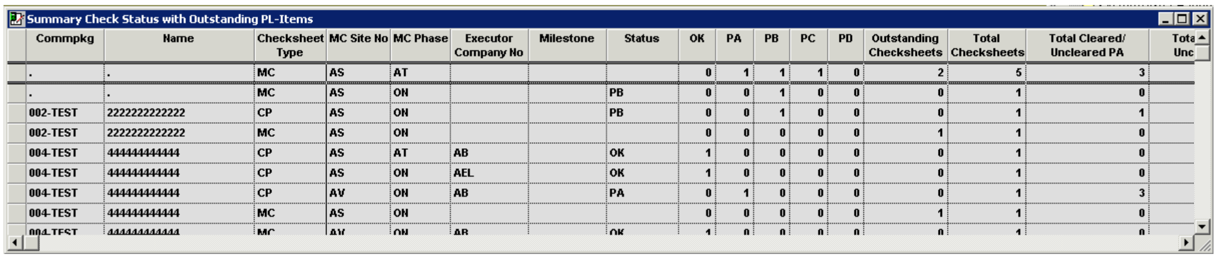
### 5.2.2. Workflow engine

The workflow engine has essentially two responsibilities besides keeping control of the workflow.

It has to invoke applications used in workflows. This is accomplished by creating a class for each "special" way an application can be started. MIPS has several modules, and each of them can be started from console. Applications like MIPS fits into a class for command line applications. Other applications may not be started directly from command line. Such applications need separate classes so that the necessary functionality can be added in order to start them.

The workflow engine must also be able to simulate user control of MIPS and other applications included in a workflow. This is accomplished by first bringing the application in question to focus, and then simulating key presses and/or mouse clicks. It may sound like a quite silly solution, but the fact is that it is very effective. The simulation of key presses and mouse clicks can be used in any application. By using this kind of simulation one doesn't have to come up with new special ways to communicate with each new application. The specification of the key presses and mouse clicks lies within one of the Activity-classes. For MIPS this is the MipsActivity.

Most functions and views in MIPS have shortcut keys. They can be activated by a short key press sequence like CTRL-SHIFT-R. If such a shortcut is not present, a key press sequence with many key presses must be made. An example can be DOWN-DOWN-DOWN--DOWN-DOWN-DOWN-DOWN-ENTER. This key press sequence will open the view in figure 15 if MIPS' Completion module is the invoked application.


Summary Check Status with Outstanding PL-Items

| Commpkg | Name | Checksheet Type | MC Site No | MC Phase | Executor Company No | Milestone | Status | OK | PA | PB | PC | PD | Outstanding Checksheets | Total Checksheets | Total Cleared/ Uncleared PA | Tota Unc |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| . | . | MC | AS | AT | | | | 0 | 1 | 1 | 1 | 0 | 2 | 5 | 3 | |
| . | . | MC | AS | ON | | | PB | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | |
| 002-TEST | 2222222222222 | CP | AS | ON | | | PB | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | |
| 002-TEST | 2222222222222 | MC | AS | ON | | | | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | |
| 004-TEST | 444444444444 | CP | AS | AT | AB | | OK | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | |
| 004-TEST | 444444444444 | CP | AS | ON | AEL | | OK | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | |
| 004-TEST | 444444444444 | CP | AV | ON | AB | | PA | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 3 | |
| 004-TEST | 444444444444 | MC | AS | ON | | | | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | |
| 004-TEST | 444444444444 | MC | AV | ON | AB | | OK | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | |

**Figure 15: View in MIPS' Completion module.**

Activities like the DecisionActivity instructs the user interface to give the user a question with the desired answer alternatives. An example can be seen in figure 16. It will not be sufficient with to buttons saying YES or NO. Questions can have more than two answers so radio buttons are used.

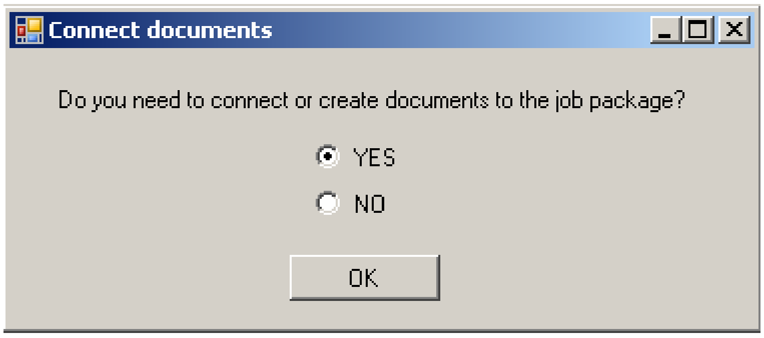

Connect documents
Do you need to connect or create documents to the job package?
● YES
○ NO
OK

**Figure 16: Example question provided by the user interface.**

### 5.2.3. Workflow importer

The workflow engine makes use of a workflow importer that reads XPDL formatted files. The importer simply converts objects in the XPDL formatted file to objects in the workflow engine. This is very easy to do with the tools the .NET technology provides.

### 5.2.4. User interface

The user interface lies on top of the workflow engine. It communicates directly with the workflow engine's objects.

The user interface has two modes. The normal mode suits new users and users not familiar with a specific workflow. The expert mode suits expert users.

A screenshot of the user interface in normal mode can be seen in figure 17. In the Navigator section of the user interface, screenshots of the last, current and next activity are shown. This will help users to find out where exactly they are in the workflow. There are two buttons called Previous and Next. These provide functionality for navigating to the previous activity and the next activity. At the bottom there is a field for help text. It will provide the user with help for the current activity.

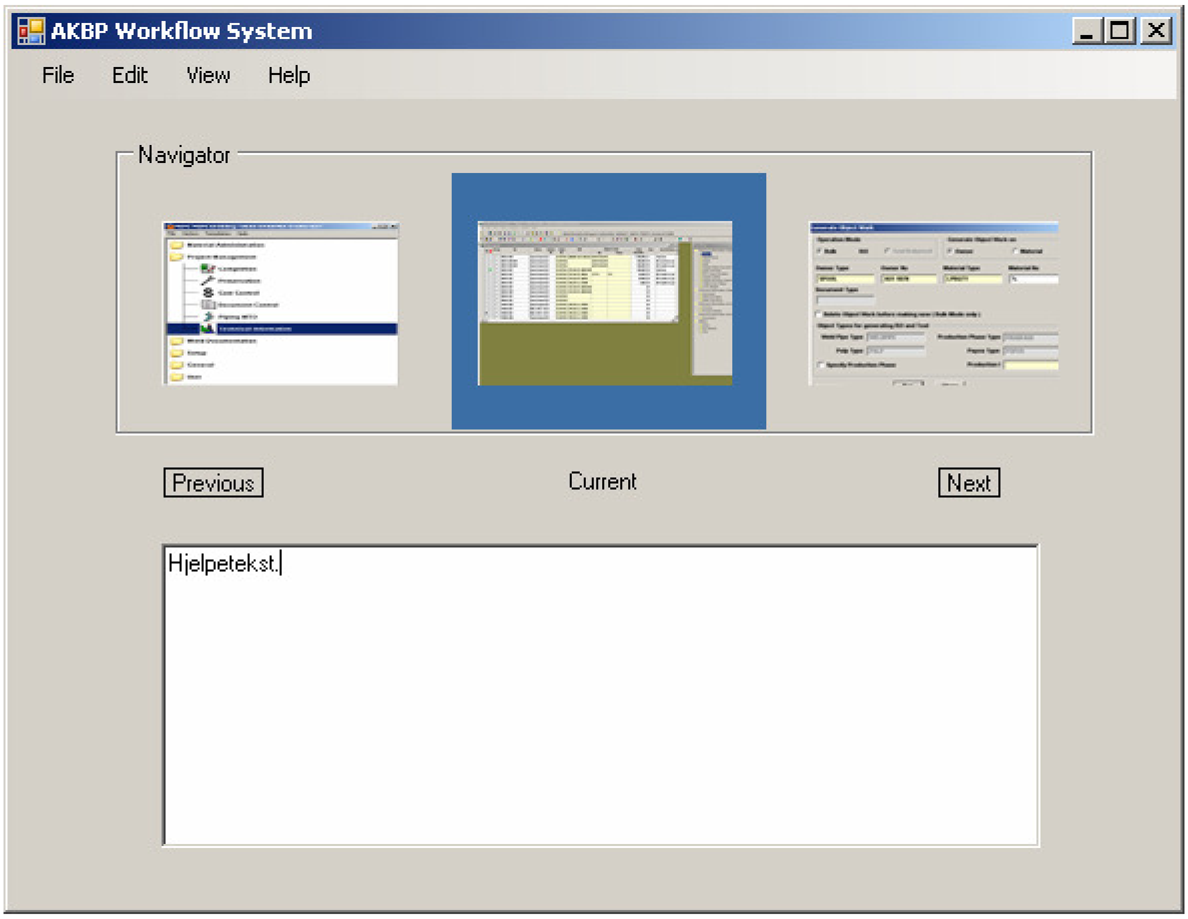


**Figure 17: Screenshot of the AKBP Workflow System user interface in normal mode.**

Figure 18 shows the user interface in expert mode. It has two buttons which provides navigation to the previous and next activity.

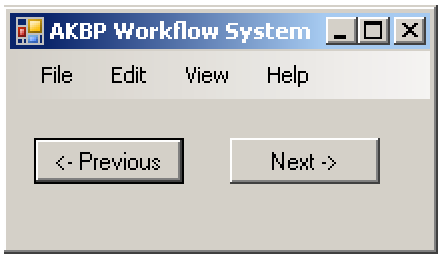


**Figure 18: Screenshot of the AKBP Workflow System user interface in expert mode.**

Both user interface modes are configured to always be on top of other open applications in Windows. The normal mode is perhaps not suited to be on top because of the screen space, but users will feel more confident with their aid close.

The conditions for a good GUI design mentioned in section 4.2.3 are met.

# 6. Implementation

This chapter covers implementation of special sections in the workflow engine.

## *6.1. Starting invoked applications*

Program listing 1 shows how the WMS starts invoked applications. In chapter 3.3 it was stated that the WMS should be able to invoke all kinds of business applications. The code below shows how MIPS is started. MIPS is an application that can be started through command line. The code in program listing 1 is easy extendable making it easy to implement start up procedures for applications not started through command line.

Invoked applications are to be read from a XPDL formatted file and converted into an object in the workflow engine.

The lines 2 through 8 contain a loop. This loop goes through all applications that should be invoked. The WMS will only start command line applications as of now.

The lines 10 through 27 show the method for starting a command line application. First, a Process must be instantiated. In line 15, the path and filename of the application is inserted into the Process' start info along with environment variables in line 16. The environment variables are typically paths to the application's needed libraries. MIPS for instance needs its communication libraries. Environment variables are provided through the definition of the invoked application in XPDL. After the application is started, the application's running state is set to true. The associated process is also connected to the application. That makes it easy to use the process in activities later on.

If the process can not be started, an error will be raised to the WMS's error handler. The error handler will instruct the user interface to give the error to the user, and then stop the WMS.

```
1 .....
2 foreach (app in invokedApplications)
3 {
4     if (app.type == "commandLine")
5     {
6           this.startCommandLineApplication(app);
7     }
8 }

9 .....

10 private void startCommandLineApplication(invokedApplication app)
11 {
12    Process proc = new Process();
13    try
14    {
15          proc.StartInfo.FileName = app.FileName;
16          proc.StartInfo.EnvironmentVariables = app.EnvironmentVariables;
17          proc.Start();
18          app.setAppStarted(true, proc);
19    }
20    catch (Win32Exception e)
21    {
        if(e.NativeErrorCode == ERROR_FILE_NOT_FOUND)
        {
              ErrorHandler.raiseError(e.Message + ". Could not start
              application because it's startup file was not found");
        }
  }
}
```
**Program listing 1: Starting invoked applications in a workflow.**

## *6.2. Starting work units in activities*

The WMS executes activities in a workflow based on the code in program listing 2. Notice the little amount of code for activity execution. If one were to build interfaces into MIPS the amount of code would be substantially larger.

The WMS must be able to gain control of user supported functions. This is shown in the lines 1 through 3. The call SetForegroundWindow is located in the windows DLL file user32.dll.

An activity is executed by first getting the activity's associated application. Then, the application's process must be located. The process of an application is accessed via the invokedApplication object. The window handle of the process is used to get the process' window into focus. After the process is focused, one can send key presses to it. All this is shown in the lines 5 through 19 in program listing 2.

If the window handle of the process can not be accessed, an error is raised to the WMS's error handler. This will provide an error message through the WMS's user interface which tells the user to restart the workflow.

The process of sending key presses to the focused application is possible using the SendKeys class located in System.Windows.Forms namespace. The key presses needed to execute and end an activity are provided along with the rest of the Activity definition in a XPDL file. The key presses are stored in each Activity object.

```
1 [DllImport("user32.dll")]
2 [return: MarshalAs(UnmanagedType.Bool)]
3 static extern bool SetForegroundWindow(IntPtr hWnd);

4 .....

5 public void executeAcivity(Activity a)
6 {
7     invokedApplication invokedApp = a.getAssociatedApp();
8     Process proc = invokedApp.getAssociatedProc();
9     IntPtr windowHandle = proc.MainWindowHandle;
10    if (windowHandle != IntPtr.Zero)
11    {
12          this.SetForegroundWindow(windowHandle);
13          this.sendKeyPressesToApp(a.keyPresses);
14    }
15    else
16    {
17          ErrorHandler.raiseError("Could not activate application.");
18    }
19 }
```

```
.....

private void sendKeyPressesToApp(List<String> keyList)
{
   foreach (String key in keyList)
   {
         SendKeys.Send(key);
   }
}
.....
```
**Program listing 2: Starting work units in activities.**

# 7. Evaluation

## 7.1. Data models

The original data model from WfMC that was used as base for the WMS has some elements that were not included in the model used. These are the main elements not included:

- Workflow relevant data
- System and environmental data
- Resource repository

Workflow relevant data is coupled with system and environmental data. It is data that the workflow management system maintains, or data retrieved from the local computer. Such data is not required in the WMS covered in this thesis because all needed data is supplied through the specification of activities or invoked applications.

A resource repository is storage for possible participants in a workflow. In the WMS a user that starts a workflow will be the only participant. No resource repository is therefore needed. According to [10], a resource repository is not required in the model.

Activities have also been more specialized to Aker Kvaerner's and MIPS' needs.

The data model has proved very useful. It provides all entities needed in order to build the WMS for MIPS. The model is also a very good staring point if Aker Kvaerner wants to extend the WMS to its fully potential.

## 7.2. Workflow engine

Working out how to communicate with MIPS has taken a lot of time. It was first discussed if it was somewhat possible to build communication interfaces into MIPS. Everything is possible, but building such interfaces would interfere with other development of MIPS, and would probably have taken a great deal of the time for the WMS project.

Another solution was to manipulate MIPS' database directly. If such a solution should be carried through, one would have to make new views for MIPS directly in the user interface of the WMS. This behaviour is not the intended way this WMS should work. The WMS must use invoked applications as they are. In chapter 4.1 we saw that the WMS should provide wizard-like behaviour for users, and that doesn't work if it is made new views.

Not many options are left, other than sending key presses to the invoked applications. And the fact is that this option turns out to be very effective. It can be used on all kinds of applications, even web applications. This solution was selected because of its effectiveness and its ease of implementation.

## 7.3. User interface

In chapter 4.1 it was stated that the user interface must have easy and intuitive design.
If the user interface is to be easy and intuitive it must:

- Have as few choices as possible

- Navigation must be simple
- Show where a user is in a workflow
- Provide help for each task

Figure 17 in section 5.2.4 shows the chosen user interface design. It was presented at a MIPS user conference and the general feedback from users was good. The interface design also meets most of the good design principles covered in section 4.2.3.

However, some users wanted a minimized solution to satisfy experienced users in order to do tasks more quickly. It was decided to make a solution with navigation buttons to previous and next task only. The minimized solution can be seen in figure 18, section 5.2.4.

# 8. Conclusion

## 8.1. Summary

This thesis has described, evaluated and implemented workflow technology for Aker Kvaerner. General basics for workflow technology were described. Two providers of workflow technology were presented, but only technology from WfMC was evaluated and implemented because of requirements from Aker Kvaerner.

The thesis' main task was to implement a workflow management system. The workflow management system should guide users from task to task in Aker Kvaerner's project application called MIPS.

Solutions for the workflow importer, workflow engine and user interface were described.

## 8.2. Conclusions

Workflow technology is very useful. It provides great support in the field of business processes. Most organizations using this technology have improved their efficiency, making management, staff and customers happier.

The workflow management system described in this system does not take the workflow technology to its full potential. It is not able to share a workflow amongst users or departments.

Although parts of the workflow engine and workflow importer are not programmed, some goals are reached:

- Understandable and intuitive user interface.
- Communicating with MIPS without interfering with its general behaviour.
- Invoke applications included in workflows (At least MIPS).
- Show and hide views inside MIPS as needed in each task in a workflow.
- Wizard-like behaviour.

Goals not reached:

- Import of workflows from XPDL files.
- Running workflows.
- Keeping track of each task in a workflow

The WMS are currently not capable of running workflows. Communication between the user interface and the workflow engine has to be built. If such communication is established, the WMS can run workflows typed directly into its workflow engine.

This thesis has described solutions which will help new users get from task to task in MIPS. The solution described is also capable of using other business applications than MIPS in a workflow if the application in question can be started via command line.

## 8.3. *Further work*

### 8.3.1. Implementation

Two components of the WMS are not finished.

The workflow importer has only been planned. Its task is to read XPDL files and convert the XML-objects to objects in the workflow engine.

Some parts of the workflow engine are missing:

- Communication between workflow engine and user interface
- Keeping track of each task in a workflow

The unfinished components must be finished in order to have a fully functional WMS.

### 8.3.2. Future work

It would be very interesting to develop the WMS further on to fully utilize the workflow technology. The WMS would have to use a server/client based model in order to let users and departments share workflows. Adding inbox-functionality where users can select incoming workflows for their department will also improve the WMS.

It would also be interesting do develop a workflow designer suited for the WMS using XPDL as file format.

### 8.3.3. Problem still not solved

A solution for keeping track of activities in workflows in MIPS has not been found. The WMS is not supposed to interfere with MIPS' behaviour, or to access its database. A solution similar to the way activities are executed can be considered, but it would be extremely complex.

## List of tables and figures

## Bibliography

[1] “Komplett har implementert SAP på 14 uker”, press release, 23.10.03, http://www.sap.com/norway/company/Presse/2003/pressemeldinger/Komplett_SAP.epx

[2] “Bioinformatics workflow management systems”, article, 06.11.07, http://en.wikipedia.org/wiki/Bioinformatics_workflow_management_systems

[3] “Introduction to Workflow”, Charles Plesums, Workflow Handbook 2002, pages 19-38.

[4] “Workflow: An introduction”, Rob Allen, Workflow Handbook 2001, pages 15-38.

[5] “Introduction to the Workflow Management Coalition”, webpage, http://wfmc.org/about/welcome.htm

[6] “Wf-XML 2.0; XML Based Protocol for Run-Time Integration of Process engines”, Keith D. Swenson, Sameer Pradham, Mike D. Gilger, draft, http://www.wfmc.org/standards/docs/WfXML20-200410c.pdf

[7] “How Does XPDL compare to BPEL”, webpage, http://wfmc.org/standards/xpdl.htm

[8] “The Workflow Reference Model”, David Hollingsworth, Document Number TC00-1003, issue 1.1, http://www.wfmc.org/standards/docs/tc003v11.pdf

[9] “Introducing Microsoft Windows Workflow Foundation: An Early Look”, David Chappell, article, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnlong/html/WFIntro.asp

[10] “Process Definition Interface – XML Process Definition Language”, Document Number WFMC-TC-1025, http://wfmc.org/standards/documents/TC-1025_xpdl_2_2005-10-03.pdf

[11] “Principles of good GUI design”, James Hobart, article, http://www.classicsys.com/css06/cfm/article_1995_10.cfm