NTNU
Det skapende universitet

MOWAHS - Evaluering og videreutvikling av MOWAHS rammeverk for mobilt arbeid

Morten Brokerud
John Erik Taskerud Jensen

Problembeskrivelse

MOWAHS rammeverk for mobil arbeidstøtte er et rammeverk for klassifisering av mobile arbeidsoppgaver (f.eks. arbeidsoppgaver utført av utrykningspersonell, militært personell, reisende selgere, service personell etc.) for å etablere krav til teknologi og system for støtte av slike oppgaver. Våren 2003 ble det laget en første versjon av et websystem for å legge inn klassifiseringer og analysere disse.

Denne oppgaven går ut på å utvide systemet. Mulige utvidelser kan være brukegrensesnitt for mobilt utstyr, GUI wizard for karakterisering av systemet, etc. Videre skal det kjøres en brukertest hvor man sammenlikner resultater fra brukere som bruker systemet og brukere som gjør analyse vha. f.eks. regneark. En annen oppgave er å fylle databasen med mobile scenario som gjøres tilgjengelig for omverden.

Oppgaven gitt: 2006-01-20
Hovedveileder: Alf Inge Wang, IDI

MOWAHS – Evaluating and improving the MOWAHS framework for mobile work

Morten Brokerud

John Erik Taskerud Jensen

20 June 2006

NTNU

Abstract

The development of mobile devices advances rapidly every year, PDAs, notebook computers and mobile phones gets more advanced and integrate new technology and functionality. The increased use of mobile technology has lead to new challenges in developing information systems and infrastructures. Some of the challenges in designing software for mobile computing systems are quite different from those involved in the design of software for stationary networked systems.

The MWCF framework (Mobile Work Characterisation Framework) has been developed as a part of the MOWAHS project to provide process support for mobile work. The MWCF framework is designed to help developers find requirements and challenges when designing and implementing mobile systems. In this thesis the MWCF framework is in focus.

Earlier a Web based version of the MWCF system has been developed to increase the usability and availability of the framework. In this thesis the MWCF Web system is improved. These improvements mostly include different issues concerning consistency checks. The consistency checks are a tool to prevent common mistakes when using the MWCF framework.

The MWCF Web system is only designed to run on stationary computers, so a portable version of the system would increase the usability. Thus a PDA version of the MWCF system was developed. The MWCF PDA system is designed to run on portable devices and has options for characterising scenarios and scenario analysis.

A test on the usability of the MWCF framework was also performed, in order to map advantages and possible improvements to the systems and its documentation. The test was carried out by ten persons, five used the MWCF Web system and five used a spreadsheet version of the MWCF system. This test had focus on the characterisation part of the two systems.

Preface

This document has been written as the report on the Master Thesis in Computer Science by John Erik Taskerud Jensen and Morten Brokerud. This thesis has been part of the MOWAHS project which is part of the Software Engineering group at the Norwegian University of Science and Technology (NTNU). The work on the thesis started in January 2006 and lasted until 20 June 2006.

We would like to thank our supervisor Alf Inge Wang for guidance and support during the project.

Trondheim, June 20, 2006

John Erik Taskerud Jensen   Morten Brokerud

Contents

List of figures

List of tables

1. Introduction

This Chapter introduces the settings and the context of the project. A general introduction to mobile work and mobility is given in Chapter 1.1. In Chapter 1.2 the project context is described. The motivation of the project is presented in Chapter 1.3. The project goals for the project are given in Chapter 1.4, the development process is described in Chapter 1.5, the research method is described in Chapter 1.6 and finally a reader's guide is presented in Chapter 1.7.

1.1 General introduction

The development of mobile devices advances rapidly every year, PDAs, notebook computers and mobile phones gets more advanced and integrate new technology and functionality. This means that more and more companies integrate mobile technology in their information systems. This makes a major impact on how employees carry out their daily work, not only can they be more mobile at the office, but they can also use information systems out in the field and work from home. According to the Telework Advisory Group [1] 45.1 million people did some type of work from home in 2005, an increase of about 2% from 2004. This has opened up for a wide range of mobile applications and systems for occupational groups that traditionally have not used information systems to support their work.

The increased use of mobile technology has lead to new challenges in developing information systems and infrastructures. Some of the challenges in designing software for mobile computing systems are quite different from those involved in the design of software for stationary networked systems. Systems have to be flexible, effective, portable and support information sharing over different platforms [2].

Mobile computing is still a quite new field, and lots of research is going on, not only on the technology, but also on user behaviour. An example of a research group that work with mobile computing is the Mobile Wireless Network Research Group at AT&T Labs-Research [3]. They work with issues such as Mobile Interdomain Roaming, Mobile Network Performance, Performance Enhancement Technologies, Mobile Applications and Signal Processing. Research projects like this helps companies, educational facilities and developers of mobile software to a better understanding on how to adapt technologies and resources to support mobility and mobile work.

1.2 Project Context

This project is part of the Mobile Work Across Heterogenous System (MOWAHS) project, which is a basic research project supported by Norwegian Research Council [4]. The people involved in this project are mainly members of IDI's software engineering group and database technology group. The MOWAHS project is divided in two parts: process support for mobile work using heterogeneous devices and support for cooperating transactions/workspaces on databases. The MWCF framework (Mobile Work Characterisation Framework) has been developed as a part of the MOWAHS project to provide process support for mobile work. The MWCF framework is designed to help developers find requirements and challenges when designing and implementing mobile systems. This is done by assigning different weights on software scenarios and then the system uses these weights to calculate different indicators. A Web based version of the MWCF system has been developed to increase the usability and availability of the framework.

1.3 Project Motivation

The MWCF framework was developed to help developers analyse mobile scenarios, and help them to better find and understand the challenges of mobile systems in order to create systems to support such environments. The framework was originally made as a spreadsheet and is described in two articles [5] [6]. These two articles were the only documents describing the MWCF framework, so for people who were not familiar with the framework the documentation was a bit insufficient. The usability and availability of the spreadsheet based framework was not that good. Thus Siv Oma Rogdar developed a Web based system for the framework in 2003, as part of her Master's Thesis "A Web System for Characterising Mobil Work" [7]. This system serves as a user-friendly tool for applying the MWCF framework on mobile scenarios, and for retrieving the complexity indicators for the scenarios.

With the MWCF Web system the user can create different mobile scenarios and then identify different roles in the scenario, further tasks for each role are identified. For each task, the user writes a task description using the task template. Then the user assigns task weights, and uses the system to characterise the tasks. By combining the scores of the characteristics several indicators can be computed. Theses indicators are used to analyse the mobile scenario. The framework is further described in Chapter 2.

The only lack of functionality in the Web based system Siv Oma Rogdar was the lack of data consistency checks and PDA support. Therefore we decided to implement the consistency rules in the MWCF Web system in our in depth study [9]. The consistency checks are made by applying a set of rules on different software developing scenarios. These rules are made to prevent common user mistakes during the assignments of scores to the scenario tasks. An example on this is when the user has decided that a task must be executed at a specific time. The system will then suggest that this task also has to be planned for. The user can chose to ignore this suggestion, but that is not recommended. By applying these rules on the MWCF system, the user will be helped during the assignments of scores on the different tasks.

When you are developing software for mobile work it is not unlikely that you have to do some of the work out in the field. If a developer could analyse different scenarios straight a way, it would simplify the developers work a lot. Since the MWCF framework only exists as a stationary system, we decided to start planning a PDA version for the framework in our in depth study [9]. This PDA version will be implemented as a part of this Masters Thesis.

Whether the MWCF Web system has better usability than the spreadsheet version of the system or not has never been thoroughly tested. The usability of the consistency checks has never been tested on regular users. A test of the usability of the MWCF Web system versus the MWCF spreadsheet version will therefore be carried out as part of this Masters Thesis.

1.4 Project Goals

This project is separated in three different goals. In this Chapter, we will describe the goals, and explain how we plan to achieve them.

Goal 1: Improving the MWCF Web system

In our in depth study [9] we implemented the consistency rules in the MWCF Web system. This include "Add consistency rule" and "Remove consistency rule" options in the Administer Framework part of the system, and a system to check characterisations for consistency issues in "Characterise Scenario". Some functionality is still lacking in the MWCF Web system, this include a "View consistency rule" option in Administer Framework and a better way to change characterisations according to consistency issues. These parts of the MWCF Web system will be updated.

To achieve this goal we have to implement the "View consistency rule" functionality and design a page for changing characterisations according to consistency rules to improve the usability of the system.

Goal 2: Implement a PDA version for the MWCF framework

When designing systems for mobile work much of the work has to be carried out in the field. The MWCF Web system is only designed to run on stationary computers, so a portable version would increase the usability of the system. We started planning and designing a portable MWCF system for PDAs in our in depth study [9]. Our goal is to finish the design and to implement this system.

To achieve this goal we have to redesign the MWCF Web system to fit PDAs. This includes reducing information and designing smart solutions to make the most of the limited resources on PDAs.

Goal 3: Test MWCF Web system versus MWCF spreadsheet system

One of the reasons the MWCF Web system was developed, was to improve usability of the MWCF framework and to implement consistency rule checking. Whether the Web system has improved the usability or not has not been thoroughly tested, thus we will perform a test on the usability of the Web system versus the spreadsheet version. This test will have focus on the characterisation part of the two systems. By performing this test the benefits of the consistency rules will also be examined.

To achieve this goal we have to design a scenario to use in the test. We will use a set of ten persons to perform the tests. They will be divided in two groups, one group will characterise the scenario using the MWCF Web system, and the other group will characterise the scenario using the MWCF spreadsheet version. After the characterisation each participator will answer a set of questions concerning usability and possible problems they had during the characterisation.

1.5 Development process

There exist many different approaches to software development and design. These different models are designed to structure the development process, and each is designed to fit different scenarios, different work methods and different settings. When we started on our in depth study [9] we had little experience with both the existing MWCF system and the technologies used to develop it. Our approach to this was to learn different aspects of the system during the development, and to adjust our requirements after we had developed a new part of the system. A development model that is designed for this type of work process is the incremental model. We discovered that this approach suited us well, so we decided to use the same model for the MWCF PDA system.

The incremental model (IM) is designed, implemented and tested as a series of incremental builds until the product is finished, as illustrated in Figure 1. A build consists of pieces of code from various modules that interact together to provide a specific function. At each stage of the IM a new build is coded and then integrated into the structure, which is tested as a whole. The product is only defined as finished when it satisfies all of its requirements. This model combines the elements of the waterfall model with the iterative philosophy of prototyping. However, unlike prototyping the IM focuses on the delivery of an operational product at the end of each increment.

Some of the advantages of the incremental model are that the system can be tested and used at an early stage. The most important parts of the system can be developed first if that is wanted. The requirement specification can also be updated during the development, to better approach the demands of the users.
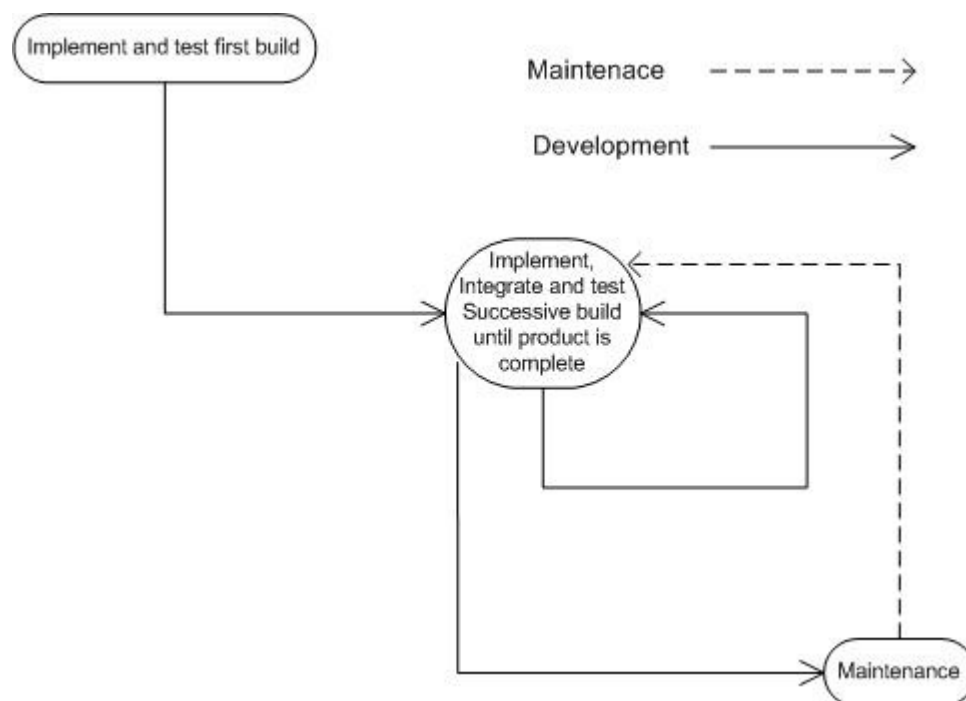
Figure 1: The incremental model

The advantages of selecting the incremental model where that we could test the different aspects of the MWCF Web and PDA systems and their technologies while developing it. We also updated our design and requirements as we got more familiar with the system and its technologies.

Software testing is an important part of the development process, and many different approaches exist. The test methods used in this project are unit and module testing. A unit test is a procedure to verify that a particular module of source code is working properly. Ideally each test case is separate from the others. Module testing consists of testing a collection of depended components, such as an object class or a collection of functions. The testing was carried out during and after developing the different components in the system.

1.6 Research method

Research methods enable software engineers to examine the benefits of new technologies, tools, methods and so on. In this thesis the engineering method described by Victor R. Basili [24] is used.

The engineering method: Observe existing solutions, propose better solutions, build/develop, measure and analyse, and repeat the process until no more improvements appear possible.

The engineering method was followed through one and a half cycle. First the framework and the existing systems was examined. The consistency validation and other improvements were implemented and tested. The results from this test lead to a set of possible improvements. These improvements are not implemented into the system and framework, but are described and discussed in the thesis.

1.7 Reader's guide

In Part I the Prestudy part of the thesis is introduced. This includes a MWCF introduction, technologies used in the system and an introduction to PDAs and Smartphones. Part II includes requirements, design and implementation for the improvements made in the MWCF Web system. In Part III requirements, design, implementation, testing and a user guide for the MWCF PDA system are presented. Part IV contains an evaluation of the MWCF system; this includes an introduction to the test of the MWCF Web system versus MWCF spreadsheet system, a presentation of the results, analysis and a conclusion. Part V contains an evaluation of the thesis and finally Part VI contains the appendixes.

Part I – Prestudy

2. Mobile Work Characterisation Framework (MWCF)

This Chapter contains prestudy for the MWCF Web system and the MWCF PDA system. The Chapter is based on our work done in our in depth study [9]. In Chapter 2.1 the Mobile Work Characterisation Framwork is described, Chapter 2.2 introduces the current practice and in Chapter 2.3 the consistency rules is specified.

2.1 The Framework

The MWCF is a framework for characterising mobile work. It was developed by the researchers in the MOWAHS research project. By using the MWCF framework on a mobile work scenario, one can elicit both functional and non-functional requirements. The framework consists of four main characterisation groups: general, information, location and time. Each group has five characteristics which can be given a value from 1 to 5 where 5 indicate a high complexity. Some characteristics only use the values 1, 3 and 5. All the characteristics of the MWCF are shown in Table 1.

To apply the framework to a mobile work scenario, one uses the following steps.

1. Select the mobile scenario and identify the different roles/actors.
2. For each role identify tasks.
3. For each task:
      a) Write a task description using a task description template
      b) Assign task priority (1-5, where 5 is most important)
      c) Characterise the task using the framework (see Table 1).
      d) Calculate the requirement indicators for the task
4. Derive system requirements and priorities from the characteristics and indicators.

Table 1: Characteristics

Characteristic	Possible values	Description
General		
G1 Decomposable	(1 No, 3 Uncertain, 5 Yes)	Decides whether the task is composed of subtasks or not.
G2 Part of sequence	(1 No, 3 Partial, 5 Yes)	Specifies if a task has order dependencies with respect to other tasks.
G3 Pre-planned	(1 Planned, 3 Partial, 5 Adhoc)	Describes to what degree a task is planned in beforehand.
G4 Data synchronisation	(1 No, 3 After task, 5 Within task)	Specifies when a task has to synchronize / merge updated data with other tasks.
G5 Data exchange rate	(1 None, 3 Once, 5 Many)	Specifies the rate of data exchange between the current task and other tasks (during its lifetime).
Information		
I1 Information contents	(1 NA, 2 Text, 3 Graphics, 4 Audio, 5 Video)	Describes the complexity of the information required or produced by the task.
I2 Information streaming	(1 NA, 3 Discrete, 5 Continuous)	Describes whether the task requires streaming of data or not.
I3 Information required related to time	(1 NA, 2 Low, 3 Medium, 4 High, 5 Real-time)	Describes how important the information required to execute the task is related to time.
I4 Information produced related to time	(1 NA, 2 Low, 3 Medium, 4 High, 5 Real-time)	Describes how important the information produced by the task is related to time.
I5 Information transmission speed	(1 NA, 2 Slow, 3 Medium, 4 Fast, 5 Very fast)	Describing the expected transmission speed of information used or produced by the task.

Location		
L1 Location dependent	(1 No, 3 Partial, 5 Yes)	Describes to what degree a task must be executed at a specific location (e.g. that a lecturer must be in a classroom to teach students).
L2 Require services at location	(1 No, 3 Partial, 5 Yes)	Specifies if a task needs electronic services available at the location such as printers, network connections, video projectors, fax machines, etc.
L3 Produce services at location	(1 No, 3 Partial, 5 Yes)	Specifies if a task produces electronic services at the location that can be used by others at this location, such as information queries, beaming of information, network connectivity etc.
L4 Location report	(1 No, 3 Partial, 5 Yes)	Specifies if a task must report its location to the system.
L5 Route constraints	(1 No, 3 Partial, 5 Yes)	Specifies if a mobile task must follow a specific route or not when moving around.
Time		
T1 Event-triggered	(1 No, 3 Partial, 5 Yes)	Decides whether a task is triggered by an event or not
T2 Time constraint	(1 No, 3 Partial, 5 Yes)	Describes if a task must be executed at a specific time or within a specific time-span (e.g. a task must be performed between 10:15 and 12:00).
T3 Temporal coordination	(1 No, 3 Partial, 5 Yes)	Describes if a task must be coordinated with other tasks (e.g. different military units must strike at the same time).
T4 Task resumption	(1 No, 3 Partial, 5 Yes)	Describes if a task can halt, and then later resume from where it left off (not requiring a complete restart of the task).
T5 Task lifetime	(1 Seconds, 2 Minutes, 3 Hours, 4 Days, 5 Weeks)	Describes the expected lifetime of a task.

From the scores of these twenty characteristics, indicators can be computed. These indicators will help in the analysis of the scenario and in the extraction of requirements. The MWCF has ten different indicators as listed below:

General Task Indicator (GTI) is an average of G1-G5. A high GTI score indicates that the underlying process and transaction infrastructure (e.g. workflow system) must be advanced.

Information Complexity Indicator (ICI) is an average of I1-I5. A high ICI score indicates that the end-system must cope with complex information presentation, management and transmission. The ICI can also be used to select the appropriate hardware and software to be used as a mobile client and server.

Location Complexity Indicator (LCI) is an average of L1-L5. A high LCI score can indicate that the end-system must be location-aware, include Geographic Information System (GIS) functionality, and use a mobile client suitable for mobility in terms of weight, size, battery power etc.

Time Complexity Indicator (TCI) is an average of T1-T5. A high TCI score indicates that time management and coordination of tasks, and advanced transaction support might be necessary. In addition, the TCI also indicates the level of performance and availability required.

Network Connectivity Indicator (NCI) is an average of G3, G4, G5, L4, and T1 and indicates the level of connectivity between the mobile client and the server. It determines the

required networking capabilities of the mobile client. Further, it indicates non-functional requirements for the system such as reliability and latency.

Network Speed Indicator (NSI) is an average of I3-I5. A high NSI score means that the transmission speed and quality of service must be high between the mobile client and supporting servers. The NSI also indicates what wireless network technology can be used for the end-system.

Energy Consumption Indicator (ECI) is an average of I5, L1, L2, L3, and T5. A high ECI score means that it is likely that the mobile client device can complete the task will consume much energy.

Transaction Support Indicator (TSI) is an average of G3, G4, G5, T1, T4, and T5. TSI describes the need for flexible/advanced transactional support. A high TSI score indicates that the transactional support must go beyond ACID transactions.

Mobility Indicator (MI) is an average L4 and L5, and indicates how much mobility is involved. The MI is useful for determining the complexity of the environment the mobile client will operate in, e.g. variation in wireless networks. A high MI will affect the choice of equipment (device) and tools necessary to accomplish the task.

Task Complexity (TC) is an average of all characteristics of a task and indicates the complexity of one task. The TC is useful for finding the most complex task that should have the most attention in a further examination.
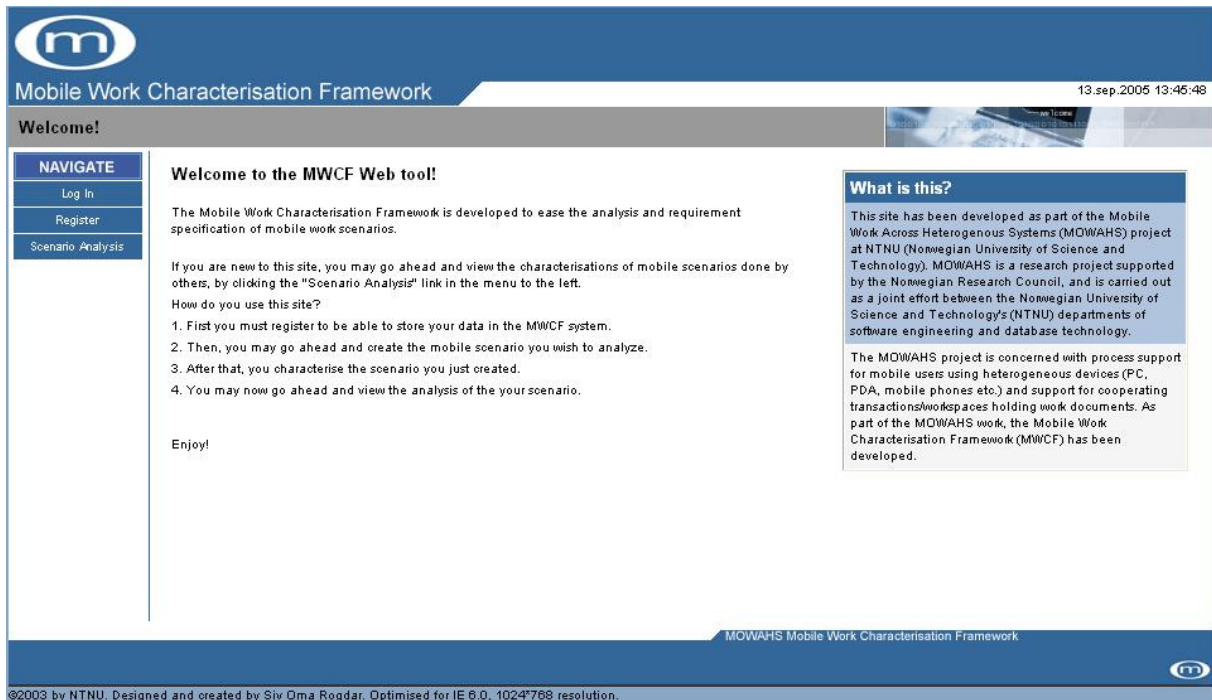
2.2 Current practice

The MWCF framework was originally a spreadsheet described in two articles [5] [6]. The spreadsheet was used to characterise mobile scenarios and to compute the mobile complexity indicators of the scenarios. One of the problems with the system was that the system itself and the information on how to use it were separated. The learning curve for users that was not well familiar with the system was too steep, so a Web based system was desired.

So far, two other students have worked on the development of the Web based MWCF system. Siv Oma Rogdar designed and implemented the system in 2003. This is a fully functional system, the only part of the system she did not implement was the consistency rules, but the design and description of these rules are written in detail in her Master's Thesis [7]. Jakub Hajduk continued working on the system in 2004. His goals were to implement the consistency rules and to plan and implement a PDA version of the system. These tasks were never fully implemented, but the design and planning exists in his Master's Thesis [8]. In 2005 we extended the MWCF system by implementing the consistency rules and started working on a plan for a PDA version on the system. Further details on this can be found in our in depth study [9].

The Web based MWCF system can be used by both registered users and non-registered users. A non-registered user can choose to view all the scenarios stored in the system, or to create an account by the register option. A registered user can create new scenarios, edit scenarios, delete scenarios and characterise scenarios. When a new scenario is created, the user adds tasks to the scenario. After a scenario is characterised, the user can see the calculated

indicators of the scenario. Registered users may also retrieve his/hers password by mail if it is forgotten, and edit their current account settings.

Figure 2: The MWCF site

2.3 Consistency rules

A set of consistency rules has been developed to help the user in the process of characterising different mobile scenarios. These rules are related to the relations between the different characteristics in the framework. The rules prevent common user mistakes during the assignments of scores to the scenario tasks in the MWCF Web system. If a user enters data in the system that is not consistent, the user will get a warning and a suggestion on how to correct the data. The user can choose to ignore these suggestions since projects vary greatly and the consistency rules may not apply on all projects.

The following rules are derived from the set of characteristics, descriptions and value scales in Table 1. The numbers of the characteristics as found in Table 1 are combined with scores to form a consistency rule. The logical terms used in the rules are as follows:

Table 2: Symbols

Symbol	Meaning
→	Implication
≥	Greater or equal
V	Logical OR
=	Equal

Rule 1

$I3 \geq 4 \lor I4 \geq 4 \rightarrow I5 \geq 4$

Involves characteristics:
• I3 - Information required related to time
• I4 - Information produced related to time
• I5 - Information transmission speed

If there is a high score on either (or both) of the characteristics I3 or I4, the information required to execute the task related to time or the information produced by the task related to time is highly important. This implies that there is a need for high transmission speed of information used or produced by the task, which means that I5 should have a high value.

Rule 2

$L5 = 5 \rightarrow L1 = 5$

Involves characteristics:
• L1 - Location dependent
• L5 - Route constraints

If a task needs to follow a specific route when moving around (score 5 (yes) on L5), it is dependent on being at specific locations to execute the task, which implies a value of 5 (yes) on L1.

Rule 3

$L4 = 5 \rightarrow L1 = 5$

Involves characteristics:
• L1 - Location dependent
• L4 - Location report

A task that has a score of 5 on L4, which means that the task must report its location to the system, implies that the task is dependent on being at a specific location (since that location is of importance to the system).

Rule 4

$T3 = 5 \rightarrow G2 = 5$

Involves characteristics:
• G2 - Part of sequence
• T3 - Temporal coordination

If a task must be coordinated with other tasks with respect to time (score 5 (yes) on T3), a task is part of a sequence of tasks (score 5 on G2).

Rule 5

$I2 = 5 \rightarrow G5 = 5$

Involves characteristics:
• G5 - Data exchange rate
• I2 - Information streaming

If a task requires streaming (score 5 on I2), it requires a high rate of data exchange between the current task and other tasks during its lifetime (score 5 - Many on G5).

Rule 6

$G3 = 5 \rightarrow T1 = 5$

Involves characteristics:
• G3 - Pre-planned
• T1 - Event-triggered

If a task is Ad-hoc (score 5 on G3), then it is event-triggered (score 5 on T1). Ad-hoc tasks are activities that cannot be planned for and typically occurs as a reaction to some event.

Rule 7

$T2 = 5 \rightarrow G3 = 1$

Involves characteristics:
• G3 - Pre-planned
• T2 - Time constraint

If a task has time constraints and must be executed at a specific time (score 5 on T2), it must be planned for (score 1 on G3).

Rule 8

$G2 = 5 \rightarrow G3 = 1$

Involves characteristics:
• G2 - Part of sequence
• G3 - Pre-planned

If a task is part of a sequence (score 5 on G2), it has to be planned for in advance (score 1 on G3).

Rule 9

$L5 = 5 \rightarrow G3 = 1$

Involves characteristics:
• L5 - Route constraint
• G3 - Pre-planned

If a task has a route constraint (score 5 on L5), it has to be planned for in advance (score 1 on G3).

Rule 10

$T4 = 5 \rightarrow T5 \geq 2$

Involves characteristics:
• T4 - Task resumption
• T5 - Task lifetime

If a task can halt and later resume from where it left off (score 5 on T4), it must have a lifetime that can at least be measured in minutes (score 2 or more on T5).

3. Technologies used in the MWCF Web and PDA system

In this Chapter the technologies used in the MWCF system will be explained. This includes the different Java API's, the Struts Platform, the Apache Jakarta Tomcat Web server and the MySQL database system. Each technology is explained thoroughly, and illustrated with figures, so readers that are not familiar with them easily can understand the different technologies used in the MWCF system.

3.1 Java 2 Software Development Kit (Java 2 SDK)

With most programming languages you either compile or interpret the program to make it run on your computer, the Java Programming language does both. Your program is compiled to an intermediate language called Java bytecodes before it is interpreted to run on your computer. Compilation happens once, interpretation happens every time you run the program. This makes the Java Language highly portable because the Java bytecodes can run on any machine as long as it has Java Virtual Machine (Interpreter) installed.
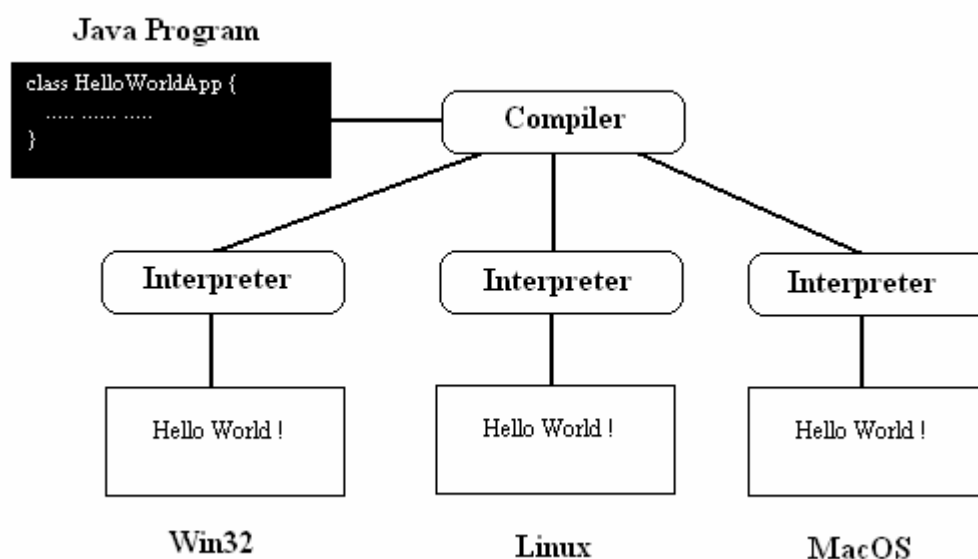
Figure 3: A Java program running on different machines

The Java technology is both a platform and a programming language. It is a software-only platform that runs on top of hardware based platforms. The platform consists of two elements, the Java Virtual Machine and the Java Application Programming Interface (Java API). The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets [10]. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages.
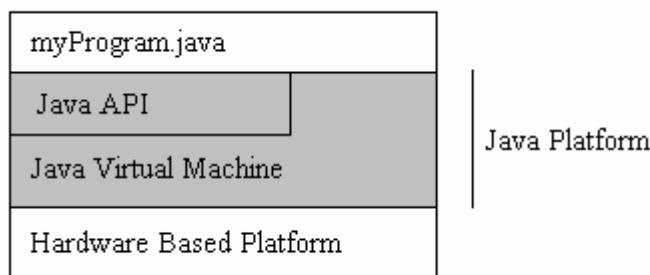
Figure 4: A program running on the Java platform

3.2 JavaBeans

JavaBeans technology is the component architecture for the Java 2 Platform. The JavaBeans was originally developed by Sun to work with GUI components, but they are now used with every aspect of Java, including Web applications [10]. JavaBeans are platform-independent self-contained reusable software units. Any java object that conforms to certain rules can be a bean. JavaBeans specification contains the following definition of a Java Bean: "A Java Bean is a reusable software component that can be manipulated visually in a builder tool" [22]. A reusable software component is a piece of software that is written once and can be reused several times. This enables developers to write reusable components once and run them anywhere. This also gives the opportunity to sell and buy software components. When software systems are developed, it can be timesaving to reuse certain components. These components can range from simple Web based components to software components used in advanced systems.

The definition of a JavaBean states that it is a software component that can be manipulated visually in a builder tool. Builder tools or application builders provide the environment to use the beans and manipulate them [11]. A builder tool is able to connect different JavaBeans without writing any code. This means that the developer can drag and drop components onto a form. A form can be a Java panel, window or canvas. Examples of builder tools are Borland JBuilder, Sun Java Workshop and JCreator. Figure 5 shows a screenshot from JBuilder. The Figure shows that it is possible to drag and drop components, buttons and text fields in this example, on to a Form. Components can then easily be modified with the object inspector.
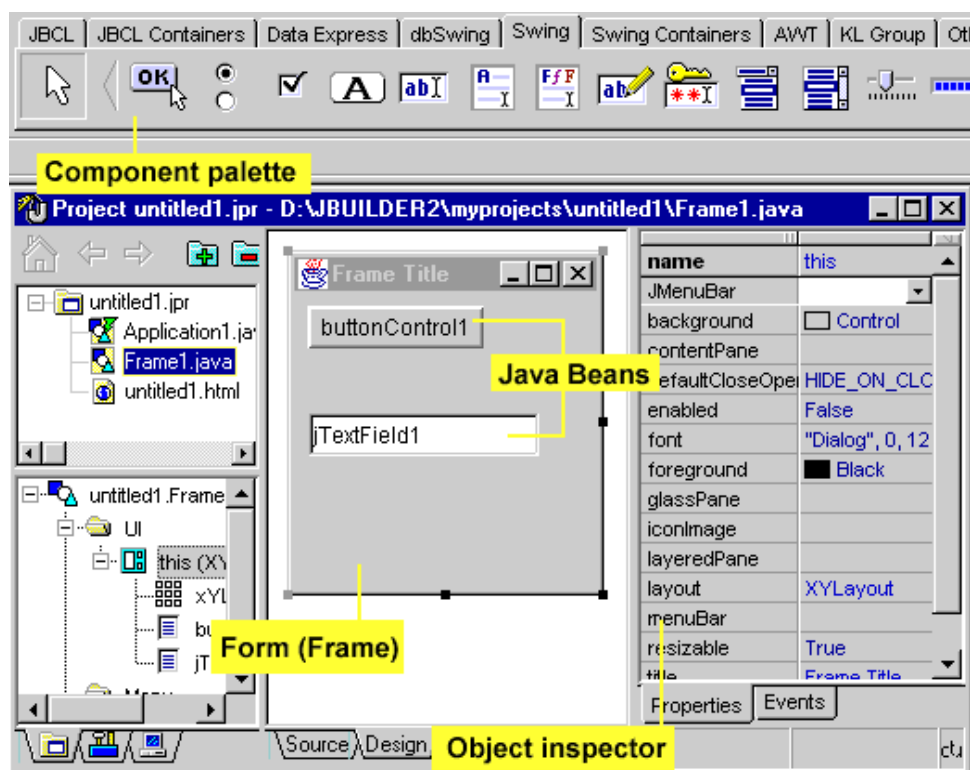
Figure 5: JBuilder 2, an example of a builder tool

3.3 Java Servlets

Servlets are modules of Java code that run in a server application to answer client requests. Servlets are the server side counterpart to *applets* (which are used only on clients). They are Java application components which are downloaded, on demand, to the part of the system which needs them [10]. Servlets provide a general framework for services using the request-response paradigm. This is illustrated in Figure 6 where clients are talking to Java Servlets in servers. Clients can be anything from a simple HTML form to a sophisticated Java applet.
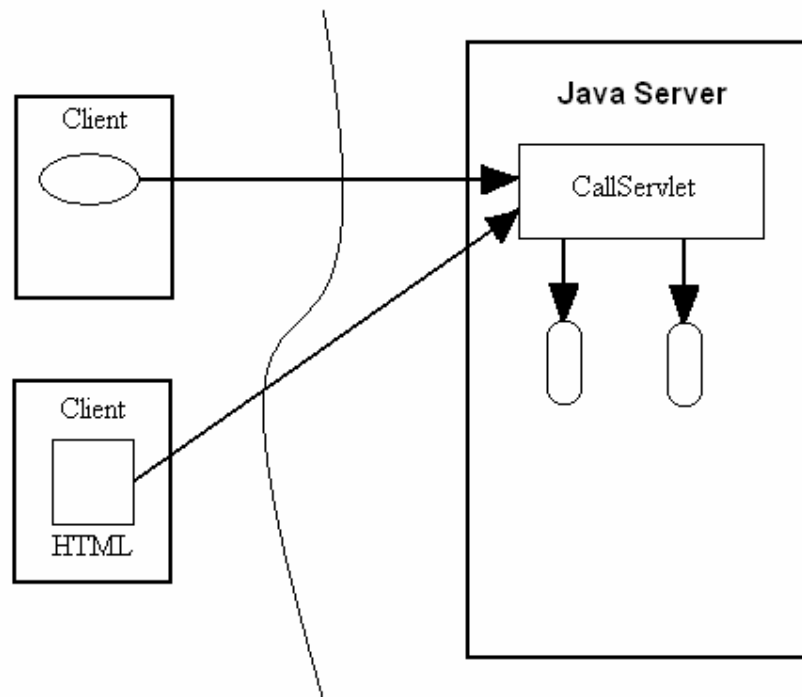
Figure 6: Clients talking to java Servlets in servers

Servlets can be found at several different sources, this is illustrated in Figure 7. They can be found in individually administered locations (a), this can be a remote server or a local file. They can be located in the directory of Servlets (b) on the server, or they can be uploaded from authorized clients.
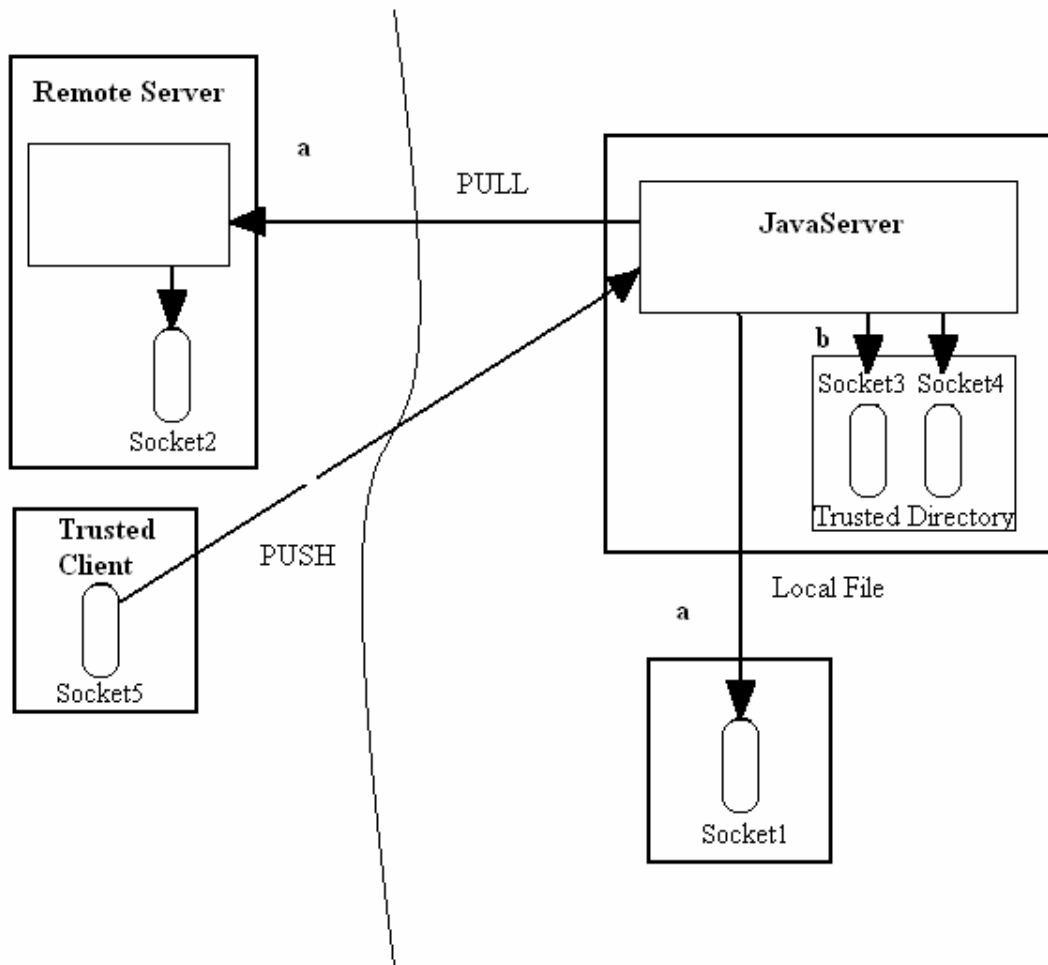
Figure 7: Different Servlet sources

Servlets make use of the Java standard extension classes in the packages "javax.servlet", for basic Servlet framework, and "javax.servlet.http" for Servlets that answer HTTP requests. Typical uses for HTTP Servlets include processing and storing data submitted by an HTML form, returning results from database queries to the client and maintaining and managing state information. As all other Java technologies Java Servlets are not dependant on the server or the platform.

Following is a simple example of a Java Servlet where the Servlet sends text to the client. The Servlet writes "Hello WWW" on the Web page:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWWW extends HttpServlet {
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
      throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0
" +

"Transitional//EN\">\n" +
              "<HTML>\n" +
              "<HEAD><TITLE>Hello WWW</TITLE></HEAD>\n" +
              "<BODY>\n" +
              "<H1>Hello WWW</H1>\n" +
              "</BODY></HTML>");
  }
```

3.4 JavaServer Pages (JSP)

JavaServer Pages is a technology for developing Web pages that displays dynamically-generated content. JSP pages use XML tags and scriptlets written in Java to encapsulate the logic that generates the content for the page. It passes any formatting tags directly back to the response page. In this way, JSP pages separate the page logic from its design and display. JSP pages are compiled into Servlets, and through the Java APIs the Servlets can access different object such as JavaBeans components. A JSP page has the extension .jsp or .jspx, this tells the server that the JSP engine will process elements on this page [10].

There are two different approaches to building applications using JSP technology [12]. These two approaches are described in the JSP Model 1 (Figure 8) and Model 2 architecture (Figure 9). In the Model 1 architecture, the JSP page alone is responsible for processing incoming requests and replying to the client. All data access are done using beans, this separates the presentation from the content. The Model 1 architecture is well suited for simple application, but it may not be so good for advanced applications with many requests and a high number of scriptles. A scriptlet is a reusable HTML or script element. It enables an HTML or script fragment to be downloaded once, maintained in a cache and referenced over and over by different HTML pages and scripts.
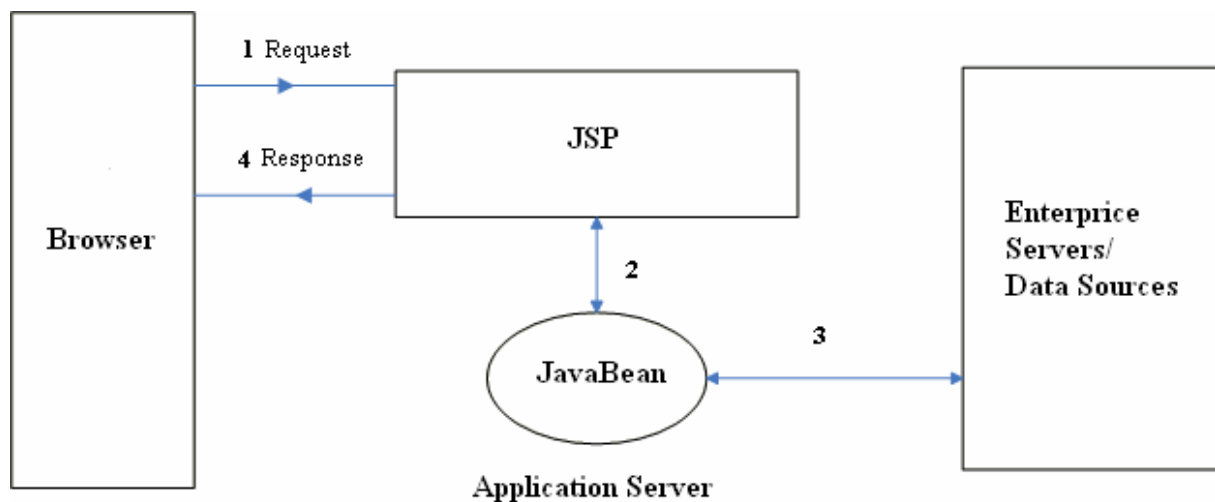
Figure 8: JSP Model 1 architecture

The JSP Model 2 architecture combines both the uses of Servlets and JSP. It uses the JSP pages to generate the presentation layer and servers to perform process-intensive tasks. It is the Servlets that is in charge of processing the requests and the creation of beans or objects used by the JSP pages, and it also decide which JSP pages to forward the requests to. The JSP does no processing; it is only responsible for retrieving any objects or beans created by the Servlet, and for extracting dynamic contents from the Servlets. With this architecture the presentation and content are well separated, and the benefits of using the Model 2 architecture increases when complexity of the application is higher.
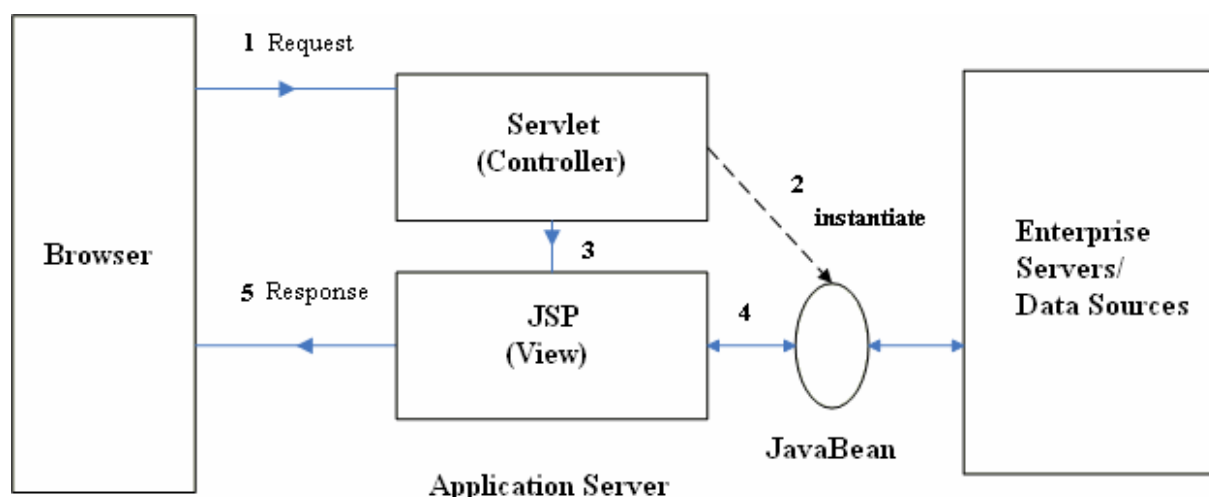
Figure 9: JSP Model 2 architecture

An example of a JSP page that writes the date and time on the Web page:

```
<HTML>
<HEAD>
<TITLE>JSP Example</TITLE>
</HEAD>
<BODY BGCOLOR="ffffcc">
<CENTER>
<H2>Date and Time</H2>
<!-- Java code start -->
<%
java.util.Date today = new java.util.Date();
out.println("Today's date is: "+today);
%>
<!-- Java code end -->
</CENTER>
</BODY>
</HTML>
```

3.5 Apache Jakarta Tomcat

Apache Tomcat is the Servlet container that is used in the official Reference Implementation for the Java Servlet and Java Server Pages (JSP) technologies [13]. At the highest level, Tomcat is composed of two pieces. The first is the Catalina Servlet container. This engine provides the primary container for Servlets. The second is the Jasper JSP container. This means that Java Servlets can be used within it to enable programmers to use the Servlet in such things as Web pages. Whereas a JSP Web page has both HTML and Java JSP code within the same document, a Servlet is a block of code that can be reused [14]. These can then be compiled once and distributed to run on any standards compliant Servlet Container. The MWCF system uses Tomcat as a JSP/Servlet container and as a stand-alone Web server.
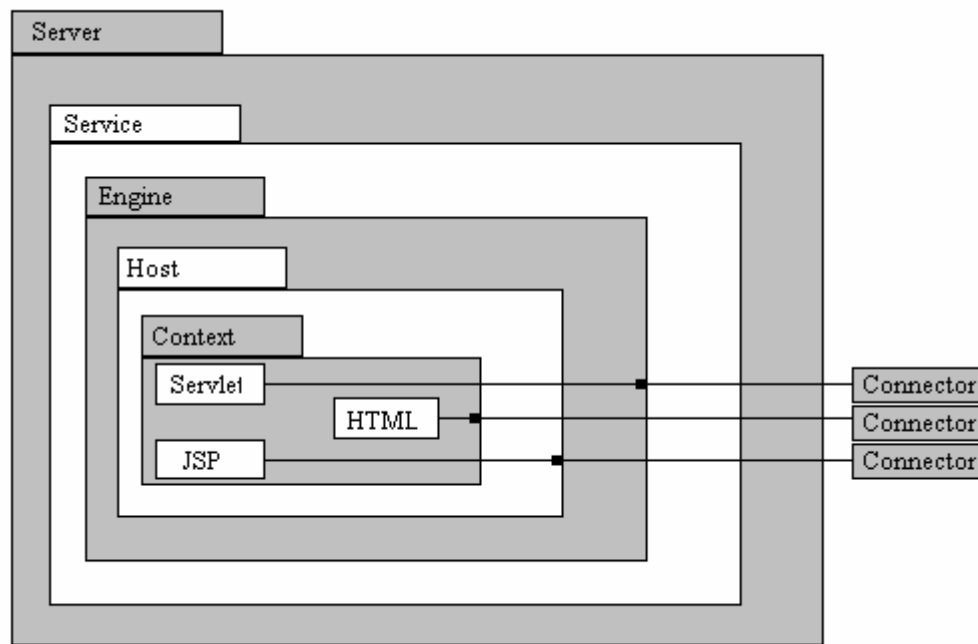
Figure 10: Key components and connectors of Tomcat

Figure 10 illustrates the main components and connections of Tomcat, further described they are:

Server: Represents the whole container.
Service: An intermediate component which lives inside a Server and ties one or more connectors to exactly one Engine.
Engine: Represents request processing pipeline for a specific Service.
Host: An association of a network name, e.g. www.yourcompany.com, to the Tomcat server.
Connector: Handles communications with the client.
Context: Represents a Web application.

3.6 Struts

Struts is an open source framework for building Web applications and is a project of the The Apache Software Foundation [15]. The core of the Struts framework is a flexible control layer based on standard technologies like Java Servlets, JavaBeans, ResourceBundles, and XML, as well as various Jakarta Commons packages. Struts enables Web application development which follow the Model-View-Controller (MVC) design.

- Model - This is the data model which is usually developed by the end user of the Struts framework. This is typically the application's business logic and code to interact with a database.

- View - In a generic sense, the View is comprised of the components which make up the application's user interface. Struts utilizes JSPs to serve as the view. Struts includes a set of custom tag libraries that facilitate creating user interfaces that interacts with ActionForm beans. ActionForms capture and validate whatever input is required by the application.
- Controller - This is the program control logic which accepts user input and decides how to respond. The response is generated via the View components. In Struts, the primary component of the Controller is a servlet of class ActionServlet. This servlet is configured by defining a set of ActionMappings. The ActionMappings defines which Action a user request will be routed to. Actions encapsulate calls to business logic classes, interpret the outcome, and ultimately dispatch control to the appropriate View component to create the response.

3.7 MySQL

A database is a structured collection of data. To be able to add access and process data stored in a database you need a database management system. The MWCF Web system uses MySQL for this purpose. The MySQL database is an open source database which can freely be used for non-commercial purposes. The MySQL server contains a series of useful properties [16]:

- Multi-threading support for use on multiple CPUs.
- Multiple user access
- Support for APIs for several programming languages
- High memory integrity

MySQL also has full support for Structured Query Language (SQL). SQL is an ANSI (American National Standards Institute) standard computer language for accessing and manipulating database systems. SQL statements are used to retrieve and update data in a database. There are many different versions of the SQL language, but to be in compliance with the ANSI standard, they must support the same major keywords in a similar manner (such as SELECT, UPDATE, DELETE, INSERT, WHERE, and others)

The MWCF Web system uses the JDBC API to communicate with the MySQL database [17]. JDBC is a Java API that contains a set of classes and interfaces designed to support database access. JDBC supports several database technologies, including MySQL. Figure 11 shows how the JDBC architecture is structured.
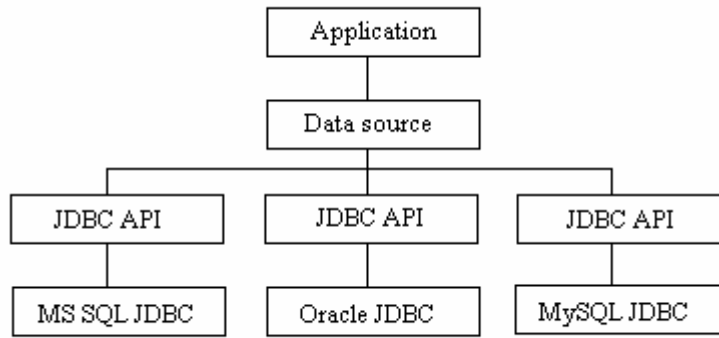
Figure 11: The JDBC architecture

4. Personal Digital Assistant (PDA) and Smartphone

This Chapter describes technologies and functionality found in PDAs and Smartphones. In Chapter 4.3 we discuss which of these two technologies is best to use in a mobile version of the MWCF system.

4.1 Personal Digital Assistant (PDA)

A PDA is a small hand-held device that provides computing and information storage for personal or business use. The main purpose of a PDA is to act as a personal planner and organizer, and to store information. PDAs are normally of small sizes so they easily can fit in a pocket. There are many manufacturers of PDA's, but all share some basic features:

- **Date book:** The most important part of the PDA. It provides organization of meetings, appointments, etc.
- **Memo pad:** This feature makes it possible to make notes or other messages. Can also be used to make to-do lists.
- **Calculator:** An integrated calculator.
- **Address book:** Keeps track of addresses, phone numbers, email addresses, etc.
- **Email:** Compose and send email on HotSync or wireless.
- **Onscreen keyboard:** Uses the touch screen for input. This makes it possible to use handwriting instead of typing.
- **Synchronisation with PC:** This gives the opportunity to synchronise PDA data with PC data.

PDA's as we think of them today was first made popular by PalmPilot. Over the years many companies have entered the PDA marked. Today there are 2 major marked leaders, Palm devices that run the Palm OS and Microsoft Pocket PC that run Windows mobile [18]. Most Palm OS PDAs are made by palmOne and their most standard features are:

- A vast library of third-party applications (more than 20,000) that you can add to the system (most devices come bundled with e-mail, productivity, and multimedia software)
- An updated version of the Graffiti handwriting-recognition application
- Synchronization with both Windows and Macintosh computers using the Palm Desktop
- Smaller displays than Pocket PCs to accommodate a dedicated Graffiti area on the device (Some higher-end Palm devices now incorporate a virtual Graffiti area in the display, resulting in a larger display area.)

Windows mobile PDAs, known as pocket PCs, typical features are:

- Pocket versions of Microsoft applications such as Microsoft Word, Excel, and Outlook
- Synchronization with Microsoft Outlook on a Windows PC (synchronization with e-mail systems other than Outlook or with Macintosh computers requires additional software)
- Three handwriting-recognition applications: Transcriber, Letter Recognizer (similar to the new version of Graffiti), and Block Recognizer (similar to the original Graffiti)
- A virtual writing area, which maximizes the display size
- Windows Media Player for multimedia content

PDAs are getting more and more advanced, and it can be hard to put them in a single category. Many PDAs come with features from mobile phones, such as sms, mms and standard phone features. PDAs are also starting to come with other features we usually know from other devises such as integrated mp3 players, cameras and GPS receivers. A good example of this is the Garmin iQue 3600 Palm OS based PDA which come with mp3 player, GPS receiver, voice recognition and all the standard PDA features. A picture of iQue 3600 is showed in Figure 12.
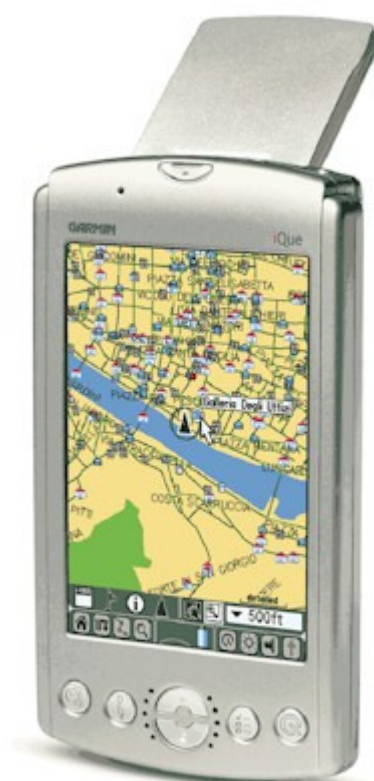

Figure 12: The Garmin iQue 3600 PDA

4.2 Smartphone

Most phones nowadays have a calendar, a few games, and downloadable applications. But some phones go further to bring voice and data needs together, including syncing e-mail, surfing the Web, shooting pictures and supporting hundreds of third-party applications. Some

of these phones have full keyboards to easily write e-mail or use other applications. This crossing between mobile phones and PDAs is called Smartphones [19]. A Smartphone is not necessary a phone with PDA capabilities, it can also be a PDA with phone capabilities.

Smartphone features include e-mail and personal information storage, including phone numbers, contacts, addresses, calendar, and to-do lists. There exist hundreds of applications for Smartphones; you can perform business tasks, like reading PDFs and Excel spreadsheets. Many of these phones can also be used for entertainment such as 3D games, music and record and play back video.

Most Smartphones come with a preinstalled OS that can host third-party applications. There are several available OSs, some of the most popular are:

- **BlackBerry OS**: This is an e-mail focused OS. It is very easy to read and send e-mail on any BlackBerry. But only a few third-party applications are available to enhance the OS. This OS is very popular in USA.
- **Microsoft Windows Mobile for Smartphone**: MS Windows mobile allows you to synchronize the phone with PCs, and offers many well known MS features such as an e-mail application similar with outlook.
- **Palm OS**: Palm OS is best known for its easy to use personal information manager, the Graffiti handwriting-recognition and for its vast range of applications.
- **Symbian**: Symbian is the most popular Smartphone OS in Europe. It has similar functionality as the Palm OS, and uses less memory and battery power than its competitors. Symbian supports the latest Java Community Process standards.

Smartphones are still in their infancy, but their popularity is increasing. According to analyst house Canalys Smartphone shipments increased over 100% from 2004Q2 to 2005Q2, with over twelve million devices shipped in the latter period. In a couple years, it is likely that most phones sold will be considered "smart", except for disposable phones [20].

Figure 13: Nokia 3620, Smartphone

4.3 Which technology to use

One of the biggest challenges of making the MWCF system mobile is the small displays of mobile devices. This means that only the most necessary information should be displayed. The device should also have wireless network support. Another challenge is to limit the input from the user, since input on mobile devices is more time consuming than on regular desktop computers. These to factors are very important when choosing which technology to use. Following are an overview of pros and cons of the different technologies.

Smartphones

Pros:	Cons:
Combines PDA and mobile phone functionality in a mobile phone	Limited text-entry capability
Plays videos and music files-a very portable multimedia-device	Smaller, harder to read screens than most PDAs
Strong support from third-party software developers, especially game/entertainment developers	Does not have the same set of built-in applications found on PDAs, word processing and spreadsheets are missing
Variety of accessories, including navigation systems, Bluetooth an IR	Inputting large amounts of data can be tedious
Good battery capacity, better than laptops	Limited storage capacity
Portable	
Network access anywhere	

PDAs

Pros:	Cons:
Have a better text-entry capability than most Smartphones, including handwriting-recognition	Do not have phone functionality
Strong support from third-party software developers, more office applications than Smartphones	Have a bigger screen than most Smartphones, but still the screen size is limited
Media Players for multimedia content	Inputting large amounts of data can be tedious
Variety of accessories, including navigation systems, Bluetooth, IR and WLAN	Limited storage capacity, but better than Smartphones
Good battery capacity, better than laptops	Limited network access, dependent on WLAN
Portable	

As shown in these pros and cons PDAs and Smartphones have many similarities. They have good battery capacity, support a wide range of software and Technologies and are highly portable. The biggest different between PDAs and Smartphones are their display sizes and their input methods. Most Smartphones have smaller displays and a more limited input technology than PDAs. This applies to Smartphones that are phone based with PDA features. PDAs with phone features on the other hand have the same display sizes and input methods as other PDAs. This means that the mobile MWCF system will be best suited for PDAs, to give

the best GUI as possible. PDAs also have a better text-entry capability, which is favourable because the mobile MWCF system will require some text-entry. The biggest down side of designing the MWCF system for PDAs is the PDAs limited network access. The PDA is dependent on WLAN to access the MWCF system. If the system is to be used in an area without WLAN access the user has to use a PDA based Smartphone with phone features to access the MWCF system.

Part II – MWCF Web system

5. Requirements and design

This Chapter contains the requirements and design for the new functionality added to the MWCF Web system. In Chapter 5.1 the functional requirements are specified, in Chapter 5.2 the non-functional requirements are described and in Chapter 5.3 the design is specified. The requirements and design for the full system can be found in Siv Oma Rogdar's Master's Thesis [7] and in our in depth study [9].

5.1 Functional requirements

This Chapter contains the functional requirements for the new functionality added to the MWCF Web system. They are presented as both graphical and textual use cases.

Before describing the use cases involved in the Web system improvements, the actors of the MWCF system are defined. The actors and their relationships are illustrated in Figure 14 The user actor, who is any user which is not registered in the system, is at the top level, while the registered user is a specialisation of the user, denoted by the *generalisation* relationship. The administrator is at the next level as a specialisation of the registered user. The actors are described as follows:

User The user represents an anonymous person that is not registered in the MWCF system. This person has only access to view mobile scenarios previously stored in the system and cannot enter data into the system.

Registered User This actor represents a person who wants to use the MWCF to create and characterise mobile scenarios. This person is registered in the system and can access the MWCF system with a user name and a password. The registered user has restricted access to the system. He may view all scenarios stored in the system (like the user actor), and store scenarios and characterize these but only edit scenarios he himself has created previously.

Administrator The administrator represents a person who is responsible for maintaining the MWCF system. The administrator has access rights to every part of the system and administrator rights to view and edit all data in the system.
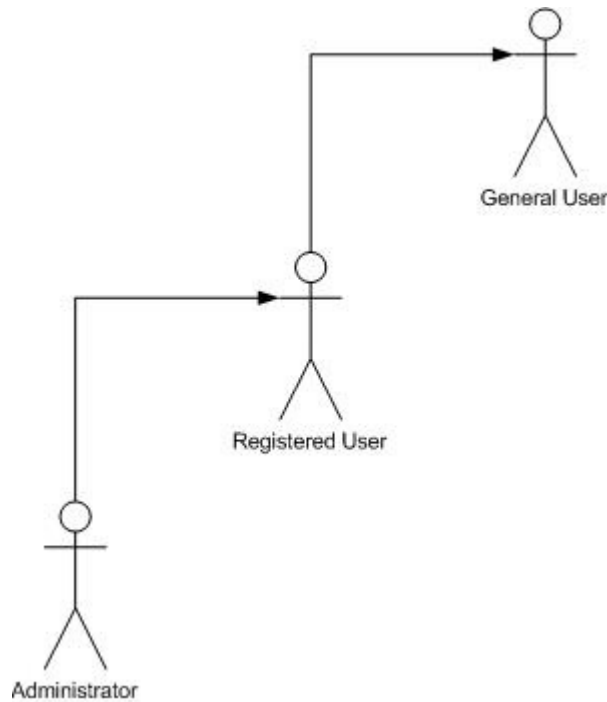
Figure 14: The actors of the system and their relationship

.

Use case 7: Characterise scenario
A use case diagram for use case "Characterise scenario" is depicted in Figure 15.

Brief description After having created a scenario, a characteriser may characterise an existing scenario by assigning each task in the scenario a value for every characteristic in the framework. The user can validate the characterisation according to the consistency rules.

Actor Registered User

Preconditions The user is logged in and has previously created a scenario in the MWCF system, which the user now wants to characterise.

Path of flow The main flow of events should be:
1. `include (Login)`. The user chooses the scenario to characterise.
2. The next steps are repeated for each task in the scenario that the user wants to characterise at this point:
(a) The user chooses a task to characterise.
(b) The user assigns the task a weight between 1 and 5, specifying the importance of the task.
(c) The user assigns the task a value on every characteristic in the framework.
(d) The user saves the characterisation done to the task.
(e) The user can validate the characterisation. If the user has assigned values that are conflicting, i.e. not in accordance with the rules for data consistency (see Chapter 2.3), the user will be made aware of what the conflicts are and may choose to revise the characterisation or leave it as it is.

3. The user finishes the characterisation and may view the analysis and interpretations for this scenario if desirable.

Postcondition The characterisation is stored in the system, and the calculated complexity indicators may be viewed.
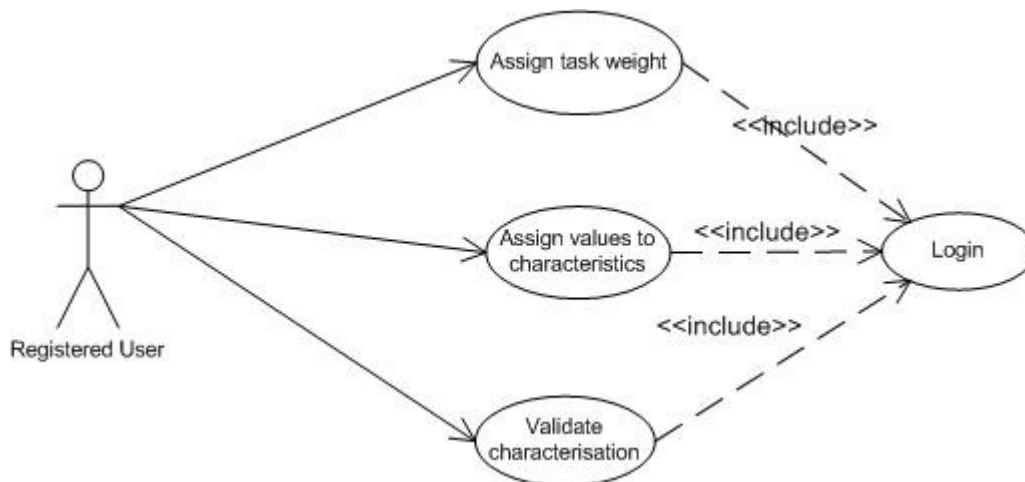

Figure 15: Use case diagram for characterise scenario

5.2 Non-functional requirements

This Chapter describes the non-functional requirements to the system. The requirements are taken from our in depth study [9] and have been edited to fit the context of this project.

Lifespan
The system shall be ready for delivery on the 20th of June 2006. The system should be available for use by interested system developers and students by the 20th of June 2006.

Interfaces to Web
All of the features of the system must be available for use by a Web browser.

Screen resolution
The Web system will be optimised for personal computer screen resolutions of 1024 * 768.

Usability
The Web system for the characterisation framework will be easy to use in the sense that the user will be guided through the characterisation process. The user will also be guided through the creation of a scenario. The user will be notified if he has entered any contradictory data when characterising a scenario. For this, the rules for inconsistencies of Chapter 2.3 will be used.

Modifiability of framework
It should be possible to make changes in the framework without changing the system.

Extensibility of framework
In addition to allowing for modifications of the existing components of the framework, the system should allow for more fundamental changes, such as extensions to the framework by e.g. adding new components to the framework. When adding new components no changes

should be needed on existing system functionality. The GUI will not take any considerations to this fact, but the system as a whole should be made such that this is possible.

5.3 Design

In this Chapter the sequence diagrams of the part of use case 7 "Characterise scenario" where the user can change the characterisation according to the validation are presented. The Sequence Diagram models the collaboration of objects based on a time sequence. It shows how the objects interact with others in a particular scenario of a use case [21].
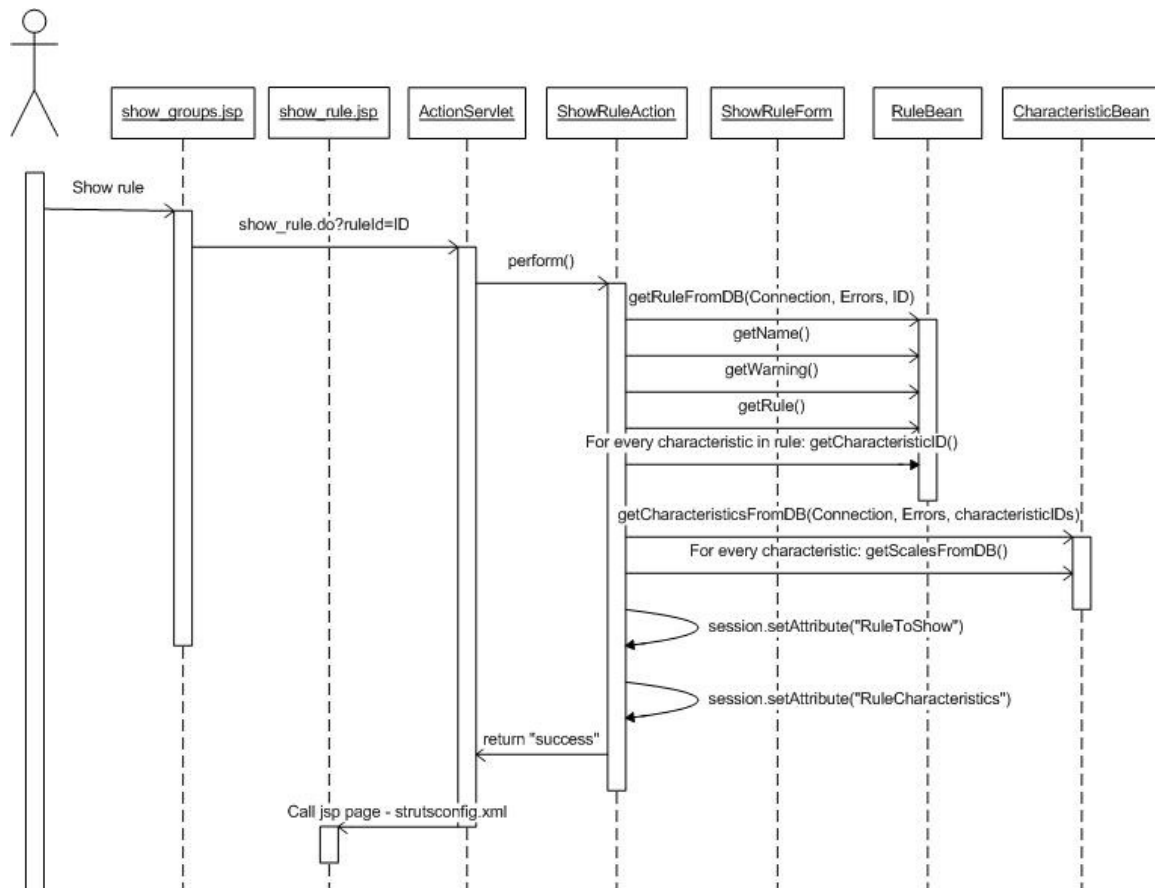

Figure 16: Sequence diagram for retrieving the characteristics of a rule

Use Case 7 Characterise scenario
When a user has validated his characterisation and there is a conflict he gets the option to edit the characterisations relevant to the rule that has been broken. Figure 16 shows the sequence of events when the user clicks the link to edit the characterisation. The "ShowRuleAction" gets the current rule and the characteristics of the rule and their scales from the database. This information is then shown to the user. Figure 17 shows what happens when the user presses the save button. First the characteristics relevant to the rule and the task id are retrieved, and then the new values, selected by the user, are stored in the database.
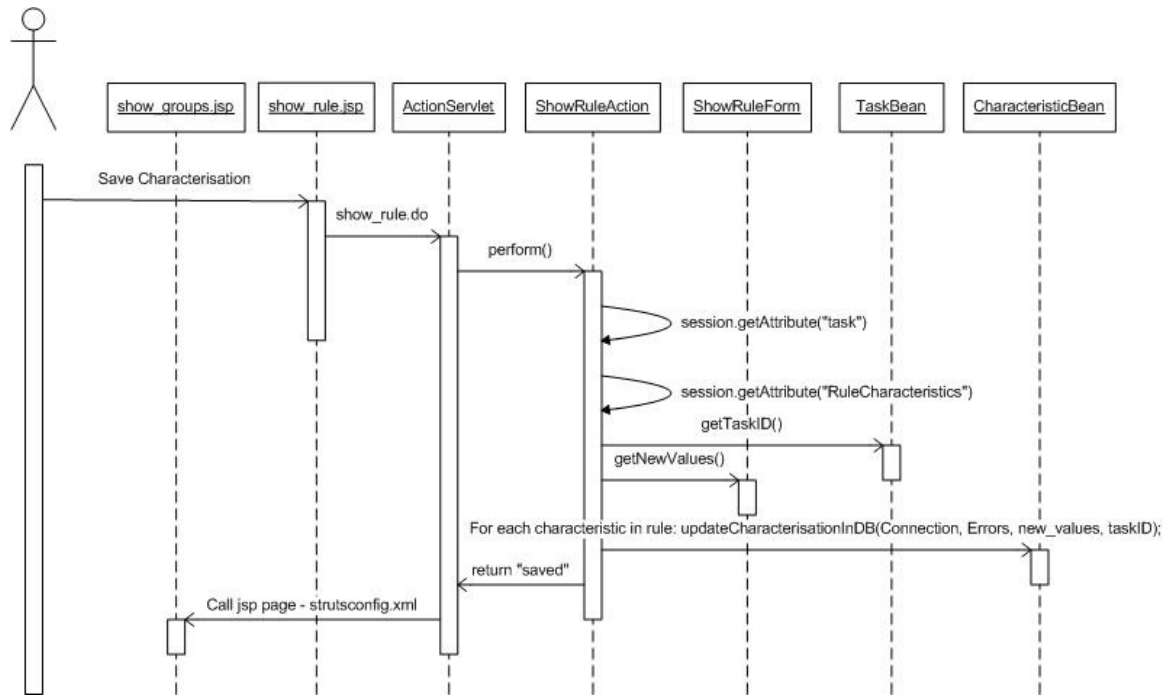
Figure 17: Sequence diagram for updating the characterisations in a rule

6. Implementation

This Chapter contains the details about the implementation of the new functionality added to the MWCF System. It also contains information about other minor changes to the system. These changes and additions have been made both to prevent errors and to improve the system.

6.1 Validate Characteristic

New functionality has been added to make it easier for a user to change his characterisation according to the rules in the framework. When a rule is broken the user will now be able to enter a new page showing information about the rule and the characteristics involved, the user can also change his characterisation of the involved characteristics on this page.

6.2 Other improvements on the MWCF Web system

- Administrators can now view the details of a rule from the "Administer Framework" pages.
- A support page with mail contact information and a "Frequently asked question" section has been added to the system. The content of the "Frequently asked question" section is limited, but that will probably improve as the system becomes available for others.
- When a user wants to edit his characterisation on a task, the system will now select the values stored in the database as default instead of selecting the first value in the scale.
- The page containing validation and listing of rules has been changed to make it look better.
- The syntax of the .css files has been changed to make the system viewable in a broader range of Web browsers.
- Some of the descriptive texts throughout the system have been changed and typing errors have been fixed.
- The method for retrieving scenarios from the database has been changed to prevent scenarios from being listed multiple times when the tasks of the scenario had different responsible roles. Scenarios will now only be listed in "Scenario analysis" if it is fully characterised.
- On pages available to both registered and unregistered users the "log out" link is no longer shown when unregistered users browse these pages.
- When moving the system to a server located at NTNU a major database connection problem was discovered. The system would work fine for a day, but the next day every operation that used the database connection would cause SQL exceptions. The cause of this was that the system did not recover if its database connection stalled and that the MYSQL server removed the connection after eight hours idle time. To solve this problem a connection pool was implemented, the system now checks if a connection is valid each time before it is used and creates a new connection if needed. To implement this, the tomcat server needed to be reconfigured and all the action classes retrieving connections needed to be changed. This caused a lot of extra work but it was necessary to make the system useable.

Part III – MWCF PDA system

7. Requirements and design

This Chapter describes the requirements and design for the MWCF PDA system. In Chapter 7.1 the functional requirements are specified, in Chapter 7.2 the non-functional requirements are described and in Chapter 7.3 the design is specified.

7.1 Functional requirements

This Chapter contains the functional requirements for the implementation of the MWCF PDA system. They are presented by graphical and textual use cases.

Before describing the use cases of the system, the actors of the MWCF PDA system are defined. The actors and their relationships are illustrated in Figure 18 The general user actor, who is any user which is not registered in the system, is at the top level, while the registered user is a specialisation of the user, denoted by the *generalisation* relationship. The actors are described as follows:

General User The user represents an anonymous person that is not registered in the MWCF system. This person has only access to view mobile scenarios previously stored in the system and cannot enter data into the system.

Registered User This actor represents a person who wants to use the MWCF to create and characterise mobile scenarios. This person is registered in the system and can access the MWCF system with a user name and a password. The characteriser has restricted access to the system. He may view all scenarios stored in the system (like the user actor), and store scenarios and characterise these but only edit scenarios he himself has created previously.
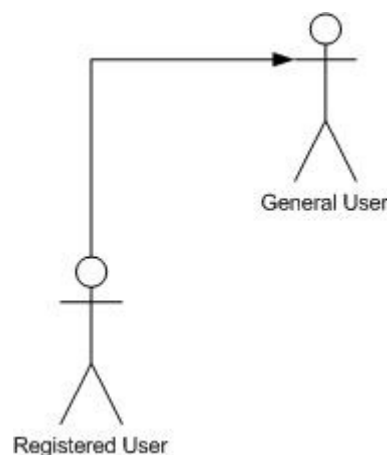

Figure 18: The actors of the system and their relationship

The main use cases (as shown in the use case diagram in Figure 19) in the system are:

UC1 Login
    A characteriser will be able to log in to the system using an e-mail address and a private password.

UC2 Characterise Scenario

> After having created a scenario, a characteriser may characterise an existing scenario by assigning each task in the scenario a value for every characteristic in the framework.

UC3 View Analysis and Interpretations

> Any user may view the characterisation done on a scenario stored in the system and its complexity indicators.

UC4 Logout

> A user of the system that is logged in should log out when the user has finished using the MWCF PDA system. This is to prevent that other persons use the access rights of the person logged in. In case a user forgets to log out, the user should be automatically logged out of the system when the user has been inactive for 1 hour.

Some of the main use cases are further divided into new use cases. These use cases are not shown in the main use case diagram (Figure 19) to keep the diagram clear and easily understandable, but are shown in the use case diagrams of the respective use cases. The main use case diagram shows the two types of actors involved in the system: the general user and the registered. As shown also in Figure 19, the relationship between the actors is of type *generalisation*. This means that a characteriser inherits the behaviour of the user actor, which means e.g. that a registered user also has an interface to the use case "View analysis and interpretations".
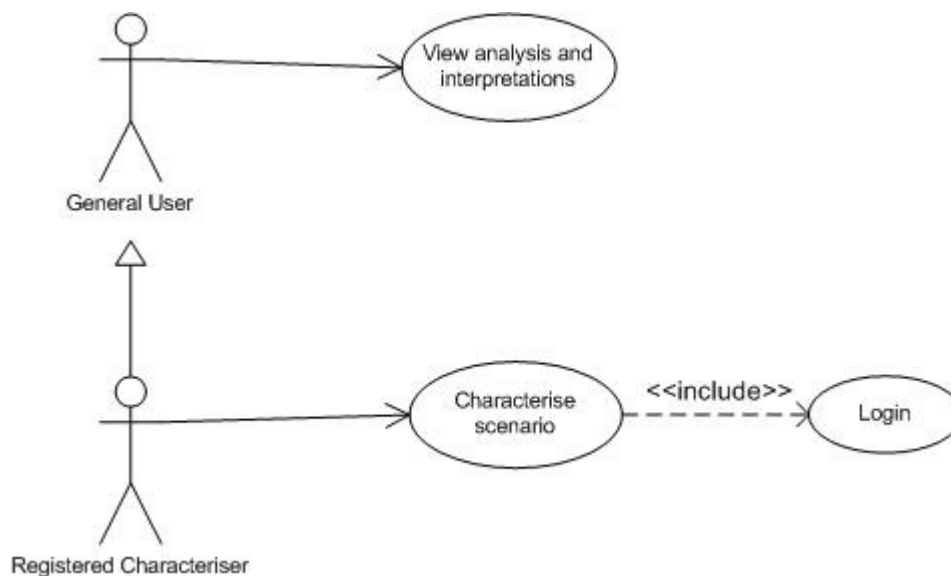
Figure 19: Main use case diagram

Use case 1: Login
Use case "Login" is depicted in Figure 20.

Brief description A registered user will be able to log in to the system using an e-mail address and a private password.

Actor Registered user

Preconditions The user has a characteriser account and has browsed to the login page.

Path of flow The main flow of events should be:
1. The user enters his e-mail address and password and clicks the Login button.
2. The user is logged into the PDA system and this state is kept by the system until the user logs out, the user has been inactive for a one hour or all open browser windows are closed.

Alternative path 1 The user has forgotten the private password, and requests the password from the system. The password is sent to the characteriser's e-mail address.

Alternative path 2 The user is not logged into the system. The user is explained what the problem was (wrong password or e-mail address) and what actions to take to solve the problem.

Postcondition The use case instance ends when either the user is successfully logged into the PDA system or the user chooses to take some other action.

Figure 20: Use case diagram for login

Use case 2: Characterise scenario
A use case diagram for use case "Characterise scenario" is depicted in Figure 21.

Brief description After having created a scenario, a characteriser may characterise an existing scenario by assigning each task in the scenario a value for every characteristic in the framework. The user can validate the characterisation according to the consistency rules.

Actor Registered User

Preconditions The user is logged in and has previously created a scenario in the MWCF system, which the user now wants to characterise.
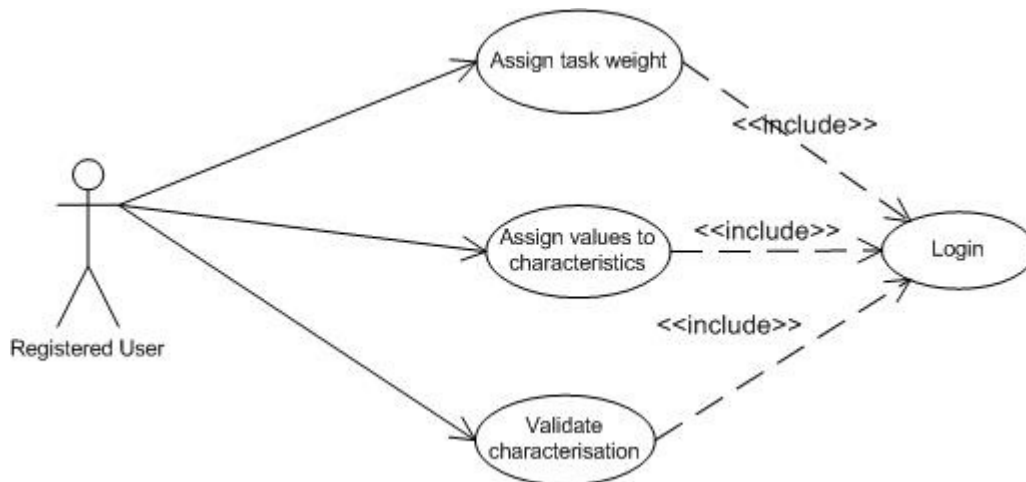
Path of flow The main flow of events should be:
1. Include (Login). The user chooses the scenario to characterise.
2. The next steps are repeated for each task in the scenario that the user wants to characterise at this point:
(a) The user chooses a task to characterise.
(b) The user assigns the task a weight between 1 and 5, specifying the importance of the task.
(c) The user assigns the task a value on every characteristic in the framework.
(d) The user saves the characterisation done to the task.

(e) The user can validate the characterisation. If the user has assigned values that are conflicting, i.e. not in accordance with the rules for data consistency (see Chapter 2.3), the user will be made aware of what the conflicts are and may choose to revise the characterisation or leave it as it is.

3. The user finishes the characterisation and may view the analysis and interpretations for this scenario if desirable.

Postcondition The characterisation is stored in the system, and the calculated complexity indicators may be viewed.

Figure 21: Use case diagram for characterise scenario

Use Case 3: View analysis and interpretations
A use case diagram for use case "View analysis and interpretations" is depicted in Figure 22

Brief Description Any user may view the characterisations done on a scenario stored in the system and view its complexity indicators.

Actor General User.

Preconditions The user has browsed to the PDA pages of the MWCF system and wants to view the characterisations of scenarios in the system.

Path of flow The main flow of events should be:
1. The user chooses a scenario.
2. The user is shown the characterisation of the scenario and the calculated complexity indicators.

Post conditions The user is able to view the analysis and interpretations of any scenario stored in the system.
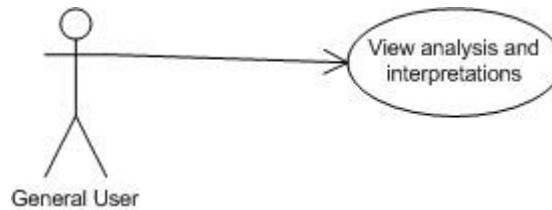
Figure 22: Use case diagram for View analysis and interpretations

Use case 4: Logout
A use case diagram for use case "Logout" is depicted in Figure 23.

Brief description A user of the system that is logged in should log out when the user has finished using the MWCF PDA system. This is to prevent that other persons use the access rights of the person logged in. In case a user forgets to log out, the user should be automatically logged out of the system when the user has been inactive for 1 hour.

Actor Registered user

Preconditions The user is logged in to the system.

Path of flow The main flow of events should be:
1. The user logs out of the system.
2. The user may not use any of the features of the system except to view the analysis and interpretations of the scenarios stored in the system without logging in again first.

Alternative path 1 The user is inactive for 1 hour and is automatically logged out of the system.

Postcondition The user is successfully logged out of the system and must log in to use the access rights of its profile.
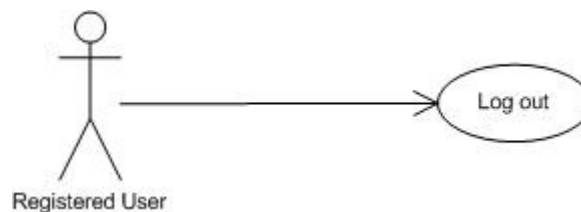

Figure 23: Use case diagram for logout

7.2 Non-functional requirements
This Chapter describes the non-functional requirements to the system. The requirements are based on the MWCF Web system requirements. The implementation of the MWCF PDA system shall follow these requirements.

Lifespan
The system shall be ready for delivery on the 20th of June 2006. The system should be available for use by interested system developers and students by the 20th of June 2006.

Interfaces to Web
All of the features of the system must be available for use by a Web browser.

Screen resolution
The Web system will be optimized for PDAs with screen resolutions of 240*320 pixels.

Usability
The PDA system for the characterisation framework will be easy to use in the sense that the user will be guided through the characterisation process. First time users should be able to use all functionality in the system without further information than what it provided by the system. The user should be notified if he has entered any contradictory data when characterising a scenario. For this, the rules for inconsistencies of Chapter 2.3 will be used.

Modifiability of framework
It should be possible to make changes in the framework without changing the system.

Extensibility of framework
In addition to allowing for modifications of the existing components of the framework, the system should allow for more fundamental changes, such as extensions to the framework by e.g. adding new components to the framework. When adding new components no changes should be needed on existing system functionality. The GUI will not take any considerations to this fact, but the system as a whole should be made such that this is possible.

7.3 Design

The MWCF Web system is implemented using the Model-View-Controller (MVC) design pattern included in the Struts framework. This model contains 3 main parts. The Model contains the code to interact with the database. The View is comprised of the components which make up the application's user interface. The MWCF system use JSP to serve as the view. The Controller is the program control logic which accepts user input and decides how to respond. A further description of the MVC pattern can be found in Chapter 3.6.

The MWCF PDA system use the same Model-Controller as the MWCF Web system, only the View is different. This means that the sequence diagrams and the database design for the MWCF PDA system is identical to the ones for the MWCF Web system, with the exception of the names of the JSP files. All the JSP files in the PDA version have a "pda_" prefix; i.e. "index.jsp" is named "pda_index.jsp". Therefore the sequence diagrams and the database design is not listed in this paper. All sequence diagrams and database tables can be found in Siv Oma Rogdar's Master's Thesis [7] and in our in depth study [9].

8. Implementation

This Chapter describes the implementation of the MWCF PDA system. Chapter 8.1 describes the different technologies relevant to the system and Chapter 8.2 describes the different aspects of the implementation.

8.1 Technologies

This Chapter contains a short summary of the technologies relevant to the system

Technologies used for the implementation
The following technologies have been used to implement the new functionality to the MWCF PDA System.

- Java Development Kit
  Java 2 SDK, version 1.4.2_06 has been used during the implementation.
- JDBC Database Access
  The JDBC 2.1 API has been used for implementing access to the database.
- Jakarta Struts Framework
  The Struts 1.0.2 framework has been used for implementation.
- Database
  The MySQL database server, version 4.1 has been used to store data in the MWCF System

All coding has been done in JCreator 3.50.

Technologies used for running the MWCF PDA System
The following software has been used for running the MWCF PDA System.

- Servlet/JSP and Web Container
  Tomcat 4.1 has been used both as the Servlet/JSP container and as a stand-alone Web server.
- Internet browser
  Microsoft Internet Explorer, version 6.0 and MS Windows Mobile 2003 SE version 4.21 has been used when testing the system.

8.2 The MWCF PDA System

The MWCF PDA system is built up in the same way as the MWCF Web system; each page in the MWCF PDA system has a corresponding page in the MWCF Web system where the functionality on each page is the same in both systems. Since the PDA system is optimized for a screen size of 240*320 pixels the general layout of the pages for the PDA system (Figure 24) needed to be different from the layout in the Web system (Figure 25). All the pages in the PDA system where developed to prevent horizontal scrolling as much as possible and to avoid excessively long vertical scrolling. In the general layout of the pages the menu was moved to the top of the page and the top and bottom logos was minimized. To make room for the vertical scroll bar the width of the PDA pages is only 220 pixels. Generally the pages in the system have a reduced amount of descriptive and informational texts.
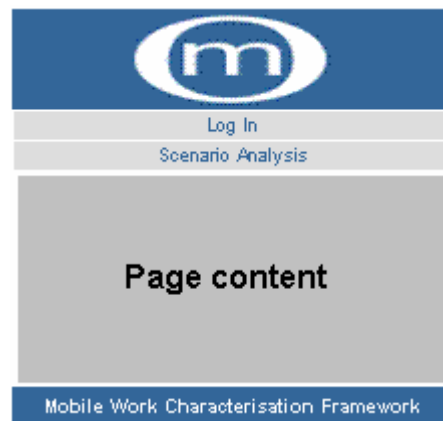
Figure 24: General layout PDA
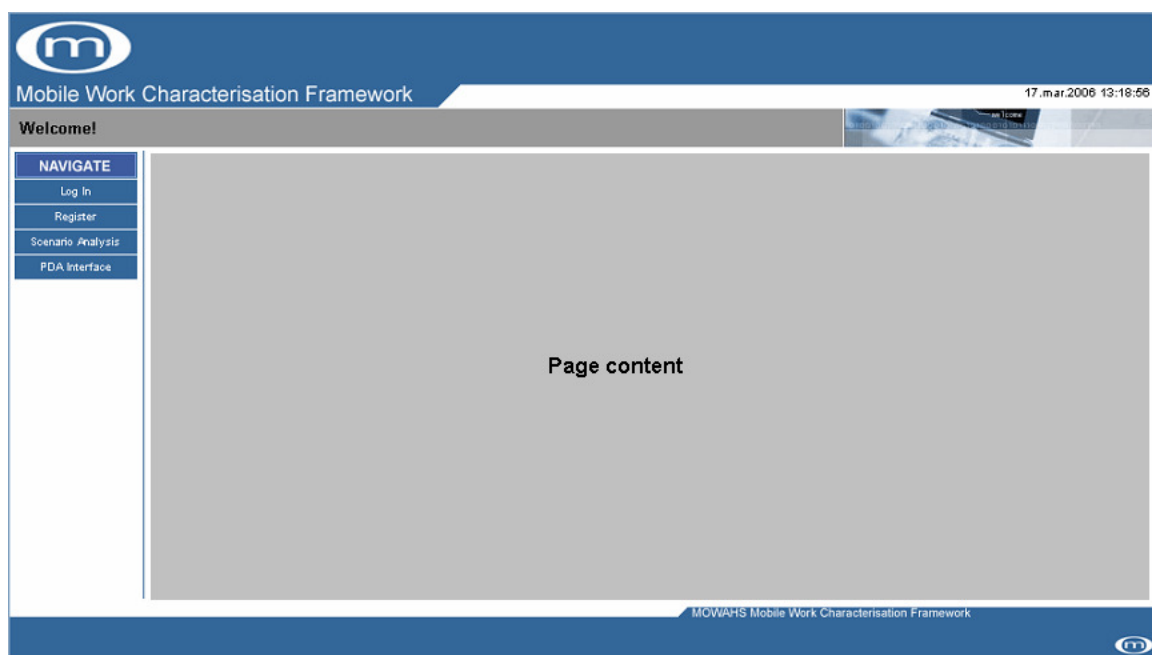
Figure 25: General layout Web

Some of the pages to be displayed in the PDA system contained so much information that it was impossible to show it all at once. In some cases information could be dropped without the page loosing its value so that the content of the page would fit the small screen. On the pages shown in Figure 26 and 27 all the information in the tables had to be available to the user. To make this possible without horizontal scrolling and still keeping the tables easy to read, all of the information could not be shown at the same time. Figure 28 shows the PDA version of the page in Figure 26, the table describing each task has been replaced by the name of each task as a link. By clicking on these links the user can show or hide the information of the task, only one task description is shown at a time. The table in Figure 27 where the different tasks are listed horizontally, the table was changed to only display one task and the average values as shown in Figure 29. The user can browse through all the tasks by clicking the arrows under the task name. By using these methods the user can view all the necessary information without excessive vertical scrolling and without any horizontal scrolling.

Scenario analysis

Below are all the details stored about this scenario and its tasks. The first table describes the scenario as a whole. All the roles in the scenario are also listed here. The second table lists all the tasks in the scenario and gives details about each task, such as preconditions, postconditions and dependencies a task may have, and whether this is in relation to location, resources or other tasks.

Show the characterisation of this scenario

Scenario details	
Name	Mobile scenario
Description	Description of the mobile scenario
Roles	Morten John
Created by user	John Erik Taskerud Jensen

No	Task details		Preconditions		Postconditions		Dependencies	
1	Name	Task 3	Location		Location		Role(s)	
	Description	Task 3 description	Resources		Resources		Composite	No
	Responsible	Morten	Tasks	Mobile task	Tasks		Composed of	
2	Name	Mobile task	Location	Home	Location	Home	Role(s)	
	Description	The first mobile task description	Resources	PDA	Resources	Computer	Composite	No
	Responsible	John	Tasks		Tasks	Task 3	Composed of	
3	Name	Task 2	Location		Location		Role(s)	
	Description	Task 2 description	Resources		Resources		Composite	No
	Responsible	John	Tasks		Tasks		Composed of	

Figure 26: View scenario – Web

Mobile complexity indicators of scenario: *Mobile scenario*

In the table below are different indicators that can help you analyse the mobile scenario and help to prioritise and extract non-functional and funcional requirements.

The Overall Scenario Complexity Indicator for this scenario is 2.48

The table first lists the different weights set by the user on each task, and the averages of the characterisations, calculated for the previous page, is repeated. The complexity indicators are thereby listed, for every task, and for the scenario as a whole (weighted values) in the rightmost column.

If you need guidance in how to use the complexity indicators, scroll down the page and read each indicator's description.

Mobile scenario	Task 3	Mobile task	Task 2	Average
Weight	5	3	3	3.6666667
Characterisation average	2.6	1.9	2.85	2.48

Calculated complexity indicators				
General Task Indivcator	1.0	2.6	2.2	1.76
Information Complexity Indicator	3.0	1.0	3.0	2.45
Location Complexity Indicator	3.4	1.8	3.4	2.96
Time Complexity Indicator	3.0	2.0	2.8	2.67
Network Connectivity Indicator	2.33	2.33	2.0	2.24
Network Speed Indicator	3.0	1.0	3.67	2.64
Energy Consumption Indicator	3.5	1.25	3.75	2.95
Transactional Support Indicator	2.0	2.67	2.17	2.23
Mobility Indicator	3.4	1.8	3.8	3.07

Figure 27: View indicators - Web

Figure 28: View scenario - PDA

**Mobile complexity indicators of
scenario: _Mobile scenario_**

In the table below are different indicators that
can help you analyse the mobile scenario
and help to prioritise and extract non-
functional and funcional requirements.

The Overall Scenario Complexity Indicator
for this scenario is 2.48

The complexity indicators are listed for the
selected task, and for the scenario as a whole
(weighted values) in the rightmost column.
Use the links (<< >>) to browse the tasks in this
scenario.

Mobile scenario	Task 3 << >>	Average
Weight	5	3.6666667
Characterisation average	2.6	2.48

Calculated complexity indicators		
General Task Indivcator	1.0	1.76
Information Complexity Indicator	3.0	2.45
Location Complexity Indicator	3.4	2.96
Time Complexity Indicator	3.0	2.67
Network Connectivity Indicator	2.33	2.24
Network Speed Indicator	3.0	2.64
Energy Consumption Indicator	3.5	2.95
Transactional Support Indicator	2.0	2.23
Mobility Indicator	3.4	3.07

Figure 29: View indicators – PDA

9. User Guide

This Chapter describes how to use the MWCF PDA system. Chapter 9.1 describes the welcome page of the MWCF PDA system, in Chapter 9.2 the log in page is described, the Scenario analysis part of the system are specified in Chapter 9.3, the Characterise scenario functionality are described in Chapter 9.4 and finally how to logout are specified in Chapter 9.5. User guides for the MWCF Web system can be found in Siv Oma Rogdar's Master's Thesis [7] and in our in depth study [9].
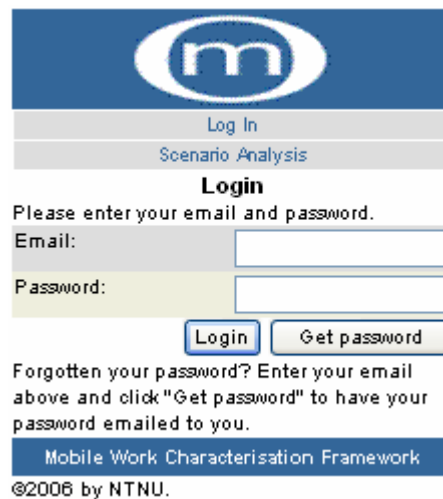
9.1 Welcome page

The first page the user will see of the MWCF PDA system is the welcome page shown in Figure 30. This page has no other functionality than to explain what this site is, and how to use it. The menu on the top contains the features that may be used by any user without registering or logging into the system.
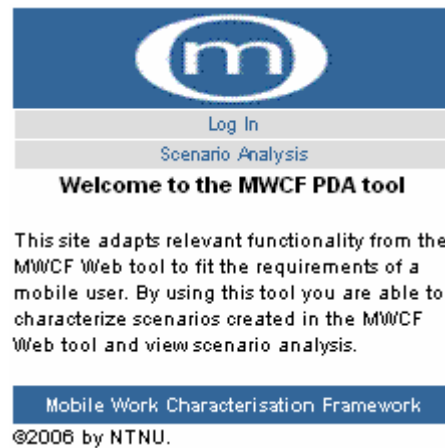
Figure 30: The welcome page

9.2 Log in

If the user is already registered in the system, he may log in using the "Login" link in the menu. The login page is shown in Figure 31. The user must enter the registered e-mail and password. If the password is correct, the user is logged into the system and the page of Figure 32 is shown. This page contains a list of scenarios the user previously has saved. When a user is logged in, his/her name appears on the top grey banner of the page.
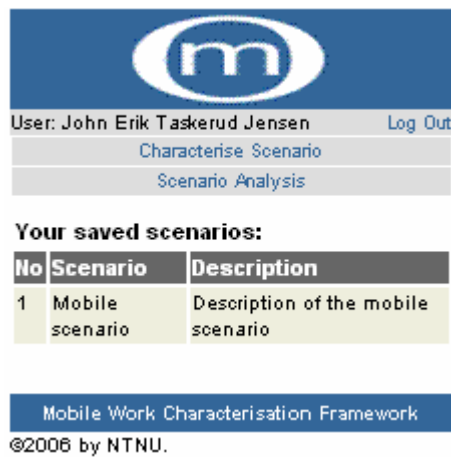
Figure 31: Login page

Figure 32: Logged in

9.3 Scenario analysis

By clicking the "Scenario analysis" link in the menu, the user may view the characterisations of scenarios stored in the system and get the computed indicators of the scenarios. A user may view the analysis of any stored scenario in the system, not only the scenarios created by the user himself/herself. If the user is logged in, he/she will be asked to choose between viewing the scenario analysis of scenarios created by him/her or by others. This is shown in Figure 33.
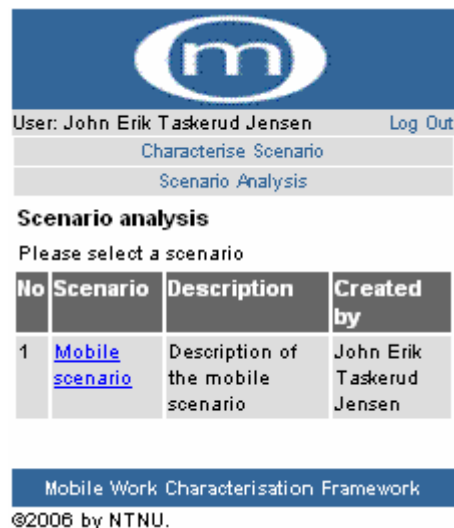
Figure 33: Choose scenarios

If the user is logged in and has chosen to see his/her scenarios in the system, the page of Figure 34 is shown. This page lists all characterised scenarios the user has stored in the system together with the description of the scenarios. The user may choose a scenario by clicking its name.

Figure 34: List of characterised scenarios

After choosing a scenario, the details of the scenario are given. This page is shown in Figure 35. This page introduces the mobile scenario to the user. In the first table on the page are the scenario details. The name, description and roles of the scenario are given, in addition to the name of the creator of the scenario. Below the table of scenario details, is the table of tasks in the scenario. The user can view the details of each task in the scenario by using the links in this table. This is shown in Figure 36. After having studied the details of the scenario, the user may click the "Show the characterisation of this scenario" link, which is located on top of the table of scenario details.

Figure 35: Details of the scenario

Figure 36: Details of the scenario and its tasks

This brings the user to the page in Figure 37. On this page the average of the task weights and the averages of the task characterisations are shown. To see the calculated indicators of the scenario, the user must click the link on the top of the page of Figure 37 saying: "View complexity indicators."
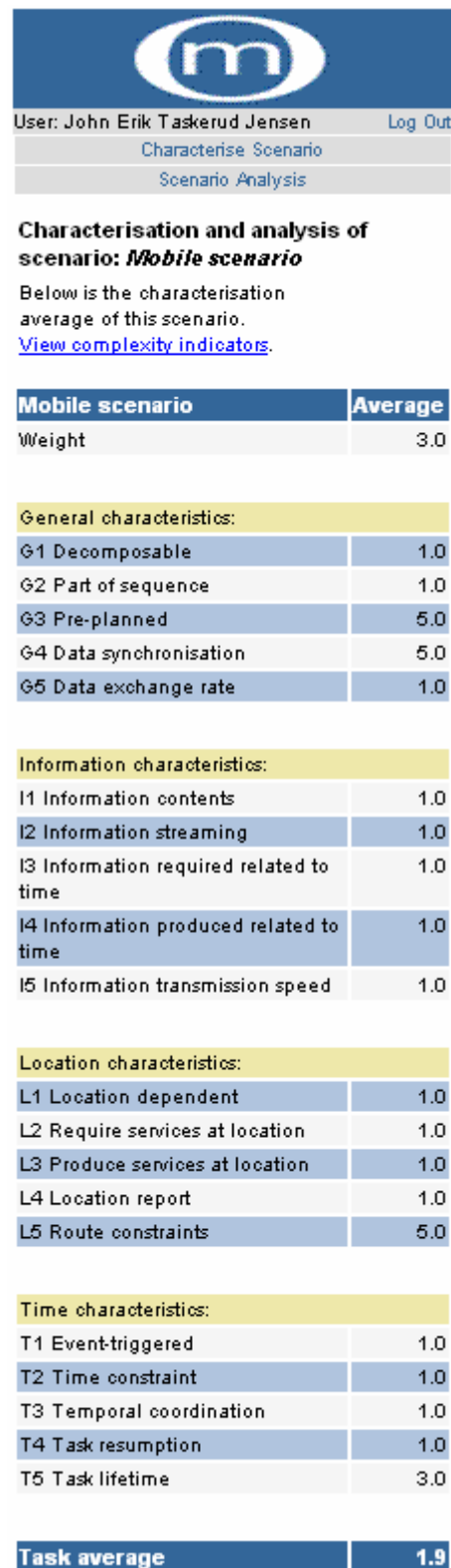
Figure 37: The characterisation of the scenario

This leads the user to the page of Figure 38. On this page, several indicators are computed. The overall complexity indicator of the scenario is shown in the upper part of the page. The tables on this page first lists the different weights set by the user on one selected task, and the averages of the characterisations. The user can browse trough the different tasks in the

scenario by using the "<< >>" links. The complexity indicators are given for the selected task, and for the scenario as a whole (weighted values) in the rightmost column.

Figure 38: The calculated indicators of the scenario

9.4 Characterise scenario

When the user has created a scenario, he may characterise it by clicking the link "Characterise Scenario" in the menu. The user is first presented a list of his scenarios, and must select one to characterise. The list of scenarios is shown in Figure 39. In this case, the list only consists of one scenario; the "Mobile scenario" scenario.
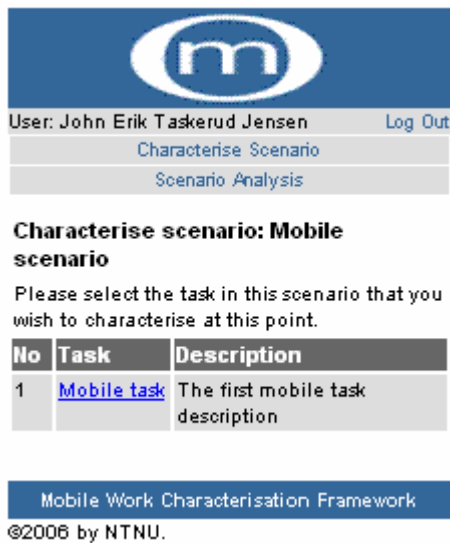
Figure 39: List of scenarios the user may characterise

After the user has chosen a scenario, the tasks of this scenario are listed, as shown in Figure 40. After choosing a task the user is taken to the page of Figure 41.This page shows all the details of the selected task to the user. The user must now select a weight for the task; in this example he/she chooses a weight of 3. After selecting a weight, the user clicks "Next" and the page of Figure 42 is shown. The characterisation process is divided into several steps, consisting of characterising each group separately. The user may now click the "characterise" link on any group to characterise the task within this group. As shown in Figure 43 and Figure 44, the user can choose to view or hide the details of the task to ease the characterisation process. The characteristics are presented by both name and description. The values that may be assigned on each characteristic are given in a drop-down box. By clicking the "Save characterisation" button, the values selected are stored in the system and the user is taken back to the list of groups. The list is slightly changed, because the group that the task has been characterised within has been marked with "Done", in Figure 42 all groups are finished and marked with "Done". The user can edit his characterisation by clicking on the name of the group.

When the task has been characterised within all groups of the framework, the user may click the Validate button to validate the characterisation according to the consistency rules. If any consistency rules are broken they will be listed, as shown in Figure 45. The user can change the values included in a consistency rule by pressing the link with the rule name, or the user can chose to ignore these warnings. If the user chooses to change the values he/she will be taken to an edit scenario page where he/she can change the score of all characterisations involved in the consistency rule, this page is shown in Figure 46. When the user has changed the characterisations, he/she can press the link: "Go back to list of tasks in scenario Name of scenario", i.e. "Go back to list of tasks in scenario Mobile scenario", and choose one of the other tasks of this scenario that has not been characterised yet (if any). After characterising all

of the tasks in a scenario, the user may get the computed indicators of the scenario by using the "Scenario analysis" link in the menu.

Figure 40: Tasks in the Scenario

Figure 41: Select a weight for the task

Figure 42: Select group to characterise

Figure 43: Characterise a group

Figure 44: Details of a task

Figure 45: Validate

Figure 46: Edit characterisation

9.5 Logging out of the MWCF PDA site

A user should always log out of the system by clicking the "Logout" link in the top horizontal banner on every page in the MWCF PDA site. A user will automatically be logged out of the system if he/she is inactive for 1 hour. Also, the user is logged out if he/she closes all open browser windows. Still, the safest way of logging out is by clicking the "Logout" link.

82

10. Testing

Software testing is a process to help identify the correctness, completeness and quality of developed software. This Chapter contains the test plans, test results and test methods used on the MWCF PDA system.

10.1 Test methods

There are many approaches to software testing, but effective testing of complex products is essentially a process of investigation, not merely a matter of creating and following rote procedure. The test methods used in this project are unit and module testing. A unit test is a procedure to verify that a particular module of source code is working properly. Ideally each test case is separate from the others. Module testing consists of testing a collection of depended components, such as an object class or a collection of functions. The testing was carried out during and after development of the different components in the system.

10.2 Functional requirements test plan

These tests are based on the functional requirements in Chapter 7.1 and the test scenarios used when testing the MWCF Web system.

10.2.1 Login

T01: Test if it is possible to log into the system with a registered e-mail and password.

T02: Test whether a user gets rejected when trying to log into the system with an unregistered e-mail and/or password.

T03: Test if the correct password is emailed when requesting the password with a registered email.

T04: Test if the password is emailed when requesting the password with an e-mail that is not registered.

10.2.2 Characterise scenario

T05: Test if the user is given a list of only *his own* scenarios to choose from, and if the list includes all the scenarios he has created.

T06: Test if it is possible to assign the task a weight.

T07: Test if the correct values are stored after having characterised a task on all of the characteristics.

T08: Test if the user is given a warning when having assigned values to characteristics which are not consistent according to the rules of Chapter 2.3.

T09: Test if it is possible to save a characterisation which is inconsistent according to the rules of Chapter 2.3.

T10: Test if the correct characteristics are listed in "edit characterisation".

T11: Test if the correct values are saved when saving all characteristics involved in a certain rule in "edit characterisation".

T12: Test if the complexity indicators may be retrieved after a characterisation of a task.

T13: Test if the complexity indicators may be retrieved after a characterisation of a scenario.

10.2.3 View analysis and interpretations

T14: Test if the user may choose between all the characterised scenarios in the MWCF system.

T15: Test if the user is able to view the characterisation of any characterised scenario in the system.

T16: Test if the calculated complexity indicators are correctly computed.

10.2.4 Test case: Logout

T17: Test whether a user, after logging out of the system, is prevented from accessing Web pages that are only accessible after logging in.

T18: Test whether a user gets logged out of the system after being inactive for 1 hour.

T19: Test if the user gets logged out of the system by closing all open browser windows.

10.3 Functional requirements test results

In this Chapter the test results for the functional requirements are documented. A test can either be approved or rejected. Comments are given if a test is rejected or in special cases.

Login			
Test	**Approved**	**Rejected**	**Comments**
T01	X		
T02	X		
T03	X		
T04	X		

Characterise scenario			
Test	**Approved**	**Rejected**	**Comments**
T05	X		
T06	X		
T07	X		
T08	X		

T09	X		The user can always save a characterisation, regardless of possible consistency issues.
T10	X		
T11	X		
T12	X		
T13	X		

View analysis and interpretations			
Test	Approved	Rejected	Comments
T14	X		
T15	X		
T16	X		

Logout			
Test	Approved	Rejected	Comments
T17		X	A user can access Web pages that should only be accessible after logging in if the user manually types in the URL. This is not possible if the user close the Web browser.
T18	X		The user gets logged out after one hour, but gets redirected to the login page in the MWCF Web system
T19	X		

10.4 Non-functional requirements test plan

These tests are based on the non-functional requirements in Chapter 7.2 and the test scenarios used when testing the MWCF Web system.

10.4.1 Lifespan

NT01: Is the system ready for delivery June 20th 2006?

10.4.2 Interfaces to Web

NT02: Test if all features are available for use by a Web browser

NT03: Test if it is possible to use a PDA to view the characterisation and the calculated complexity indicators of a scenario.

10.4.3 Screen resolution

NT04: Test if the Web pages are viewable with a screen resolution 240*320 pixels

10.4.4 Usability

NT05: Test if the user is notified when entering contradictory data.

10.5 Non-functional requirements test results

In this Chapter the test results for the non-functional requirements are documented. A test can either be approved or rejected. Comments are given if a test is rejected or in special cases.

Test	Approved	Rejected	Comments
NT01	X		
NT02	X		
NT03	X		
NT04	X		
NT05	X		

Part IV – Evaluation of the MWCF system

11. The MWCF Web versus MWCF spreadsheet test
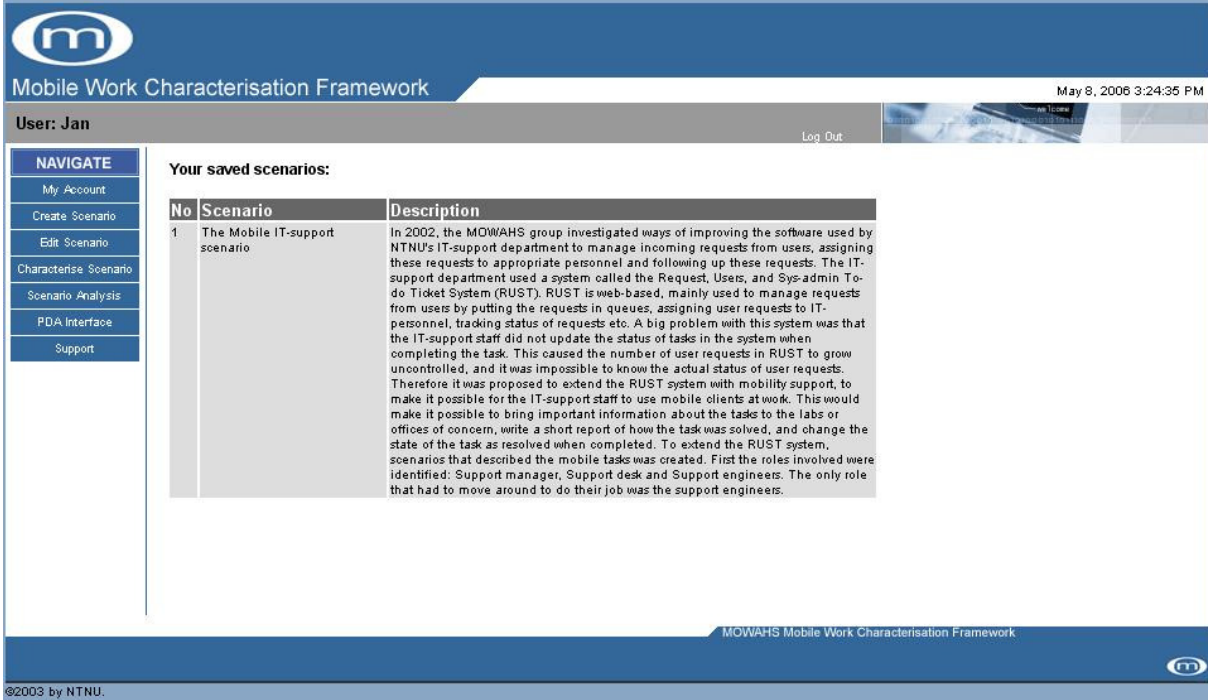
This Chapter describes how the test was carried out and shows the documentation that was given to the testers.

11.1 Intro

The purpose of the test is to compare the usability of the original MWCF spreadsheet system (Figure 47) and the MWCF Web system (Figure 48), and to reveal possible advantages of the usage of consistency rules. The test will be carried out by ten persons. Five of these persons are fifth year's students in master computer science at NTNU and five students are bachelors in computer science from "Høgskolen i Gjøvik", all with software development experience. The testers will be divided in two groups, where one group will characterise a scenario with the spreadsheet system and the other group will characterise the same scenario with the Web system. When the characterisation is completed the testers will be given a set of question concerning the usability of the systems. After the tests are finished all test data will be analysed and compared.

	A	B	C	D	E
1	IT-support	Setting up new comp	Upgrading computers	Assisting users	Average
2	Task Weigth	4	2	5	3,67
3	**GENERAL CHARACTERISTICS**		*Possible scores: 1 - 3 - 5*		
4	G1 Decomposable	1	1	1	1,00
5	G2 Part of sequence	3	3	1	2,09
6	G3 Pre-planned	1	1	1	1,00
7	G4 Data synch	3	3	3	3,00
8	G5 Data exchange rate	3	3	3	3,00
9	**INFORMATION CHARACTERISTICS**		*Possible scores: 1 - 2 - 3 - 4 - 5 , but I2 is only 1 - 3 -*		
10	I1 Info contents	2	2	2	2,00
11	I2 Info streaming	3	3	3	3,00
12	I3 Freshness data required	2	2	2	2,00
13	I4 Freshness data produced	2	2	2	2,00
14	I5 Data transmission speed	2	2	2	2,00
15	**LOCATION CHARACTERISTICS**		*Possible scores: 1 - 3 - 5*		
16	L1:Loc. dependent	5	5	5	5,00
17	L2:Req service at loc	1	1	3	1,91
18	L3:Prod service at loc	1	1	1	1,00
19	L4 Location report	3	3	3	3,00
20	L5:Route constraints	1	1	1	1,00
21	**TIME CHARACTERISTICS**		*Possible scores: 1 - 3 - 5 , but T5 is 1 - 2 - 3 - 4 - 5*		
22	T1:Event-triggered	3	3	3	3,00
23	T2:Time constrained	3	3	5	3,91
24	T3:Temporal coordination	1	1	1	1,00
25	T4:Task resumption	5	5	1	3,18
26	T5:Task lifetime	3	3	2	2,55
27					
28	**General score:**	2,20	2,20	1,80	2,02
29	**Information score:**	2,20	2,20	2,20	2,20
30	**Location score:**	2,20	2,20	2,60	2,38
31	**Time score:**	3,00	3,00	2,40	2,73
32					
33	**Task Average:**	2,40	2,40	2,25	2,33
34					
35	**MOBILE SCENARIO COMPLEXITY INDICATOR**				2,33
36					
37	**Network Connectivity**	2,60	2,60	2,60	2,60
38	**Network Speed**	2,00	2,00	2,00	2,00
39	**Energy Consumption**	2,50	2,50	2,75	2,61
40	**Transaction Support**	3,00	3,00	2,17	2,62
41	**Mobility**	2,00	2,00	2,00	2,00

Figure 47: MWCF spreadsheet system

Figure 48: MWCF Web system

11.2 The Scenario

In 2005 the MOWAHS project group evaluated the MWCF framework by applying the framework on the developing process on a mobile scenario [23]. We decided to use the same scenario for our test. By doing so, we can compare our results with the results from the previous MOWAHS evaluation. Following are the scenario and tasks information introduced to the testers:

The Mobile IT-support Scenario
In 2002, the MOWAHS project group investigated ways of improving the software used by NTNU's IT-support department to manage incoming requests from users, assigning these requests to appropriate personnel and following up these requests. The IT-support department used a system called the Request, Users, and Sys-admin To-do Ticket System (RUST). RUST is Web based, mainly used to manage requests from users by putting the requests in queues, assigning user requests to IT-personnel, tracking status of requests etc. A big problem with this system was that the IT-support staff did not update the status of tasks in the system when completing the task. This caused the number of user requests in RUST to grow uncontrolled, and it was impossible to know the actual status of user requests. Therefore it was proposed to extend the RUST system with mobility support, to make it possible for the IT-support staff to use mobile clients at work. This would make it possible to bring important information about the tasks to the labs or offices of concern, write a short report of how the task was solved, and change the state of the task as "resolved" when completed. To extend the RUST system, scenarios that described the mobile tasks was created. First the roles involved were identified: Support manager, Support desk and Support engineers. The only role that had to move around to do their job was the support engineers. The mobile tasks for the support engineer role are as follows:

Task 1: Install new PCs (Support engineers): Install new hardware and software.
Priority: High (4).

Task 2: Upgrade such PCs (Support engineers): Upgrade existing computers with new hardware and/or software.
Priority: Low (2).

Task 3: Assist users (Support engineers): Help users to solve computer problems (like malfunctioning mouse, keyboard, software etc.).
Priority: Very high (5).

11.3 Documentation

In order to perform the test the testers needed an introduction on the MOWAHS project and the MWCF framework. The information given to the testers are shown in Chapter 11.3.1. The testers using the spreadsheet system also needed documentation on all the characteristics in the MWCF framework, this documentation is shown in Chapter 11.3.2. The characteristics documentation is integrated in the MWCF Web system, so no additional documentation is needed for the testers using the Web system.

11.3.1 MOWAHS and MWCF introduction

The following introduction to the MOWAHS project and the MWCF framework was given to the testers:

MOWAHS is a research project supported by the Norwegian Research Council, and is carried out as a joint effort between the Norwegian University of Science and Technology's (NTNU) departments of software engineering and database technology. The MOWAHS project is concerned with process support for mobile users using heterogeneous devices (PC, PDA, mobile phones etc.) and support for cooperating transactions/workspaces holding work documents.

When developing mobile work support solutions, the analysis phase is of great importance in order to specify the requirements of the system. To support the analysis of mobile work, the Mobile Work Characterisation Framework (MWCF) was developed by the MOWAHS project group. The framework consists of groups of characteristics from which mobile complexity indicators are computed. The requirements to the mobile work support system may be extracted based on these indicators.

With the MWCF Web system the user can create different mobile scenarios and then identify different roles in the scenario. After that, tasks for each role are identified. For each task the user writes a task description, using the task template. Then the user assigns task weights, and uses the system to characterise the tasks. By combining the scores of the characteristics the complexity indicators can be found.

11.3.2 Task characteristics

The following descriptions of the characteristics were given to the testers using the spreadsheet system.

Table 3: Characteristic documentation

Characteristic	Possible values	Description
G1 Decomposable	*(1 No, 3 Uncertain, 5 Yes)*	Decides whether the task is composed of subtasks or not.
G2 Part of sequence	*(1 No, 3 Partial, 5 Yes)*	Specifies if a task has order dependencies with respect to other tasks.
G3 Pre-planned	*(1 Planned, 3 Partial, 5 Adhoc)*	Describes to what degree a task is planned in beforehand.
G4 Data synchronisation	*(1 No, 3 After task, 5 Within task)*	Specifies when a task has to synchronise/merge updated data with other tasks.
G5 Data exchange rate	*(1 None, 3 Once, 5 Many)*	Specifies the rate of data exchange between the current task and other tasks (during its lifetime).
I1 Information contents	*(1 NA, 2 Text, 3 Graphics, 4 Audio, 5 Video)*	Describes the complexity of the information required or produced by the task.
I2 Information streaming	*(1 NA, 3 Discrete, 5 Continuous)*	Describes whether the task requires streaming of data or not.
I3 Information required related to time	*(1 NA, 2 Low, 3 Medium, 4 High, 5 Real-time)*	Describes how important the information required to execute the task is related to time.
I4 Information produced related to time	*(1 NA, 2 Low, 3 Medium, 4 High, 5 Real-time)*	Describes how important the information produced by the task is related to time.
I5 Information transmission speed	*(1 NA, 2 Slow, 3 Medium, 4 Fast, 5 Very fast)*	Describing the expected transmission speed of information used or produced by the task.
L1 Location dependent	*(1 No, 3 Partial, 5 Yes)*	Describes to what degree a task must be executed at a specific location (e.g. that a lecturer must be in a classroom to teach students).
L2 Require services at location	*(1 No, 3 Partial, 5 Yes)*	Specifies if a task needs electronic services available at the location such as printers, network connections, video projectors, fax machines, etc.
L3 Produce services at location	*(1 No, 3 Partial, 5 Yes)*	Specifies if a task produces electronic services at the location that can be used by others at this location, such as information queries, beaming of information, network connectivity etc.
L4 Location report	*(1 No, 3 Partial, 5 Yes)*	Specifies if a task must report its location to the system.
L5 Route constraints	*(1 No, 3 Partial, 5 Yes)*	Specifies if a mobile task must follow a specific route or not when moving around.
T1 Event-triggered	*(1 No, 3 Partial, 5 Yes)*	Decides whether a task is triggered by an event or not.
T2 Time constraint	*(1 No, 3 Partial, 5 Yes)*	Describes if a task must be executed at a specific time or within a specific time-span (e.g. a task must be performed between 10:15 and 12:00).
T3 Temporal coordination	*(1 No, 3 Partial, 5 Yes)*	Describes if a task must be coordinated with other tasks (e.g. different military units must strike at the same time).
T4 Task resumption	*(1 No, 3 Partial, 5 Yes)*	Describes if a task can halt, and then later resume from where it left off (not requiring a complete restart of task).
T5 Task lifetime	*(1 Seconds, 2 Minutes, 3 Hours, 4 Days, 5 Weeks)*	Describes the expected lifetime of a task.

11.4 Questions

The following questions were given to the testers after they had finished the characterisation:

Web system:

- Does the Web system give a good guidance on how to perform the characterisation, if not what is insufficient?
- Did you get any consistency rule violations, if so did you change your characterisation accordingly?

Spreadsheet system:

- Is the accompanying documentation sufficient to perform the characterisation, if not what is insufficient?

Both systems:

- How user friendly is the system, on a scale from 1 to 5 (Where 5 is very user friendly)?
- Was there anything you did not understand during the characterisation, if so, what did you not understand?
- How long time did you use on the characterisation?

12. Test results

In this Chapter the test results are presented. This includes analysis of the characteristic documentation, analysis of the consistency validation, analysis of the characterisation results and suggested improvements and comments from the testers.

12.1 Analysis of the characteristic documentation

Table 4 shows the distribution of results from the characterisation part of the test. The results from both the Web system and the spreadsheet have been combined. Since everyone participating in the test was given the same information regarding the scenario and the characteristics, the results from each tester should not differ too much. The expected distributions should look like the graphs shown in Figure 49, 50 and 51. Other types of distributions, like those shown by the graphs in Figure 49 and 50, may indicate that a particular characteristic is not easily understandable or that the scenario used in the test is vague in that area. A characteristic with strange distributions on all tasks suggests that the reason for the results lie in the description of the characteristic. If there is only one strange distribution on the characteristic it is more likely because the scenario is unclear in that area. If there are two strange distributions the description of the characteristic can not be excluded as the reason and should therefore be looked into. All characteristics with distributions similar to those in Figure 49 and 50 on two or more of its tasks have been investigated to see if there should be made improvements on the explanatory text of these characteristics.

Table 4: Characterisation results

Characteristics	Task 1					Task 2					Task 3				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
G1 Decomposable	4		1		5	4		2		4	4		3		3
G2 Part of sequence	4		5		1	3		5		2	6		2		2
G3 Pre-planned	8		0		2	6		4		0	1		4		5
G4 Data synch	5		5		0	5		5		0	3		5		2
G5 Data exchange rate	6		3		1	7		3		0	3		5		2
I1 Info contents	3	7	0	0	0	2	7	1	0	0	2	7	0	1	0
I2 Info streaming	7		3		0	8		2		0	4		4		2
I3 Information required related to time	3	5	1	1	0	1	7	2	0	0	1	3	2	1	3
I4 Information produced related to time	3	4	3	0	0	2	4	4	0	0	1	2	2	3	2
I5 Information transmission speed	3	2	4	0	1	2	3	4	0	1	1	1	4	1	3
L1 Location dependent	1		6		3	1		6		3	2		5		3
L2 Require services at location	3		4		3	4		5		1	1		6		3
L3 Produce services at location	4		4		2	2		4		4	5		4		1
L4 Location report	6		1		3	5		3		2	4		5		1
L5:Route constraints	8		2		0	7		2		1	6		4		0
T1 Event-triggered	4		4		2	1		6		3	2		1		7
T2 Time constrained	6		3		1	6		3		1	3		4		3
T3 Temporal coordination	7		2		1	6		2		2	6		1		3
T4 Task resumption	2		2		6	3		3		4	3		7		0
T5 Task lifetime	0	1	9	0	0	0	4	5	0	1	0	7	2	1	0
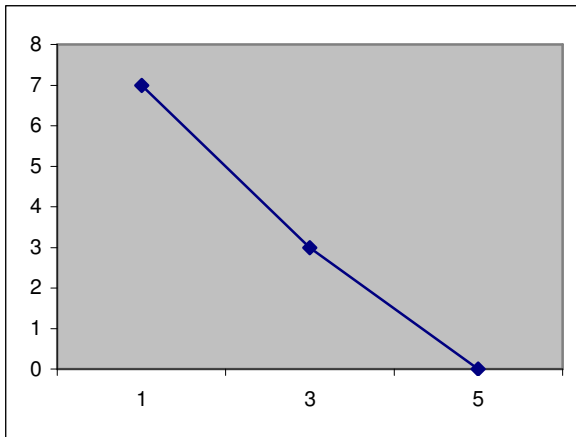
Figure 49: Decreasing distribution
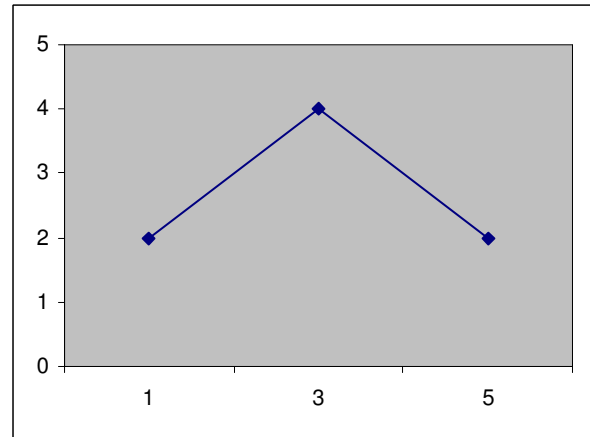
Figure 50: Topped distribution
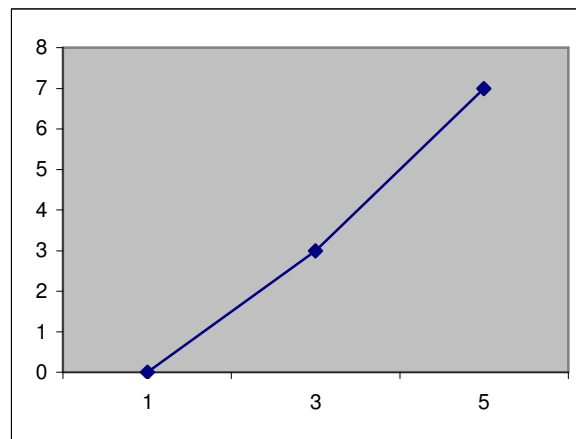
Figure 51: Increasing distribution
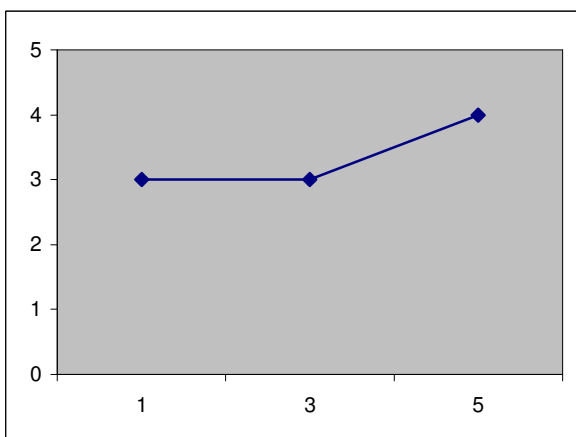
Figure 52: Flat distribution
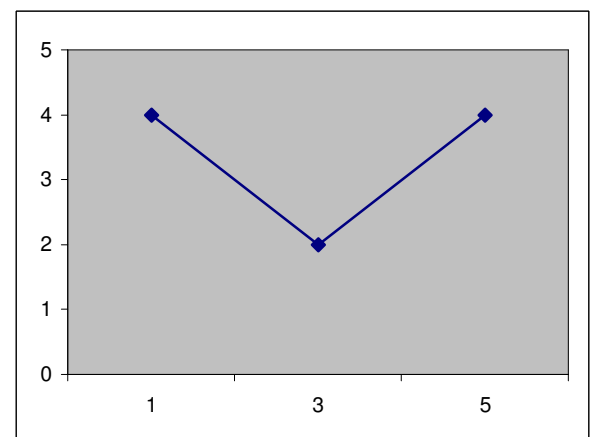
Figure 53: V-shaped distribution
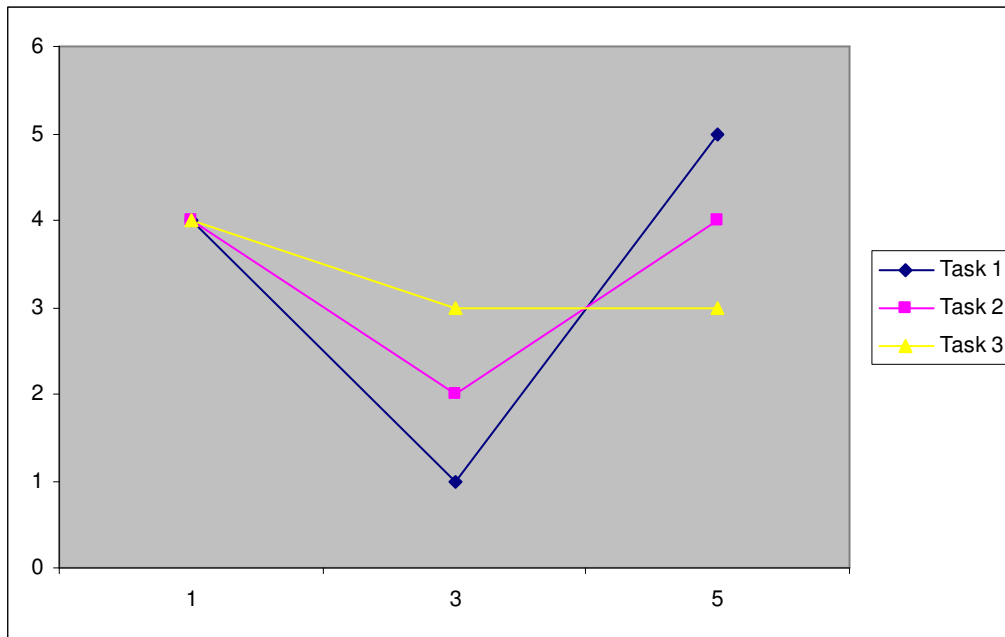
Figure 54: G1 Decomposable

G1 Decomposable
Description: Decides whether the task is composed of subtasks or not.

Comment: As shown in Figure 51 the results from task 1 and task 2 give a v-shaped graph, while the results from task 3 give a flat graph. This clearly indicates that this characteristic has been interpreted differently among the testers and that a revision of its explanatory text is necessary.

Possible improvements: A definition of subtask and an example.
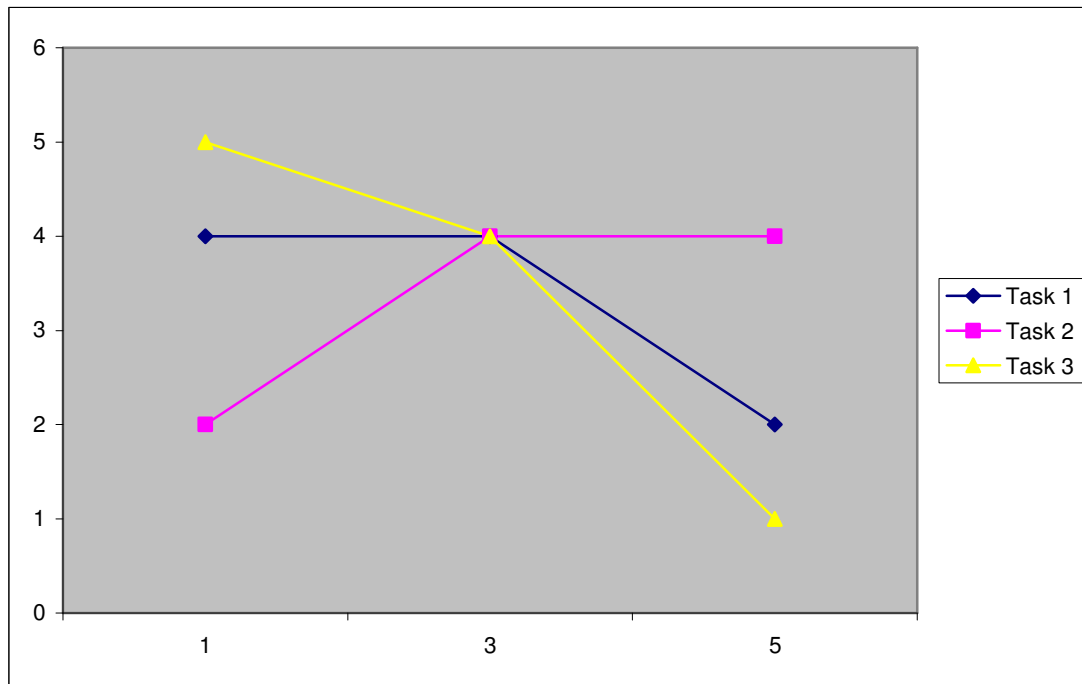
Figure 55: L3 Produce service at location

L3 Produce service at location
Description: Specifies if a task produces electronic services at the location that can be used by others at this location, such as information queries, beaming of information, network connectivity etc.

Comment: As shown in Figure 52 the results from task 1 and task 2 give flat graphs indicating different interpretation among the testers, task 3 does not indicate a problem with the description of the characteristic. The description of this characteristic should be made more understandable.

Possible improvements: Improved documentation of the L3 characteristics with examples of what electronic services are and how they are used.

12.2 Analysis of the consistency validation

One of the main advantages in the Web system is the consistency validation; the spreadsheet system has no option for this. We manually checked the test data from the spreadsheet testers in order to discover possible consistency violations. We also asked the Web system testers to list all consistency violations and to write down if they chose to change their characterisation or not. In total there were fifteen consistency violations, five in the spreadsheet system and ten in the Web system. Following is a listing of the consistency violations, for a full listing of the consistency rules and their descriptions see Chapter 2.3.

Table 5: Consistency violations

Rule	Number of violations		
	Spreadsheet	Web	Total
Rule 1: $I3 \geq 4 \lor I4 \geq 4 \rightarrow I5 \geq 4$		1	1
Rule 3: $L4 = 5 \rightarrow L1 = 5$	2		2
Rule 4: $T3 = 5 \rightarrow G2 = 5$	1	3	4
Rule 5: $I2 = 5 \rightarrow G5 = 5$		1	1
Rule 7: $T2 = 5 \rightarrow G3 = 1$	2	1	3
Rule 8: $G2 = 5 \rightarrow G3 = 1$		4	4

In eight of ten of the consistency violations in the Web system the testers chose to change their characterisation accordingly to the consistency rules. Only one tester chose to keep his/her characterisation on two cases, both were violations of rule 8.

12.3 Analysis of the characterisation results

This Chapter shows the results from the tests on the Web-system and the spreadsheet compared to the suggested characterisation provided by the project manager of the MOWAHS project, Alf Inge Wang. The suggested characterisations have their cells marked with grey. When the answers from the test match the suggested characterisation the answers have been marked bold. This is if the answer in the grey cell is 3 or higher or if the answer in the grey cell is 2 and there are no higher weights for that characteristic. Table 6 shows the results form the spreadsheet test while Table 7 shows the results from the Web-system. Even though the data sets are small, there are still indications as to whether the scenario and the characteristics are easily understandable or not.

Table 6: Spreadsheet results

Spreadsheet	Task 1					Task 2					Task 3				
	1	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
G1 Decomposable	2	0			**3**	2	1			**2**	**3**	1			1
G2 Part of sequence	**2**	3	0			0	**5**	0			**3**	1	1		
G3 Pre-planned	**4**	0	1			**3**	2	0			1	1			**3**
G4 Data synch	**1**	4	0			**1**	4	0			1	**2**	2		
G5 Data exchange rate	**1**	3	1			**3**	2	0			1	**4**	0		
I1 Info contents	1	**4**	0	0	0	1	**4**	0	0	0	1	**3**	0	1	0
I2 Info streaming	**4**	1	0			**4**	1	0			**2**	1			2
I3 Information required related to time	1	**4**	0	0	0	1	**4**	0	0	0	0	**2**	1	0	2
I4 Information produced related to time	2	**2**	1	0	0	2	**2**	1	0	0	0	**2**	1	0	2
I5 Information transmission speed	1	2	**2**	0	0	1	2	**2**	0	0	0	1	**2**	0	2
L1 Location dependent	0	3			**2**	0	4			**1**	1	2			**2**
L2 Require services at location	2	**3**	0			3	**2**	0			0	**4**	1		
L3 Produce services at location	1	**3**	1			2	**2**	1			**3**	2	0		
L4 Location report	**3**	1	1			**3**	1	1			1	**4**	0		
L5:Route constraints	**4**	1	0			**4**	1	0			**3**	2	0		
T1 Event-triggered	**1**	3	1			0	**4**	1			0	1			**4**
T2 Time constrained	**3**	2	0			**3**	2	0			1	**1**			3
T3 Temporal coordination	**4**	1	0			**3**	1	1			**3**	1	1		
T4 Task resumption	0	1			**4**	0	2			**3**	1	**4**	0		
T5 Task lifetime	0	0	**5**	0	0	0	2	**3**	0	0	0	**3**	1	1	0

The spreadsheet results have a total of 51 hits out of 60 possible. If the limit for accepting a hit is raised to 4, there will be a total of 19 hits.

Table 7: Web results

Web	Task 1					Task 2					Task 3				
	1	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
G1 Decomposable	2		1		**2**	2		1		**2**	**1**		2		2
G2 Part of sequence	**2**		2		1	3		**0**		2	**3**		1		1
G3 Pre-planned	**4**		0		1	**3**		2		0	0		3		**2**
G4 Data synch	**4**		1		0	**4**		1		0	2		**3**		0
G5 Data exchange rate	**4**		0		1	**4**		1		0	2		**1**		2
I1 Info contents	2	**3**	0	0	0	1	**3**	1	0	0	1	**4**	0	0	0
I2 Info streaming	**3**		2		0	**4**		1		0	**2**		3		0
I3 Information required related to time	2	**1**	1	1	0	0	**3**	2	0	0	1	**1**	1	1	1
I4 Information produced related to time	1	**2**	2	0	0	0	2	**3**	0	0	1	**0**	1	3	0
I5 Information transmission speed	2	0	**2**	0	1	1	1	**2**	0	1	1	0	**2**	1	1
L1 Location dependent	1		3		**1**	1		2		**2**	1		3		**1**
L2 Require services at location	1		**1**		3	1		**3**		1	1		**2**		2
L3 Produce services at location	3		**1**		1	0		**2**		3	**2**		2		1
L4 Location report	**3**		0		2	**2**		2		1	3		**1**		1
L5:Route constraints	**4**		1		0	**3**		1		1	**3**		2		0
T1 Event-triggered	**3**		1		1	1		**2**		2	2		0		**3**
T2 Time constrained	**3**		1		1	**3**		1		1	2		**3**		0
T3 Temporal coordination	**3**		1		1	**3**		1		1	**3**		0		2
T4 Task resumption	2		1		**2**	3		1		**1**	2		**3**		0
T5 Task lifetime	0	1	**4**	0	0	0	2	**2**	0	1	0	**4**	1	0	0

The Web-system results have a total of 44 hits out of 60 possible. If the limit for accepting a hit is raised to 4, there will be a total of 10 hits.

The spreadsheet system has more hits than the Web system. The Web system should have an equal or higher correctness compared to the spreadsheet system, since the Web system has consistency validation. These results are probably caused by the limited scenario description and that the characteristic descriptions in the framework are not sufficient. Thus the results from the test may be random and not linked to the system. In order to verify this result the test should be performed with more testers.

Overall the results shows a good match with the correct characterisation, in most cases where there is a difference the results from the test is close to the correct characterisation. This indicates that first time users can get quite good results from characterising their scenarios. If the limit for accepting a match between the results and the correct characterisation is raised there is a dramatic fall in accuracy. This indicates that even though the overall understanding of the characteristics and the scenario is acceptable, there is room for improvements preventing poor characterisations for inexperienced users and making the framework easier to understand.

12.4 Suggested improvements and comments from the testers

In this Chapter suggested improvements and comments from the testers that are not listed in previous Chapters are listed and discussed.

12.4.1 Both systems

The most discussed topic by both the spreadsheet testers and the Web testers are the characteristic descriptions. The characteristic description used in the Web system and the spreadsheet system are identical, the only differences are that the spreadsheet documentation is on a separate document. Some testers find the descriptions insufficient in general, and some users list specific characteristic descriptions. These characteristics are:

- **G2 Part of sequence**
  Description: Specifies if a task has order dependencies with respect to other tasks.

  Possible improvements: Add more information about "other tasks", and an example.

- **I3 Information required related to time**
  Description: Describes how important the information required to execute the task is related to time.

  Possible improvements: Add an example.

- **I5 Information transmission speed**
  Description: Describing the expected transmission speed of information used or produced by the task.

  Possible improvements: Better description of the transmission speed, is it needed transmission speed or expected transmission speed. Add an example.

- **L4 Location report**
  Description: Specifies if a task must report its location to the system.

  Possible improvements: Add an example

- **L5:Route constraints**
  Description: Specifies if a mobile task must follow a specific route or not when moving around.

  Possible improvements: Clarify "specific route", is it always the same or can it change for each time the task is executed. Add an example.

- **T1:Event-triggered**
  Description: Decides whether a task is triggered by an event or not.

  Possible improvements: A definition of "event". Add an example.

12.4.2 Web system

Another issue commented by the testers is that the Web system has a steep learning curve, it take some time to understand what they are supposed to do, but when they overcome that the system is easy to use. Many of the Web system testers comment that the guidance given by the system is good.

When validating the characterisation the Web system gives no indication that the validation is completed if there is no consistency violations .A suggested improvement of the Web system is that the system should give an indication that the validation is completed when there are no consistency violations.

12.4.3 Spreadsheet system

A tester commented that to use the system it presuppose that the user has some background knowledge of the system, and that it is bothersome to look up the information needed to perform the characterisation. The system also requires that specific software (MS Excel) is available for the user.

12.4.4 Time used and usability

The testers were asked how user friendly is the system, on a scale from 1 to 5. In average the Web system got a score of 3.8 and the spreadsheet system got a score of 2.8. The reason for the high score on the Web system probably is that the Web system gives a more step by step introduction to the framework, and it requires less prestudy on the framework. The Web system also has consistency validation which helps the user in the characterisation process.

The testers were also asked to time their characterisation, in average the Web system testers used 21 minutes and the spreadsheet testers used 20 minutes. Testers commented that since it is their first time using the system it takes some time getting used to it, and that they would use shorter time on their next characterisation.

12.5 Conclusion

The main goal of this test was to determine if the MWCF Web system has improved usability compared to the MWCF spreadsheet system, and to examine possible benefits of the consistency rules. The feedback from the testers indicates that the Web system is more user friendly than the spreadsheet system. On a scale from 1 to 5, the Web system got a score of 3.8 and the spreadsheet system got a score of 2.8 concerning how user friendly the systems are.

Even though the Web system got a higher score concerning user friendliness both the Web system and spreadsheet system works satisfactorily as tools to characterise scenarios. First time users have no problems with learning how to use the tools and then to use the tools to characterise a scenario. The problems some users experienced in the characterisation processes came mainly from the characteristic descriptions. Many of the characteristic descriptions are short and insufficient for first time users. To improve the usability of the framework and to ease the learning curve for first time users, improvements to the characteristic descriptions should be made and more examples should be added. In the Web system this can be solved by adding an option in the characteristic descriptions which gives a

more detailed description with examples for each characteristic. Another option would be to add a tutorial.

The spreadsheet system had higher correctness than the Web system. This is probably a result of the limited descriptions of the characteristics and the scenario, and the size of the data set. To further investigate this issue a new test with a higher amount of testers should be performed. The framework is subjective, so the suggested characterisation is not necessarily correct.

During the characterisation process, the Web system testers had ten consistency validations. In eight of ten of these consistency violations the testers chose to change their characterisation accordingly to the consistency rules. Only one tester chose to keep his/her characterisation on two cases. This indicates that the validation process is important, and a helpful option in the Web system.

The Web system testers used slightly more time on their characterisation than the spreadsheet testers. The reason for this can be that the Web system has a validation option, and it takes longer time to navigate through the pages. But time is not critical in the characterisation process. The most important properties are consistency and quality.

Part V – Evaluation

13. Further work

This Chapter contains suggestions about what should be done to the MWCF system to increase its usability.

To ease the steep learning curve of the framework, better descriptions for the characteristics is needed. These descriptions should contain good examples of how the characteristics should be interpreted and maybe also how they should not be interpreted. To further help the understanding of the characteristics, a scenario with a detailed description and a "correct" characterisation with an explanation of why that particular characterisation is correct can be added. A tutorial that helps first time users to get to know the system and the framework will also help to ease the learning curve. This may be an online service or a downloadable file available from the MWCF site.

To adapt to a large number of scenarios some changes has to be made to the "View Scenarios" page both on the Web system and the PDA system. The ability to search for specific scenarios and to sort them by different criteria's should be added. The number of scenarios shown on the page should also be limited to a suitable number to prevent a lot of scrolling for the user. Scenarios not shown on the page should be available through browsing functionality.

The "Support" page should be made updatable through the administrator interface so that the FAQ can be updated with relevant questions and answers.

The result from the characterisation is currently only available through the different pages under "Scenario Analysis". Giving the user an option to generate a PDF file containing the scenario and its tasks, the characterisation with the characteristic descriptions and the computed indicators with their descriptions is a possible improvement to the system. This will help the user document his scenario and will also be helpful if he/she wants to discuss the characterisation with others face to face.

Finally the system and the framework should be tested in a real development process, to see whether it is sufficient in its current state, or if other improvement or changes has to be made.

14. Conclusion

This thesis was separated in three goals, to improve the MWCF Web system, to implement a PDA version for the MWCF framework and to test the MWCF Web system versus the MWCF spreadsheet system.

Goal 1: Improving the MWCF Web system
By implementing the "view rule" functionality and by designing and implementing functionality that let the users easily change their characterisations according to the consistency validation, this goal has been met.

Goal 2: Implement a PDA version for the MWCF framework
The desired functionality from the MWCF system is now available for PDA. This has been achieved by designing and implementing a new set of jsp-files adapted to the small screens of PDA's.

Goal 3: Test MWCF Web system versus MWCF spreadsheet system
This goal has been met by letting five persons characterise a scenario through the Web system and letting another five persons characterise the same scenario through the spreadsheet system. By analysing the results from the test it becomes clear that the consistency validation is a helpful addition to the system and that the testers feel that the Web system is more user friendly than the spreadsheet system. Another conclusion from the test is that the framework itself needs some changes and additions, mainly in the descriptions of each characteristic, to improve the usability.

During this thesis a few problems and ways to further improve the systems was discovered. Some of these problems have been fixed and a few improvements have been made to the system. This is described in Chapter 6.2. Due to time constraints, all problems have not been fixed and some improvements are not implemented. These are further described in Chapter 13.

The MWCF Web system and PDA system are now available for use in software development settings. The systems are fully functional and support all the features of the MWCF framework. To make it a helpful tool not only to experienced developers the characteristics of the framework should be made more understandable and their descriptions should be rewritten to prevent misunderstandings resulting in low quality characterisations.

Part VI – Appendix

A - References

[1] The Telework Advisory Group. Online 2005 Available from: www.workingfromanywhere.org/index.htm [Accessed March, 2006]

[2] George H. Forman and John Zahorjan. Challenges of mobile computing. Article 1994

[3] Mobile Wireless Network Research Group. Online 2005. Available from: www.research.att.com/areas/wireless [Accessed March, 2006]

[4] MOWAHS - Mobile Work Across Heterogenous System. Online 2001. Available from: www.mowahs.com [Accessed March, 2006]

[5] Carl-Fredrik Sørensen, Alf Inge Wang, Hien Nam Le, Heri Ramampiaro, Mads Nygård, And Reidar Conradi. The MOWAHS characterisation framework for mobile work.In Proc. IASTED International Conference on Applied Informatics 2002 (AI'2002), Innsbruck, Austria, 18-21, February, 2002.

[6] Heri Ramampiaro, Alf Inge Wang, Carl-Fredrik Sørensen, Hien Nam Le, and Mads Nygård. Requirement Indicators for Mobile Work: The MOWAHS Approach. Technical Report, Dept. of Computer and Information Science, NTNU, 2001.

[7] Siv Oma Rogdar. A Web system for characterising mobile work. Master thesis at NTNU, July 2003. Provided by MOWAHS project manager Alf Inge Wang.

[8] Jakub Hajduk. Extension of the Web system for the Mobile Work Characterisation Framework. Master thesis at NTNU, June 2004, provided by MOWAHS project manager Alf Inge Wang.

[9] Morten Brokerud, John Erik Taskerud Jensen. MOWAHS - Evaluering og videreutvikling av MOWAHS rammeverk for mobilt arbeid. In depth study at NTNU, December 2005. Available from: www.idi.ntnu.no/grupper/su/fordypningsprosjekt-2005/brokerud-jensen-fordyp05.pdf [Accessed March, 2006]

[10] Sun Developer Network. Online 2005. Available from: http://java.sun.com [Accessed November, 2005]

[11] Dr. Bob's Introduction to Java Beans. Online 2005 Available from: www.drbob42.com/JBuilder/jb210t.htm [Accessed November, 2005]

[12] Java World. Online 2005. Available from: www.javaworld.com [Accessed March, 2006]

[13] Apache Tomcat. Online 2005. Available from: http://jakarta.apache.org/tomcat [Accessed March, 2006]

[14] The Tomcat Book Project. Online 2005. Available from: http://tomcatbook.sourceforge.net [Accessed November, 2005]

[15] Apache Struts Project. Online 2005. Available from: http://struts.apache.org [Accessed March, 2006]

[16] MySQL 5.0 Reference Manual. Online 2005. Available from: http://dev.mysql.com/doc/refman/5.0/en [Accessed November, 2005]

[17] O'Reilly Online Catalog. Online 2005. Available from: www.oreilly.com/catalog/prdindex.html [Accessed November, 2005]

[18] How Stuff Work. Online 2005. Available from: http://electronics.howstuffworks.com [Accessed March, 2006]

[19] PC Magazine – Smart Phones. Online 2004. Available from: www.pcmag.com/article2/0,1759,1651067,00.asp [Accessed March, 2006]

[20] Canalys. Online 2005. Available from: www.canalys.com [Accessed December, 2005]

[21] Visual Paradigm – UML 2 Diagrams. Online 2005. Available from: http://www.visual-paradigm.com/VPGallery/diagrams/Sequence.html [Accessed December, 2005]

[22] Sun Developer Network, JavaBeans FAQ. Online 2006. Available from: http://java.sun.com/products/javabeans/faq/faq.general.html [Accessed March, 2006]

[23] Alf Inge Wang, Carl-Fredrik Sørensen, Heri Ramampiaro, Hien Nam Le, Reidar Conradi and Mads Nygård.Using the MOWAHS Characterisation Framework for Development of Mobile Work Applications. Proc. 6th International Conference on Product Focused Software Process Improvement, 13-16 June, 2005, Oulu Finland.

[24] Victor R. Basili, The Experimental Paradigm in Software Engineering, Online 1993. Available from: http://www.cs.umd.edu/~basili/publications/chapters/C17.pdf [Accessed June, 2006]

B - Installation guide

This guide shows you how to get the MWCF application up and running. The installation of the necessary software and how to configure them, and how to install and configure the MWCF application is explained in this guide.

Database installation

The database used for this system is a MySQL database. If you use another version of MySQL make sure that the database driver is compatible.

Download

MySQL 4.1.20 can be downloaded from http://dev.mysql.com/downloads/mysql/4.1.html

Install

Install MySQL by running the downloaded exe-file.

Setup

After you have installed MySQL you may create the database by running the SQL script found at the CD accompanying this report. This will also create a fictive user, which you may use to log into the application as an administrator. You should change the fictive e-mail, name and password of this user after logging in.

Installing Java Software Development Kit (JDK)

The Java Software Development Kit (J2SDK), version 1.4.1 03 or newer, must be downloaded, installed and included in the classpath.

Download

J2SDK 1.4.1 03 can be downloaded from http://java.sun.com/j2se/1.4.1/download.html

Install

Install the J2SDK by running the downloaded exe-file.

Include in classpath

Set an environment variable JAVA HOME to the pathname of the directory into which you installed the JDK.

Installing Tomcat

Tomcat is used both as a Servlet container and as a stand-alone Web server in the MWCF application. During the development of the system, Tomcat version 4.1.31 was used.

Download

Tomcat 4.1.24 can be downloaded from ttp://tomcat.apache.org/download-41.cgi

Install

Run the exe-file to start a wizard that will install Tomcat on your computer.

Configure

Set an environment variable CATALINA HOME to the pathname of the directory into which you installed Tomcat. The Windows installer places a "Start Tomcat" item in the Start menu. Click the "Start Tomcat" to check that Tomcat is installed correctly. Use an internet browser to navigate to http://localhost:8080. A Tomcat welcome page will show if the installation was successful.

Configure the MWCF application

To get the MWCF application to fit the new environment, there are some configurations that
need to be done.

Copy MWCF to Webapps directory

Included on the CD-ROM accompanying this report is a folder named mwcf. Copy this folder
into the "Tomcat/Webapps" directory of your Tomcat installation, for example C:/Program
Files/ApacheGroup/Tomcat 4.1/Webapps.

Edit properties file

In the mwcf/WEB-INF/classes directory is a properties file named
ApplicationResources.properties. Open this file (using NotePad or a similar program) and
scroll down to the last part where the following text appears:

```
//---------------------------------------------//
// CHANGE THESE FIELDS TO FIT THE MWCF CONTEXT//
//
// You should ONLY change the right-hand side of the
// equal symbols.
//---------------------------------------------//
mail.smtp.host=smtp.stud.ntnu.no
mail.from.address=your@mail.adr
//---------------------------------------------//
```

Here you should change the value of the smtp server to the server the MWCF application will
be using, and the e-mail address to the e-mail of the Web administrator of the MWCF system.

Edit DB-config.txt configuration file

In the mwcf/WEB-INF/ directory is a file named DB-config.txt. Open this file, and navigate
to the following xml lines:

```
<!-- MySQL dB username and password for dB connections  -->
   <!-- Change to your DB's username and password-->
   <parameter>
    <name>username</name>
    <value>theusername</value>
   </parameter>
   <parameter>
    <name>password</name>
    <value>thepassword</value>
   </parameter>

   <!-- Class name for mysql JDBC driver -->
   <parameter>
     <name>driverClassName</name>
     <value>com.mysql.jdbc.Driver</value>
   </parameter>

   <!-- The JDBC connection url for connecting to your MySQL dB.
      The autoReconnect=true argument to the url makes sure that the
      mm.mysql JDBC Driver will automatically reconnect if mysqld closed the
      connection.  mysqld by default closes idle connections after 8 hours.
      -->
   <!-- Change to the url to match your DB -->
   <parameter>
```

```
  <name>url</name>
  <value>jdbc:mysql://localhost/test</value>
</parameter>
```

The parameters "username" and "password" must be changed to fit your database, the same goes for the "url" parameter. The "driverClassName" parameter can be left as it is. The driver is included in the CD and does not have to be downloaded. You must however copy it from the "driver" folder on the CD to your "Tomcat\common\lib" folder. After changing these parameters copy the whole contents of "DB-config.txt", navigate to your "Tomcat\conf" folder and open the "server.xml" file. Locate the "</Host>" tag and paste the text from "DB-config.txt" directly above it, save the "server.xml" file and restart the Tomcat server.

Start application and edit user

You may now test theMWCFapplication. Start Tomcat and navigate to http://localhost:8080/mwcf/. The index.jsp page should be shown. In the database script that was run to setup the database, a default administrator was inserted into the database. Log into the MWCF application through the Login page with the following data:

E-mail: **admin@mwcf.com**, and password: **admin**

This logs you into the system as an administrator. You should change the e-mail, password and the name of the user after logging in, by clicking the "User account" link in the menu and providing the correct data.

C - Test: Questions and answers

Spreadsheet systemet:

- Er vedlagt dokumentasjon god nok til å utføre karakteriseringen, eventuelt hva er mangelfullt?
  - Beskrivelsen av enkelte Characteristics er litt kortfattet. Kan være vanskelig i enkelte tilfeller å sette dem i sammenheng med aktuell task
  - Bedre forklaring av characteristics, kanskje eksempler
  - Descriptions kan være bedre forklart. For eksempel var det ikke klart for meg at "information" i I3 characteristics kun betydde elektroniske data – ikke for eksempel samtale med personen du utfører support for
  - Noen problemer med å forstå hva som menes med de forskjellige punktene. Følte at en del av vektene jeg fylte inn ble formet veldig av min tolkning av oppgaven og beskrivelsen til karakteristikkene
  - Avhengigheter av andre tasks, gjelder det tilsvarende tasks, eller også tasks av de to andre typene?

- Hvor brukervennlig er systemet, rangert på en skala fra 1 til 5 (Der 5 er meget brukervennlig)?
  - 2
  - 2
  - 5
  - 1 – Det forutsettes at brukeren har bakgrunnsforståelse for systemet, og det er tungvint å lete frem informasjonen som behøves for å fylle inn skjemaet. Systemet krever også at spesiell programvare er tilgjengelig hos den som fyller inn skjemaet.
  - 4

- Var det noe du ikke forsto under karakteriseringsprosessen, eventuelt hva?
  - Var greit nok
  - Nei
  - Descriptions kunne være litt vage.
  - Hva som eksakt mentes med lokasjonsrapportering, ruterestriksjoner og om oppgavene blir trigget av en hendelse. Generelt ble vektingen gitt utifra min tolkning og hvordan jeg subjektivt vil se på løsingen av oppgavene.
  - Nei

- Hvor lang tid brukte du på å karakterisere?
  - 16 min
  - 15 min
  - 20 min
  - Ca. 30 min (men ville brukt mye kortere tid ved neste karakterisering)
  - 17 – 18 min

Web-systemet:

- Gir Web-systemet bra veiledning på hva og hvordan karakteriseringen skal gjennomføres, eventuelt hva er mangelfullt?
  - Ja, men ikke alt som er like klart
  - Jeg synes Websystemet ga en bra veiledning på hvordan karakteriseringen skulle gjennomføres.
  - Greit nok, men syntes kanskje spørsmålene var litt sære i forhold til hva man egentlig spørres om
  - Bra veiledning
  - Ja

- Fikk du noen regelbrudd da du validerte karakteriseringen? Hvis ja, valgte du å endre på karakteriseringen?
  - Upgrade PC's rule4, rule 5 og 8, endret Assist Users., rule 1, endret
  - Nei
  - 2:
    Regel 4, T3:5, G2:3 – Forandret da G2 til 5
    Regel 8, G2:5, G3: 3 – Lar det stå
    3:
    Regel 4, T3:5, G2:1 – Forandret da G2 til 5
    Regel 7 T2: 5, G3: 5 – Forandret da T2 til 1
    Regel 8 G2: 5, G3:1 - Lar det stå
  - Nei
  - Ja, i task 3 fikk jeg brudd på regel 8, endret G2 fra (5) Yes til (1) No

- Hvor brukervennlig er systemet, rangert på en skala fra 1 til 5 (Der 5 er meget brukervennlig)?
  - 4
  - 4
  - 3, systemet har en brå læringsterskel, men når man først skjønner hva man skal gjøre, går det greit
  - 4
  - 4

- Var det noe du ikke forsto under karakteriseringsprosessen, eventuelt hva?
  - Information transmission speed ?
  - Nei
  - Nei
  - Nei
  - Validate ga ingen direkte indikasjon på at valideringen hadde blitt utført om det ikke var noen regelbrudd.

- Hvor lang tid brukte du på å karakterisere?
  - 22 min
  - 18 min
  - 30 min
  - 15 min
  - 20 min