Chapter IV
From Scenarios to Requirements in Mobile Client-Server Systems

Alf Inge Wang,
Norwegian University of Science and Technology, Norway

Carl-Fredrik Sørensen,
Norwegian University of Science and Technology, Norway

Hien Nam Le,
Norwegian University of Science and Technology, Norway

Heri Ramampiaro,
Norwegian University of Science and Technology, Norway

Mads Nygård,
Norwegian University of Science and Technology, Norway

Reidar Conradi,
Norwegian University of Science and Technology, Norway

ABSTRACT

This chapter describes a requirement analysis framework that may be used as a tool for developing client-server systems for mobile workers. The framework concerns the initial analysis process for the development of a mobile system starting by describing scenarios of the work of mobile workers and resulting in a description of priorities both nonfunctional and functional requirements of the end-system. The framework describes a three step requirement process that includes 1) Elicit scenarios, 2) Scenario analysis, and 3) Requirement analysis. These steps will produce outputs that will be used to assess the requirements of the system. The requirement analysis process is described in detail through templates used to produce output and illustrating examples from the analysis of the development of mobile IT-support system.

INTRODUCTION

The development in mobile computing has changed the way we communicate and work. Mobile phones, smart phones, and PDAs, whose computing capacities become more and more powerful in terms of processing speed, storage capacity and operating time, have become necessary tools through functionality such as SMS, calendars, and WAP-browsers. The more powerful mobile devices have made it possible to create software systems for mobile workers that can better support and also improve their work processes. Such systems typically consist of various mobile clients connected to a server. They provide the mobile worker with information related to the running of planned tasks, and opportunities for filling in forms and reports on various locations, while being on the move.

At the same time, many wireless network technologies are also developed and deployed, for example, Bluetooth, wireless USB, wireless LAN or Universal Mobile Telecommunications System (UMTS). Many mobile devices can communicate with other electronic devices in the working environment, making it possible to make measurements and take advantage of services in the work environment through wireless communication. The environment for accessing and processing information is rapidly changing from stationary to mobile and location independent. This new work environment, called the mobile work environment, allows people to work in flexible and efficient ways.

A mobile environment is different from traditional distributed environments due to its unique characteristics such as the mobility of users or computers, the limitation of computing capacity of mobile devices, and the frequent and unpredictable disconnection of wireless networks (Forman & Zahorjan, 1994; Satyanarayanan, 1996). Therefore, development of mobile systems is different from development of distributed systems. In other words, when designing a mobile system, we have to overcome challenges due to physical mobility of the clients, the portability features of mobile devices and the fact that the communication is wireless. Thus, it is important that these issues are examined carefully when considering the system requirements, in terms of both functional and nonfunctional requirements. As defined in Sindhgatta and Thonse (2005), (IEEE), functional requirements include all the logical and specific behaviours of a system, while nonfunctional requirements concern the overall characteristics of the system-like performance, reliability and security. Note that, in mobile systems, the nonfunctional requirements play vital roles in system development.

We have developed a requirement analysis framework that examines important issues to be considered when designing and developing such mobile systems. The framework concerns the initial analysis process in the development of a mobile system starting by describing scenarios of the work of the mobile workers[1], and resulting in a description of priority, nonfunctional and functional requirements of the end-system.

In this chapter, we will describe the MOWAHS analysis framework consisting of a process, the characterisation of scenarios, computation of complexity indicators, and determination of nonfunctional and functional requirements. One important goal of this framework is to identify the parts that are most complex and perhaps hardest to implement in a mobile support system. The framework has previously been successfully used to analyse mobile scenarios such as mobile journalists, mobile researchers and m-learning (Ramampiaro, Wang, et al., 2003) to compare their characteristics. The framework has also been used to develop a mobile IT-support system (Wang, Sørensen, et al., 2005).

Our framework is aimed at development of systems for supporting any kind of mobile workers, like craftsmen, salesmen, emergency staff, police, public transportation drivers, and service staff. For system developers, we expect that they have a basic knowledge of software engineering for he/she to be able to use our MOWAHS framework to derive the requirements. This includes knowledge of all relevant phases in a software development life cycle (e.g., the waterfall model and the like). Knowledge of UML would also be useful.

BACKGROUND

The process to identify the requirements of a mobile client-server-based system is very different from a nonmobile one. This is due to the unique characteristics of mobile environments that are the mobility of users or computers, the limitation of computing capacity of mobile devices, and the frequent and unpredictable disconnections of wireless networks. In this section, we first address the main characteristics of mobile environments, which have a strong impact on the behaviours of mobile clients and the development process of mobile systems. Later, we review work related to our MOWAHS requirement analysis framework.

The main characteristics of the mobile environments that are addressed in this section include: the mobility of clients, the limitation of wireless communications, and the resource constraints of mobile computing devices.

Physical Mobility

Mobility is the main characteristic that distinguishes the mobile environments from the traditional distributed environments. Table 1 summarizes and compares the characteristics of mobile and nonmobile environments. In traditional distributed environments, computers are stationary hosts. In mobile environments, mobile computers are continuously moving from one geographical location to another.

The mobility of a mobile host is a real-time movement. Therefore, it is affected by many environmental conditions. For example, a preplanned travel route of a mobile host can be changed because of traffic jams or weather conditions. If there is a mobile task[2] whose operations depend on the travel route of the mobile host, these operations can become invalid or extra support may being required. For example, a new route-map directory must be downloaded into the mobile host if the travel route is changed. Moreover, the movement of a mobile host can also depend on the objective of the mobile task. For example, an ambulance wants to arrive at the accident scene by selecting the shortest route within the fastest allowed speed, a bus must follow the strict timetable of a bus route, while a postman only wants to travel through each road once. During the movement, a mobile host can stop at some locations for some periods; therefore, the mobility of the mobile host includes both moving and nonmoving intervals.

Table 1. Nonmobile vs. mobile environments

	Nonmobile environments	Mobile environments
Computing hosts	Stationary sites	Mobile and nonmobile hosts
	Powerful computing capacity	Limited computing capacity of mobile hosts
	Reliable computing hosts	Less reliable computing hosts
Network connectivity	Wired and high-speed networks	Wireless, unstable, and lower-speed networks
	Reliable networks	Unreliable, error-prone, frequent and long disconnection periods

The location of a mobile host changes dynamically and frequently in accordance with the speed and the direction of the movement. The faster the mobile host moves, the more frequently the environment changes. The objective of mobile tasks may also specify the locations at which the mobile host must be in order to carry out the mobile tasks. For example, a computer technician must come to the customer locations to fix computer problems. A mobile support system must provide the utilities to manage the locations of mobile hosts (a demand which is not needed in a distributed environment). Changes of locations can cause changes in the operating environments of mobile hosts, for example, network addresses, communication protocols, mobile services, or location dependent data (Pitoura & Samaras, 1998; Ramampiaro, Wang, et al., 2003).

The behaviour of a mobile host describes the actual mobility states of the mobile host. While operating in mobile environments, the mobile host can be either in *stopped* or *moving* state. The two movement states are explained as follows:

- **Stopped:** A mobile host is said to be in the stopped state either when its movement velocity is zero, or when the location of the mobile host is not considered changing within a period of time. For example, a bus stops at a bus stop to pick up passengers, a salesman is selling products at a shopping centre, or two mobile hosts are moving close to each other.
- **Moving:** A mobile host is in the moving state either when its movement velocity has a value greater than zero, or when the location of the mobile host is considered changing over time. For example, a bus is moving along a road, or a salesperson travels to new places during the day. While in the moving state, the mobile host can continuously change its velocity and direction of movement.

Further, the behaviour of a mobile host can affect the tasks that are carried out by the mobile host, for example, a public transport vehicle needs to strictly follow a timetable. On the other hand, the movement of a mobile host may also be affected by the surrounding environmental conditions, for example, traffic jams. The movement behaviour of the mobile host demands additional support such as location management (Majumdar, Ramamritham, et al., 2003), and awareness of location dependent data (Dunham & Kumar 1998).

Mobile Devices

There are many types of mobile computing devices such as mobile phones, laptop computers, or personal digital assistants (PDAs). Mobile devices typically are smaller and lighter than stationary computers. Consequently, mobile computers have limited energy supply, limited storage capacity, and limited functionality compared to stationary computers. The characteristics of mobile computing devices are elaborated as follows.

- **Limited energy supply**: The operation of mobile computers heavily depends on the electrical power of batteries. This limited energy supply is one of the major disadvantages of mobile computing devices. The energy consumption of a mobile device depends on the power of electronic components installed on the mobile device (e.g., mobile network, CPU, display, etc.) and how these components are used (e.g., brightness of screen, reduced CPU speed, etc.). Moreover, the battery lifetime also depends on the number of applications and the application types that operate on the mobile devices (Flinn & Satyanarayanan, 1999). For example, a mobile phone can live up to 5 days, while a laptop may only be able to operate for several hours. Mobile applications that are executed on a mobile host may be interrupted or rescheduled if the mobile host is exhausting its energy supply.

- **Limited storage capacity**: The storage capacity of a mobile computer (i.e., hard disks or main memory) is much less than for a stationary computer and is harder to be expanded. Therefore, a mobile host may not be able to store the necessary data that is required for its operations in a disconnected mode (Pitoura & Samaras, 1998). Consequently, applications on mobile clients may be delayed due to data unavailability, or may require longer processing time due to frequent memory swapping operations.
- **Limited functionality**: The functionality of mobile devices is also limited in terms of the graphical user interface, the application functionalities, and the processing power. Therefore, a mobile host may be unable to perform some operations, or may require longer processing time to perform these operations. For example, the small screen of a mobile phones makes is hard or impossible to display full-size Web pages.

In the process of eliciting the requirements for a system to support mobile work, the mobile device characteristics must be taken into account. The limitations of the mobile device will naturally be a part of the nonfunctional requirement, but they will also affect the functional requirements, for example, be able to buffer information on device to save battery on network usage.

Wireless Communication

Mobile hosts communicate with other hosts via wireless networks. Compared to wired networks, wireless networks are characterized by lower bandwidth, unstable networks, disconnections, and adhoc connectivity (Schneiderman, 2002). The characteristics of wireless networks are described as follows.

- **Lower bandwidth:** The bandwidth of a wireless network is lower than for a wired network. A mobile network does not have the capacity of wired networks. For example, a mobile GSM network has bandwidth in the order of 10Kbps, a wireless local area network (WLAN) has bandwidths of 10Mbps; while gigabits (Gbps) are common in wired LANs (Schneiderman, 2002). Therefore, it may take a longer time for a mobile host to transfer the same amount of information via a wireless network than with a wired network.
- **Unstable networks:** A wireless network has high error rates, and the bandwidth of a wireless network is variable. Due to errors during data transmission, the same data packages may have to be retransmitted many times; thus, extra overhead in communication inducing higher costs. Due to the varying bandwidth, it is hard to estimate the time required to completely transmit a data package from/to a mobile host. These problems will affect the data availability of mobile hosts. As a result, the executions of applications at the mobile clients can be delayed or aborted.
- **Disconnections:** Wireless networks pose disconnection problems. Disconnection in communication may interrupt or delay the execution of transactions. Further, ongoing transactions may be aborted due to disconnections. Disconnection in communication is categorized according to two dimensions: disconnection period and disconnection rate.
    - **Disconnection period:** The disconnection period indicates how long a mobile host is disconnected. While being disconnected, the mobile host will not be able to communicate with other hosts for sharing data. Furthermore, the duration of a disconnected period of a mobile host is not always as planned; that is, it may be longer than expected. A mobile system must be able to continuously support mobile

clients while a mobile host is being disconnected by caching needed data in beforehand.

- **Disconnection rate:** The disconnection rate indicates how often the wireless communication is interrupted within a predefined unit of time. The execution of mobile applications may be affected when an interruption occurs. If the applications on the mobile host are executing collaborative operations with other applications on other mobile hosts, the collaborative activities may be suspended or aborted. To cope with such problems, a mobile system must be able to support mobile transactions to resume at or recover from previously interrupted points.

- **Adhoc connectivity:** The wireless network technologies introduce a new way to support direct and nearby communication among mobile hosts, called *any-to-any* or *mobile peer-to-peer* communication (Flinn & Satyanarayanan, 1999; Ratner, Reiher, et al., 2001). For example, two mobile hosts can directly share information via the support of Bluetooth or infrared technologies (Pradhan, Lawrence, et al., 2005). The characteristics of this peer-to-peer communication are: unstructured (i.e., adhoc), short-range, and mobility dependent (Ratner, Reiher, et al., 2001; Le & Nygård, 2005).

The issues high-lighted above concerning wireless communication is mainly related to nonfunctional requirements, but some aspects will also affect the functionality in the system like transaction support to handle disconnections.

Operations of Mobile Clients

The operational behaviour of mobile hosts depends on the availability of mobile resources such as network connectivity and battery energy. We distinguish two operational modes for mobile hosts in mobile environments: *interactional and isolation*. These operational modes of mobile hosts are explained as follows:

- **Interactional.** When a mobile host is sharing data with other hosts, it is said to be in the interactional mode. The two essential prerequisite conditions for the interactional mode are: (1) the mobile host is operational, and (2) network connectivity is available. It is not necessary that the mobile host connects to other hosts all the time. This may help the mobile host to save battery energy and reduce the communication cost. However, in the interactional mode, communication channels between the mobile host and other hosts must always be available and present whenever needed.
- **Isolation.** When a communication channel between a mobile host and other hosts is not available, the mobile host is disconnected from other hosts and is said to be in the isolation mode. There are many factors that contribute to disconnections of a mobile host; for example, the mobile host moves out of the wireless communication range, network services are not available, or the mobile host is running out of energy.

The behaviour of mobile hosts also illustrates the correlations among the different characteristics of mobile environments. Disconnection in communication may be the result of the movement of mobile hosts or the limitation of mobile resources. When mobile hosts communicate with others via short-range wireless network technologies, for example, infrared, Bluetooth or wireless LAN, the communication will be disconnected if the mobile hosts move outside the communication range. Mobile hosts may be disconnected more frequently and for shorter periods, that is, seconds or minutes, when they move in and out of the shadows of physical obstructions such as high buildings. The disconnection periods may also be longer, that is, hours or days, when mobile hosts stay in some locations within which the wireless network service is not available. Further, a mobile host may volunteer to disconnect if its supplied energy is running out.

Related Work

Scenario analysis is used in requirement engineering (Sutcliffe 1998), and thus the use of scenarios is not novel. The novel part is the focus on mobility issues in the requirement engineering process. Thus, this section will outline research that is related to characterisation of mobile scenarios that support development, that is, requirement analysis and design, of mobile applications. Many research papers on mobile work have proposed a support system for specific mobile work scenarios, resulting in the development of tailor-made systems (Sindhgatta & Thonse, 2005; Florio & Deconinck, 2002). Other papers (Forman & Zahorjan, 1994; Satyanarayanan, 1996; Zimmerman, 1999) have attempted to give an overview of the characteristics of mobile computing based on state-of-the-art technology. However, to our knowledge, there is no other similar characterisation framework for mobile works.

Satyanarayanan (1996) identifies four constraints of mobile computing concerned with limited resources, physical security (e.g., theft), communication and durability issues. Another approach is proposed by Forman and Zahorjan (1994) who examine three basic features of mobile computing including wireless communication, mobility and portability. These two approaches provide different ways of addressing mobility issues. The former focuses on connectivity issues, while the latter deals with Quality of Service (QoS), such as network bandwidth and device durability. There are several disadvantages of these proposals. First, the strong connection among these features of mobile computing has not been addressed. Second, how these features impact the operation of mobile clients, that is, mobile works, has not been discussed.

Rakotonirainy (1999) discusses current and future technologies (e.g., CORBA and mobile IP) adaptable to mobile computing environments. For this, he presents a scenario revealing the limitations of the current technologies. Although characteristics of mobile work may be derived from this approach, it does not provide a comprehensive framework for characterising mobile work environments.

Kangas and Kinnunen (2005) have argued and demonstrated that the most important aspect of the design process for mobile applications is the usability of the final products. This means that important features such as applied technologies and contextual design must be taken into consideration (otherwise, the developers may not know enough about the real needs or expectations of the end clients). Maccari (1999) also presents an example of requirement engineering challenges in the mobile telephone industry due to the complex mobile phone architecture and its corresponding performance. The author argues that requirement engineering for mobile telephones is a collaborative task and has to cope with many issues, such as protocols and technology standards. Further, the limitations of wireless devices, such as network connectivity and speed, imply important challenges that developers have to deal with. However, the author has not shown how developers can be supported to be aware of these factors.

Zimmerman (1999) suggests a MOBILE framework for determining when mobile computing technology should be used to solve challenges related to mobility. This framework focuses on current technology and software development trends in mobile computing. Further, some common scenarios are discussed, including news reporting and hotel operations. The framework provides a useful overview of support needed for specific mobile environments. However, the framework does not provide any guidelines for how to analyse or design systems for mobile support.

Designing applications for mobile platforms is more challenging than traditional software design (Holtzblatt, 2005). Holtzblatt presents a contextual design method, where customer centring plays a major role, for mobile applications. In this customer-centred design methodology, the users' situation, that is, the tasks, operating conditions, and device capacities, must be taken into account. However, this method is mainly focusing on the mobile devices, while other challenging factors like client mobility or network issues have not been fully addressed.

Raptis, Tselios, et al. (2005) have proposed a context-based design framework for mobile applications. In this framework, context is defined in four dimensions: system, infrastructure, domain and physical context. The authors have discussed in detail how such context factors impact the design decisions of mobile applications. Furthermore, the paper addresses how developers should take into consideration such context information. The disadvantage of the proposal is that the context is too coarse-grained to be useful. For example, the clients are either fixed or mobile. In our characterisation framework, we apply more fine-grained levels for the mobility of clients, that is, our framework ought to be more suitable for mobile application development.

The related work discussed above mainly focuses on the technical aspects of mobile computing. Our framework investigates mobile computing from another point of view, by focusing on the mobile scenarios to be supported and the identification of various system complexities introduced by the scenarios. In addition, our framework focuses on software support for mobile works and identifies issues that are not directly related to mobility, like process and transactional infrastructure. This is especially useful in design and development of systems for mobile work environment.

MOWAHS REQUIREMENT ANALYSIS FRAMEWORK

This section describes the MOWAHS requirement analysis framework. The starting point for using this framework is a mobile scenario. Using the mobile scenario, we can derive requirements based on an analysis of the tasks and roles described by the scenario. Identification and specification of requirement is useful with respect to the design and development of systems to support for mobile work.

The first question is, however, what minimum knowledge is required, or how much knowledge is expected to use the framework. To answer this question, the underlying assumption is that a user of the framework must have basic knowledge of software engineering to be able to use the MOWAHS framework to derive requirements. This includes knowledge of all relevant phases in a software development life cycle (e.g., the waterfall model, the spiral model, RUP, etc.). Knowledge of UML would also be useful.

The second question concerns the completeness of the scenarios. What is lacking when deriving requirements from the scenarios? To be able to produce a complete requirement specification, scenarios that cover all relevant aspects (also the nonmobile parts) are needed. However, this cannot be guaranteed with our framework due to following:

- The scenario description may be subjective and could be dependent on the person that performs the description; and
- Our focus is mainly on mobility; thus the framework will normally not include aspects not related to mobility.

Further, the produced requirements are not formal; thus, two developers will rarely produce identical requirements. However, this is not the ultimate goal of the framework. Rather, it emphasises on highlighting different aspects of the scenario to get as complete requirements as possible (with focus on mobility aspects). The framework is, thus, meant to act as a guideline for developers to extract requirements from scenarios. This means that the main contributions relate to the process of producing requirements and comprehensive checklists useful for developers to identifying requirements.

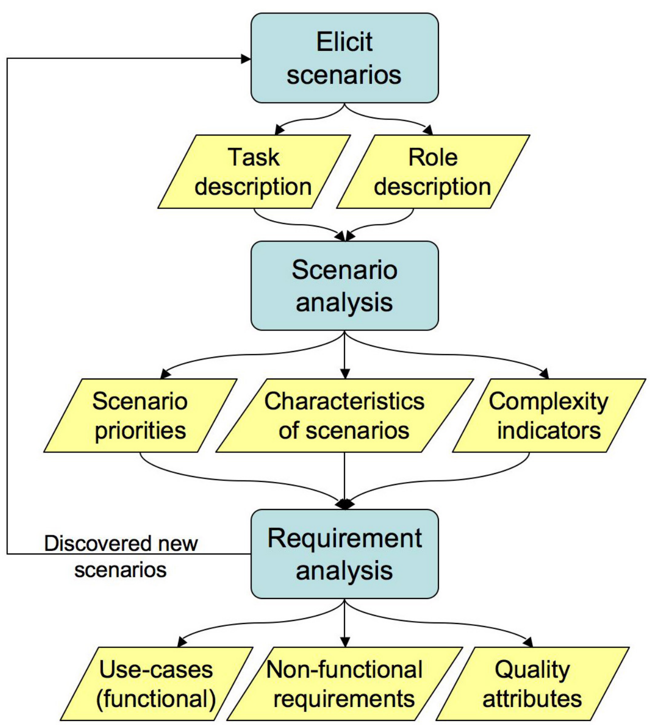
Before the MOWAHS requirement analysis framework is described in detail, a scenario that will be used as an example for the rest of the chapter is introduced to illustrate how the framework can be applied to a scenario.

The scenario is based on experiences from an IT-support organisation that wanted to look into the possibilities of introducing mobile clients to its existing Web-based system to support its daily work. A typical working day for the IT-support organisation consists of buying new hardware and software, installing new hardware and software, updating hardware and software, and assisting users with various problems concerning hardware and software. A nonmobile system was used to manage incoming requests form users sent by e-mail or filled out using Web forms. The system supported assignment of incoming tasks to various persons within the organisation. The clients of the system were running on a Web browser on a standard PC, and the IT-staff could read through their assigned tasks, look at the task history, change states, write reports and so forth. The main goal of the Web-based system was to maintain information about incoming requests during the lifetime of a specific task. In the following sections, first the theory of the requirement analysis method will be presented followed by an example of how the method can be used illustrated with the scenario described in this paragraph.

Requirement Analysis Process

The high-level process of our framework consists of three main steps, as shown in Figure 1.

Figure 1. High-level process



In the *first* step, *elicit scenarios*, the mobile workers' daily routines are described. This description will typically involve several roles and result in a description of both roles and tasks of the mobile workers.

The *second* step, *scenario analysis*, goes through the tasks described in the first step using a set of predefined mobile task characteristics. The results of this step are priorities of the tasks from the first step, characteristics of all tasks, and a set of computed complexity indicators.

The *third* and last step, *requirement analysis*, uses the results from the two previous steps to produce nonfunctional and functional requirements of the end-system. The following sections describe each step of our method by introducing the main concepts followed by an example of practical usage based on the scenario described in beginning of this section.

Step 1: Elicit Scenarios

The elicitation step can be carried out using various approaches depending on the type of work for which to develop a system and the type of company or organisation. One approach is to interview various mobile workers and the people working with back-end support. Another approach is to observe the mobile workers over time and then carry out interviews about daily routines. In this phase, it is important to identify and describe all roles involved in the mobile work, as well as people working with economy, inventory, project planning, customer service and so forth.

In our framework, a scenario contains a collection of mobile work task descriptions and role descriptions. A scenario is given a name along with a short description of the theme and process goal. The roles are described by a name and a short description of responsibilities.

The next step is to describe all tasks part of the scenario along with assignment of responsible roles to each task. A task is described by name, description, responsible role, precondition to start the task, postcondition to complete the task, and dependencies. The pre- and post-conditions are divided into three parts:

- **Location:** The worker must be in a specific place before or after completion of a task,
- **Resources:** Resources that must be in place before or will be available after completion of the task, and
- **Tasks:** Tasks that must be completed before starting the current task or tasks that can start after completing the current task.

The *tasks'* pre- and post-conditions define the horizontal relationships among tasks. The dependency part of the task description is used to identify dependencies on other roles and tasks in the scenario. This includes task composition that describes vertical relationships among tasks (hierarchical process model) and instantiation of roles (number of people involved in the task). A template for a task description is shown in Table 2.

Some parts of the task template above might not be applicable and will then be filled out with NA (Not Applicable).

Example:
Various persons from the IT-support organisation introduced in the beginning of this section are interviewed to elicit their daily routines. The first step then is to describe the scenario as indicated above. (see Box 1)

From the description above, it is obvious that the role *support engineer* is the only one involved in mobile tasks. Thus, the rest of the example will focus on the support engineer. For this role, the following tasks have been identified: *set up new computer*, *upgrade computer*, and *assist user*. In this example, we will focus on the task *assist user*. In Table 3, a task description of the task *assist user* is presented.

The task description is based on information acquired from interviewing the IT-staff. Note that

Table 2. Task description template

Task name	<unique task name within a scenario>
Priority	<from very low (1) to very high (5)>
Responsibility	<responsible role>
Description	<short description of the task>
Precondition	*Location*: <location requirement to start the current task> *Resources*: <resource requirement(s) to start the current task> *Tasks*: <task(s) to be completed to start the current task>
Postcondition	*Location*: <location when completing the current task> *Resources*: <resource(s) available after completing the current task> *Tasks*: <task(s) allowed to be started after completing the current task>
Dependencies	*Other roles*: <other role(s) involved in the task> *Agents*: <number of people involved in the task> *Parent task*: <relationship to parent task> *Child tasks*: <relationships to child tasks>

Box1.

Scenario:
Name: Mobile IT-support
Description: Describes the mobile aspects of IT-support work processes

Roles:
Name: Support manager
Description: Responsible for managing the IT-support including assigning tasks to IT-support staff

Name: Support desk
Description: Responsible for registration of user inquires and solving problems that can be solved remotely using a computer

Name: Support engineer
Description: Responsible for solving problems concerning hardware and software at specific locations (computer labs, offices, etc.)

Table 3. Task description–Assist user

Task name	**Assist user**
Priority	Very high (5)
Responsibility	Support engineer
Description	Help users to solve computer problems, like malfunctioning mouse, keyboard, software, etc.
Precondition	*Location*: Must be at user's location (office, lab) *Resources*: Information or equipment dependent on user problem *Tasks*: User has reported problem, task assigned to support engineer
Postcondition	*Location*: NA *Resources*: Solution report (electronic), changed task state *Tasks*: NA
Dependencies	*Roles*: None *Agents*: 1 *Parent task*: NA *Child tasks*: Find user/equipment, show task information, solve problem, report solution, change task state

in order to specify the child tasks, the task *assist user* had to be decomposed into subtasks that describe necessary steps to carry out the task.

Step 2: Scenario Analysis

The scenario analysis can be used in several ways. The analysis can be used as a checklist for issues one should consider when making computer support for mobile work. Further, it provides a more careful examination of the requirements for making a system to support mobile scenarios. This examination will produce requirement indicators to identify, for example, complex parts of the system, type of client device, type of network, and services needed. In addition, the framework can be used to identify interfaces between a mobile client and back-end systems and indicate information requirements for the mobile worker.

Table 4. MOWAHS characteristics

Characteristic	Possible values	Description
General		
G1. Decomposable	(1 No, 5 Yes)	Composed of subtasks?
G2. Part of sequence	(1 No, 5 Yes)	Order dependencies with other tasks?
G3. Preplanned	(1 Planned, 3 Partial, 5 Adhoc)	To what degree planned in beforehand?
G4. Data synchronisation	(1 Never, 3 After end, 5 Duration)	When update data with other tasks?
G5. Data exchange rate	(1 Never, 3 Once, 5 Many)	How often will the task exchange data with other tasks within its lifetime?
Information		
I1. Info content	(1 Text, 3 Graphics, 5 Multimedia)	Complexity of info required/produced?
I2. Info streaming	(1 NA, 3 Discrete, 5 Continuous)	Does the task require streaming of data?
I3. Freshness of data required	(1 NA, 2 Day, 3 Hour, 4 Min, 5 Real-time)	How fresh must data received from a server be to execute the task?
I4. Freshness of data produced	(1 NA, 2 Day, 3 Hour, 4 Min, 5 Real-time)	How fresh must data received by a server and produced by the task be?
I5. Data transmission	(1 NA, 2 Slow, 3 Medium, 4 Fast, 5 Very fast)	What is the expected transmission speed required to execute the task?
Location		
L1. Location dependent	(1 No, 5 Yes)	Must be executed at a specific location?
L2. Require services at location	(1 No, 5 Yes)	Require electronic services at the location?
L3. Produce services at location	(1 No, 5 Yes)	Produce electronic services at the location?
L4. Location report	(1 No, 5 Yes)	Must report the current location to a server?
L5. Route constraint	(1 No, 5 Yes)	Must follow a specific route when moving?
Time		
T1. Event-triggered	(1 No, 3 Partial, 5 Yes)	Is the task triggered by an event?
T2. Time constraint	(1 No, 3 Partial, 5 Yes)	Must the task be executed at a specific time?
T3. Temporal coordination	(1 No, 3 Partial, 5 Yes)	Must the task be timed with other tasks?
T4. Task resumption	(1 No, 3 Partial, 5 Yes)	Can the task be halted for later to be resumed from where it left off without a restart?
T5. Task lifetime	(1 Sec, 2 Min, 3 Hour, 4 Day, 5 Week)	What is the expected lifetime?

The characteristics are divided into four main categories:

- **General characteristics** that specify task structure and organization,
- **Information characteristics** that specify how information and data is used by a task,
- **Location characteristics** that specify how a task depends on a geographical location, and
- **Time characteristics** that specify the temporal properties of a task.

For each of the four main categories, there are five questions related to the category. Thus, there are 20 questions to be answered for every task described in Step 1: Elicit scenarios. Table 2 shows the 20 characteristics used in the scenario analysis of tasks described in a mobile scenario.

To carry out the scenario analysis, the following procedure should be performed.

- For each role; list the tasks
- For each task:
  a) Assign task priority (1-5; 1 is least important and 5 is most important)
  b) Characterise the task using the characterisation questions (as shown in Table 4 above)
  c) Calculate the requirement indicators for the task

To measure the different characteristics, an ordinal scale (1-5) is used. High values indicate

Table 5. Complexity indicators

Indicator	Characteristic	Explanation
General Task Indicator (GTI)	Average of G1-G5	A high GTI score indicates that the underlying process and transactional infrastructure (e.g., workflow system) must be advanced.
Information Complexity Indicator (ICI)	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)	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)	Average of T1-T5	A high TCI score indicates that time management and coordination of tasks, and advanced transactional support might be necessary. In addition, the TCI also indicates the level of performance and availability required.
Network Connectivity Indicator (NCI)	Average of G3, G4, G5, L4 and T1	The NCI 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 nonfunctional requirements for the system such as reliability and latency.
Network Speed Indicator (NSI)	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)	Average of I5, L1, L2, L3 and T5	A high ECI score means that it is likely that the mobile client device to complete the task will consume much energy.
Transaction Support Indicator (TSI)	Average of G3, G4, G5, T1, T4 and T5	The 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)	Average L4 and L5	The MI 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)	Average of all characteristics of a task	The TC 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.

higher complexity in terms of system requirements, while low values indicate lower complexity. Many of the characteristics do not use the full scale, but only the values 1, 3 and 5, or simply 1 and 5 to get a uniform representation of extreme values. We are aware that it is mathematically fuzzy to calculate average values on the ordinal scale, but we have found the usage of extreme values valuable for discovering complexity in specific areas of the scenario. From the scores of the 20 characteristics in the framework, indicators can be computed that help analysing the mobile scenario and thus prioritising and extracting nonfunctional and functional requirements (see Step 3: Requirement analysis). The indicators are an average of selected characteristics related to certain complexity aspects. The indicators are described in Table 5.

Because the indicators are averages of the characteristics, the provided values can be used for suggesting complexity for certain properties. The actual numbers are used to compare the complexity of different tasks to identify which tasks that require more attention in the following requirement analysis, and the design and implementation of the end-system. In addition, the values of the indicators will identify particular areas concerned with nonfunctional requirements that must be carefully managed, for example, network connectivity and speed, energy consumption, or location management.

Example:
In this step of the process, the focus is on going through the scenario. For the IT-support scenario, we had to go through a characterisation of all the tasks for the roles *support manager, support desk* and *support engineer*. Here we will focus on the role *support engineer*. To illustrate how to carry out the characterisation, we will demonstrate how to determine the score for three characteristics, namely *G1 Decomposable*, *L1 Location dependent* and *T5 Task lifetime* for the task *assist user*.

G1: *Is the task assist user composed of subtasks?*
Answer: If we look at dependencies/child tasks in the task description, we clearly see that this task is decomposable.
Score: 5 (Yes).

L1: *Must the task assist user be executed at a specific location?*
Answer: If we look at the precondition/location in the task description, we see that the *support engineer* must be at the location of the user.
Score: 5 (Yes).

T5: *What is the expected lifetime of the task assist user?*
Answer: In this case, the task description does not say anything about the duration of a task. To get this information, the IT-support staff may be asked about the expected time for assisting a user.
Score: 3 (Hour).

We will now use the results from characterising the three tasks *set up new computer*, *upgrade computer* and *assist user* for the role *support engineer* to calculate the requirement indicators (see Table 5). The results of the calculations of the requirement indicators are shown in Table 6.

The *assist user* task has the highest scores or equally high scores for all the indicators compared to the other two tasks. This means that this task is probably the most complex with respect to mobile client, mobile network, and back-end support. This also means that it is the task that is most important to analyse more in detail because it will likely cause most challenges when designing the system in terms of mobile support for both nonfunctional and functional requirements.

Step 3: Requirement Analysis

The final step of the process is to identify and structure the nonfunctional and functional requirements of the end-system. It relies on results

Table 6. Requirement indicator scores from the IT-support scenario

Requirement Indicator	Set up new computer	Upgrade computer	Assist user
General Task Indicator (GTI)	2,2	2,6	3,4
Information Complexity Indicator (ICI)	2,4	3,4	3,4
Location Complexity Indicator (LCI)	2,6	2,6	3,4
Time Complexity Indicator (TCI)	3,0	3,0	4,0
Network Connectivity Indicator (NCI)	2,2	2,2	5,0
Network Speed Indicator (NSI)	2,0	3,7	3,7
Energy Consumption Indicator (ECI)	2,5	2,5	3,3
Transaction Support Indicator (TSI)	2,3	2,3	4,5
Mobility Indicator (MI)	3,0	3,0	3,0
Task Complexity (TC)	2,6	2,9	3,6

from the two previous steps. The result of this step is a collection of nonfunctional and functional requirements. Parts of the functional requirements are found by decomposing tasks from the scenario description. The decomposed tasks can be used as an input for a new iteration (back to step 1), where these tasks are further described, characterised and analysed.

The ISO9126 taxonomy (International Organization for Standarization, 2001) describes six main quality characteristics of a software system: functionality, maintainability, portability, usability, reliability and efficiency. Some nonfunctional requirements in mobile systems are difficult to directly map onto ISO9126. The nature of mobile systems is very dependent on the execution environment, for example, mobile device capabilities, network infrastructure and availability, power availability, and the physical environment. The execution environment affects how the system architecture (both hardware and software) is designed to operate in a mobile setting with nondeterministic properties. For example, the mobile device capabilities relate to functionality, portability, usability, reliability and efficiency in ISO9126. It is, however, difficult to map the mobility properties onto the taxonomy.

Nonfunctional requirements: The mobile device
Table 7 shows how device requirements are addressed in the MOWAHS analysis framework. These nonfunctional requirements need to be considered in all mobile systems. The nonfunctional requirements for a mobile device is determined by going through all the requirements in the table one by one, looking at the associated characteristics and indicators for the given requirement. If the characteristics and indicators related to a requirement indicate high complexity, this should be translated into, for example, high CPU speed, or high memory usage.

Some of the issues will require adaptation strategies with respect to hardware and software heterogeneity. Others related to informational and computational requirements force one to consider computational distribution and information mobility not normally specified in nonmobile systems. Issues like load balancing should also be considered to off-load computations (using cyber foraging) on a mobile device to a server to save battery, CPU, memory and communication costs. Note that the characteristics and indicators must be considered as help on the way to specifying the nonfunctional requirements.

Table 7. Framework mapping onto device requirements

Requirement	Issues	Characteristics	Indicators
CPU	Speed	I1, I2, I5	ICI, NSI, ECI
Memory	Size and speed (primary, secondary)	I1, I2, I3, I4, I5	ICI
Graphics hardware	3D, video, speed	I1	None
Audio	Speakers, earphone, loudness	L1, L2, L3, T2, T3	LCI, TCI
Notification	Vibration, alarm, ring tone, low energy	L1, L2, L3, T2, T3	LCI, TCI, ECI
Network support	Type (BT, WiFi, 3G, GPRS), handover, session mobility	G4, G5, I2, I5, T2	NCI, NSI
Screen	Resolution, colour, lightness	I1	None
Battery capacity	Lifetime wrt. CPU, screen, communication, graphics, audio	I5, L1, L2, L3, T5	ECI
Input device	Modal / Nonmodal	None	None
Location-detection	GPS, network detection	L1, L2, L3, L4, L5	LCI
Weight and size	Portability	I1, I5, L1, L2, L3, T5	LCI, ECI

Example:
Consider again the *assist user* task (because this was the task with the highest complexity scores) from the IT-support scenario and the following nonfunctional requirements for the mobile device: *memory*, *network support* and *location-detection*.

Requirement: *Memory*
Indicator: ICI (Information Complexity Indicator)
Reasoning: If both characteristics and indicators are available for a requirement, indicators should be used (because indicators are computed from the characteristics). The ICI score for the *assist user* task was 3,4 (see Table 6) and just above average for a task. This means the expected memory usage of the application is about average, and we do not need a mobile device with the highest requirements in terms of memory.

Requirement: *Network support*
Indicators: NCI (Network Connectivity Indicator), NSI (Network Speed Indicator)
Reasoning: The scores of the *assist user* task for NCI and NSI were (see Table 6) 5,0 and 3,7 respectively. These results indicate the highest value of required connectivity for the device. Thus, it is important that the wireless networks supported by the mobile device always are available in various locations. If the *support engineer* only should work within office buildings where WiFi is available, it is sufficient that the mobile device has support for WiFi. However, if there is a mixture of networks available that varies from building to building where the IT-support organisation is responsible, it is important that the mobile device supports various networks and automatic handover processes from one wireless network to another.

Requirement: *Location-detection*
Indicator: LCI (Location Complexity Indicator)
Reasoning: The LCI score of the task *assist user* was 3,4 and a bit above average. This means that location-aware services should be considered. It is also possible to look at some of the characteristics to clarify this issue for example L1 (Is the task location dependent? - see Table 4). From the precondition of the task description in Table 3, it is obvious that the task *assist user* is location dependent. This means that the mobile device must support some kind of location detection, for example, GPS, base station triangulation, and so forth.

Nonfunctional requirements: The network

Table 8 shows how network requirements are addressed in the MOWAHS analysis framework. A mobile system must deal with issues related to network availability and quality of service (QoS). Knowledge of network coverage and QoS can be used to predict which information should be managed and stored on the mobile device and which information should be managed and stored on a back-end system.

In addition to the nonfunctional requirement described above, other issues like device heterogeneity and client software requirements (like portability) should be addressed (but is not covered in this chapter). Further, nonfunctional requirements for the back-end systems must be addressed and analysed.

An example of how network requirements are elicited is not presented in this chapter, as it is carried out in the same way as for mobile device requirements described above.

Functional requirements: Starting from the role and task descriptions

A starting point for capturing the functional requirements using the MOWAHS analysis framework is the scenario descriptions (role and task descriptions). To carry out the requirement elicitation, all the tasks in the scenario must be analysed with respect to system boundaries; that is, establish use-cases of the end-system. In this process, the tasks are decomposed into a set of steps that describe the basic functionality to be provided by the 1) mobile client or 2) the back-end system. The roles will identify stakeholders and use-case actors. The requirement analysis can be used in combination with the scenario-based requirements analysis method described in Sutcliffe (1998).

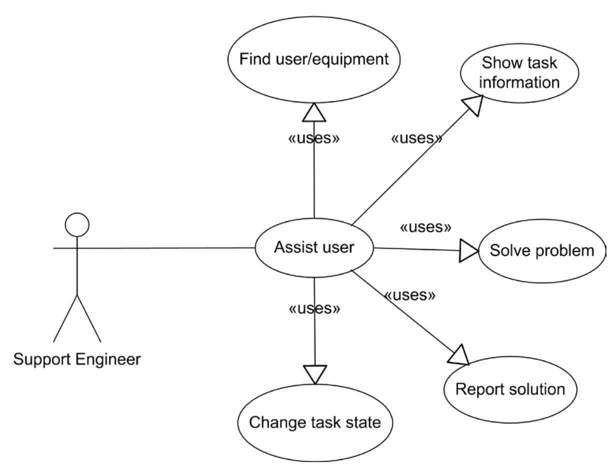
Example:

Consider the task *assist user* from the IT-support scenario: This task is decomposed into the

Table 8. Framework mapping onto network requirements

Requirement	Issues	Characteristics	Indicators
Bandwidth	Transfer speed	G5, I3, I4, I5	NSI
Latency	Performance, delay	G4, I2, I3, I4 (L2, L3), T3	ICI
Availability	Coverage, connectivity, reliability	G4, L1, L2, L3, L4, L5, T3	LCI, NCI
Cost	Transfer cost	G5, I1, I2, I5, L2, T5	NSI

Figure 2. Use-case diagram of the task assist user



following child tasks: *find user/equipment*, *show task information*, *solve problem*, *report solution*, and *change task state*. These child tasks can be directly considered as use-cases. The *support engineering* will thus be the use-case actor. Use cases derived from the task *assist user* are shown in Figure 2.

The requirement analysis process can be further elaborated by analysing the identified use-cases. In use-case *find user/equipment*, the following functional requirements can be derived:

- **FR1:** The system needs to store or update information about users and equipment with unique ID;
- **FR2:** The system needs to store or update location information related to user or equipment ID;
- **FR3:** The system needs to provide functionality to locate user or equipment;
- **FR4:** The system needs functionality to map user or equipment onto problem task;
- **FR5:** The system needs to provide functionality to show location of user or equipment; and
- **FR6:** The system needs to provide functionality to show route information to user or equipment.

Functional requirements: Starting from the requirement indicators

Another way of using the MOWAHS analysis framework to identify functional requirements is to use the values of the requirement indicators. The indicators themselves cannot directly be used to derive functional requirements, but the indicators may imply areas of functionality that must be looked into or supported to provide system support for the scenario being analysed. Table 9 shows a mapping between indicators and functional requirements. To use the table, one should check if some of the indicators in the table have high scores. If this is the case, the functional requirement linked to these indicators ought to be considered in the end-system.

Example:

Again we go back to our IT-support scenario focusing on the task *assist user* and the two functional requirements areas *network management* and *transaction management*.

Requirement: *Network management*
Indicator: NCI (Network Connectivity Indicator)
Reasoning: The NCI score for the *assist user* task has the highest value 5,0 (see Table 6). This means that connectivity is very important for the end-system. In terms of functional requirements, this might include data buffering management (in case of a network failure), automatic handover between various networks without loosing data, caching of data and so forth.

Table 9. Framework mapping into functional requirements

Requirement	Issues	Indicators
Workflow infrastructure	Provide a process support infrastructure	GTI
Multimedia management	Provide support for advanced GUI, audio/video playback, multimedia streaming	ICI
Context management	Provide support for location-aware/context-aware information and services	LCI
Time management	Provide support for task arbitration and coordination, task scheduling and task planning	TCI
Network management	Provide support for network services required	NCI
Storage management	Provide support for storage management (cyber foraging vs. local storage vs. communication)	NSI/ECI
Transaction management	Provide support for transaction management and infrastructure	TSI

Requirement: *Transaction management*
Indicator: TSI (Transaction Support Indicator)
Reasoning: The TSI score for the *assist user* task has a very high value 4,5 (see Table 6). This indicates that the end-system most likely should provide support for transactions beyond standard ACID (Ramampiaro & Nygård, 2004), for example, support for long transactions, updating or viewing the same data on several mobile devices as well as the back-end system concurrently.

CONCLUSION

This chapter has presented a requirement analysis framework useful for analysing mobile work scenarios to implement systems supporting mobile users. It describes the process with steps starting with a vague description of a scenario, and then detailing the scenario description through definition of roles and tasks. These tasks are then analysed according to 20 characteristics, and 10 complexity indicators are calculated. The final step elicits nonfunctional and functional requirements based on knowledge from the previous steps.

Our discussion shows the limitations of our framework with respect to coverage of nonmobile types of requirements. The framework might be used as a tool in any software process life cycle. We believe it provides important input and knowledge in the development of systems to support mobile users. Further, although our framework has been tested against real-world scenarios, and that it has been implemented, there are room for other improvements. As such, our future work includes validating the MOWAHS framework within mobile software system design environments. This will allow us to reveal all-important aspects that we have to take into account, in addition to those our framework already covers.

FUTURE RESEARCH DIRECTIONS

In the recent years, the fact that mobile software systems have become larger and more complex has been widely recognised. Such systems increasingly involve interaction between various mobile devices and interfaces, various servers, and even services offered by legacy systems. Thus, it becomes more important to carefully analyse requirements for such systems, especially requirements related to mobile environments. Having this in mind, we believe that our requirement analysis framework will be even more useful for future design and development of mobile systems. In addition, we recognise the increasing need for computer support for mobile occupations. Despite the fact that our framework does not cover all aspects of requirement analysis of a software system, we believe it plays an important role for highlighting mobile issues.

Another recent observed trend is that sensors have become smaller (e.g., use of nanotechnology), cheaper and widespread (Satyanarayanan, 2003). Sensors combined with the rising numbers of activators (or triggers), give new opportunities and challenges for new types of mobile applications. Such applications will utilise the state of the environment around the user to provide services that are relevant to current task, time, location and situation. Our framework can be used to analyse parts of emerging scenarios (e.g., issues related to location and time). The framework can be extended and adapted to also handle new aspects and issues that arise with new technologies.

Currently, service-oriented computing and service-oriented architectures are considered as suitable approaches for designing and developing mobile applications and services. Our mobile characterisation framework can be useful in terms of providing context information of mobile clients. This can enhance the availability and reliability of the provided mobile services.

The proposed process can be regarded as agile by the use of scenarios, and iterative and incremental analysis. Because tasks have been given priorities, the priorities can be used to manage and plan implementation and deployment of functionality in the system. Our process is based on collaboration between the stakeholders, but do not explicitly define this collaboration. The method can be used as part of other software processes like eXtreme Programming (XP) and RUP.

REFERENCES

Dunham, M. H., & Kumar, V. (1998). Location dependent data and its management in mobile databases. In *Proceedings of the Database and Expert Systems Applications (DEXA) Workshop*.

Flinn, J., & Satyanarayanan, M. (1999). Energy-aware adaptation for mobile applications. In *Proceedings of the Symposium on Operating Systems Principles (SOSP).*

Florio, V. D. & Deconinck, G. (2002). On some key requirements of mobile application software. In *Proceedings of the International Conference on Engineering of Computer-based Systems (ECBS 2002).*

Forman, G. H., & Zahorjan, J. (1994). The challenges of mobile computing. *Computer, 27*(4), 38-47.

Holtzblatt, K. (2005). Customer-centered design for mobile applications. *Personal and Ubiquitous Computing, 9*(4), 227-237.

IEEE IEEE-Std 830 1993 IEEE Recommended Practice for Software Requirements Specifications.

International Organization for Standarization. (2001). *ISO/IEC Standard 9126: Software engineering--product quality*, part 1, ISO.

Kangas, E., & Kinnunen, T. (2005). Applying user-centered design to mobile application development. *Communications of the ACM (CACM), 48*(7), 55-59.

Le, H. N., & Nygård, M. (2005). A mobile affiliation model for supporting mobile collaborative work. In *Proceedings of the Workshop on Ubiquitous Mobile Information and Collaboration Systems (UMICS*).

Maccari, A. (1999). The challenges of requirements engineering in mobile telephones industry. In *Proceedings of the 10th International Conference and Workshop on Database and Expert Systems Applications (DEXA '99*). Springer-Verlag.

Majumdar, R. K., & Ramamritham, K., et al. (2003). Adaptive location management in mobile environments. In *Proceedings of the International Conference on Mobile Data Management (MDM 2003).* Melbourne, Australia, ACM Press.

Pitoura, E., & Samaras, G. (1998). *Data management for mobile computing*. Kluwer Academic.

Pradhan, S., & Lawrence, E., et al. (2005). Bluetooth as an enabling technology in mobile transactions. In *Proceedings of the International Symposium on Information Technology: Coding and Computing (ITCC 2005).*

Rakotonirainy, A. (1999). Trends and future of mobile computing. In *Proceedings of the 10th IEEE International Workshop on Database and Expert Systems Applications*. Florence, Italy: IEEE CS Press.

Ramampiaro, H., & Nygård, M. (2004). CAGIS-Trans: Providing adaptable transactional support for cooperative work--an extended treatment. *Information Technology & Management (ITM) Journal, 5*(1-2), 23-64.

Ramampiaro, H., & Wang, A. I., et al. (2003). Requirement indicators derived from a mobile characterisation framework. In *Proceedings of the IASTED International Conference on Ap-*

plied Informatics (AI'2003). Innsbruck, Austria: Acta Press.

Raptis, D., & Tselios, N. K., et al. (2005). Context-based design of mobile applications for museums: A survey of existing practices. In *Proceedings of the 7th Conference on Human-computer Interaction with Mobile Devices and Services*. Salzburg, Austria: ACM Press.

Ratner, D., & Reiher, P. L., et al. (2001). Replication requirements in mobile environments. *Mobile Networks and Applications (MONET), 6*(6), 525-533.

Satyanarayanan, M. (1996). Fundamental challenges in mobile computing. In *Proceedings of the 15th Annual ACM Symposium on Principles of Distributed Computing*. Philadelphia, PA: ACM Press.

Satyanarayanan, M. (2003). Of smart dust and brilliant rocks. *IEEE Pervasive Computing, 4*(2), 2-4.

Schneiderman, R. (2002). *The mobile technology question and answer book: A survival guide for business managers*. American Management Association.

Sindhgatta, R., & Thonse, S. (2005). Functional and non-functional requirements specification for enterprise applications. *Product focused software process improvement (PROFES 2005)*. Berlin/Heidelberg: Springer-Verlag.

Sutcliffe, A. (1998). Scenario-based requirements analysis. *Requirements Engineering, 3*(1), 48-65.

Wang, A. I., & Sørensen, C.-F., et al. (2005). Using the MOWAHS characterisation framework for development of mobile work applications. *Product focused software process improvement (PROFES 2005)*. Berlin/Heidelberg: Springer-Verlag.

Zimmerman, J. B. (1999). *Mobile computing: Characteristics, business benefits, and the mobile framework* (Tech. Rep. No. INSS 690 CC). University of Maryland European Division.

ADDITIONAL READING

The following is a list of references we recommend for further reading.

Aoyama, M. (2005). Persona-and-scenario based requirements engineering for software embedded in digital consumer products. In *Proceedings of the 13th IEEE International Conference on Requirements Engineering (Re'05)*. IEEE Computer Society.

Bohn, J. (2006). Prototypical implementation of location-aware services based on super-distributed RFID tags. *Architecture of computing systems (ARCS 2006*). LNCS 3894. Springer-Verlag.

Burns, M. (2006). Utilizing scenarios for user testing of mobile social software. In *Proceedings of the CHI 2006 Workshop on Mobile Social Software*.

Chung, L., Nixon, B., Yu, E., & Mylopoulos, J. (2000). *Non-functional requirements in software engineering.* Boston: Kluwer Academic.

Dunham, M. H., Helal, A., & Balakrishnan, S. (1997). A mobile transaction model that captures both the data and movement behavior. *Mobile Networks and Applications, 2*(2), 149-162.

Følstad, A., & Rahlff, O. W. (2002). Basic user requirements for mobile work support systems—three easy steps. In *Proceedings of the ER/IFIP8.1 Workshop on Conceptual Modelling Approaches to Mobile Information Systems Development (MobIMod 2002).*

Goguen, J., & Linde, C. (1993, January 4-6). Techniques for requirements. Elicitation. In *Proceedings of the 1st IEEE International Symposium on Requirements Engineering (RE'93)*, San Diego, CA, USA, (pp. 152-164).

Gregoriades, A. (2005). Scenario-based assessment of nonfunctional requirements. *IEEE Transactions on Software Engineering, 31*(5), 392-409.

Haumer, P., Pohl, K., & Weidenhaupt, K. (1998). Requirements elicitation and validation with real world scenes. *Software Engineering, 24*(12), 1036-1054.

Lee, V., Schneider, H., & Schell, R. (2004). *Mobile applications: Architecture, design, and development*. HP Professional Books/Prentice Hall.

Lindgaard, G., Dillon, R., Trbovich, P., White, R., Fernandes, G., Lundahl, S., & Pinnamaneni, A. (2006). User needs analysis and requirements engineering: Theory and practice. *Interacting with Computers, 18(*1), 47-70.

Manaouer, A. B., Shekhar, A., & Liang, Z. Y. (2004). A generic framework for rapid application development of mobile Web services with dynamic workflow management. In *Proceedings of the 2004 IEEE International Conference on Services Computing (SCC'04).* IEEE CS Press.

Maiden, N., & Robertson, S. (2005). Developing use cases and scenarios in the requirements process. In *Proceedings of the 27th international Conference on Software Engineering (ICSE 2005*). ACM Press.

Maiden, N., Seyff, N., Grunbacher, P., Otojare, O., & Mitteregger, K. (2006). Making mobile requirements engineering tools usable and useful. In *Proceedings of 14th IEEE International Requirements Engineering Conference (RE'06),* (pp. 29-38). IEEE CS Press.

Mummert, L.B., Ebling, M.R., & Satyanarayan, M. (1995). Exploiting weak connectivity for mobile file access. In *Proceedings of the Fifteenth ACM Symposium on Operating Systems Principles.* ACM Press.

Nakamura, H. (2001). A development scenario for ubiquitous networks: Viewed from the evolution of IT Paradigms. *NRI Papers, 26*, 1-9.

Seyff, N., Grunbacher, P., Maiden, N., & Tosar, A. (2004). Requirements engineering tools go mobile. In *Proceedings of the 26th International Conference on Software Engineering*. IEEE CS Press.

Sommerville, I., & Sawyer, P. (2000). *Requirements engineering: A good practice guide*. John Wiley.

Sutcliffe, A. (1995). Scenario-based requirements analysis. *Requirements Engineering Journal, 3*(1), 48-65.

Yacoub, S. M., Cukic, B., & Ammar, H. H. (1999). Scenario-based reliability analysis of component-based software. In *Proceedings of the 10th International Symposium on Software Reliability Engineering (ISSRE 99*). IEEE CS Press.

END NOTES

[1] A mobile worker is defined as a person that has to change location/context to perform his/her work.

[2] A mobile task describes a working task of a person with a mobile occupation, like a traveling salesman.