REQUIREMENT INDICATORS FOR MOBILE WORK: THE MOWAHS APPROACH

Heri Ramampiaro, Alf Inge Wang, Carl-Fredrik Sørensen, Hien Nam Le and Mads Nygård
Dept. of Computer and Information Science
NTNU, NO-7491 Trondheim, Norway
E-mail: {heri, alfinge, calle, hien, mads}@mowahs.com

ABSTRACT

This paper describes eight requirement indicators derived from a mobile work characterisation framework (MWCF). The MWCF is a framework for analysing mobile scenarios (mobile tasks) in order to implement a system to support such scenarios. The requirement indicators are computed from the score of the MWCF. These are used to reveal the complexity of the different parts of a final mobile support system (software and hardware). Further, these indicators can be a help to prioritise the non-functional and functional requirements of the end-system. We have identified the following indicators: General Task Indicator, Information Complexity Indicator, Location Complexity Indicator, Time Complexity Indicator, Network Connectivity Indicator, Network Speed Indicator, Energy Consumption Indicator, Transactional Support Indicator and Mobility Indicator. In this paper, we apply these indicators on four different scenarios and discuss how they can be used in practice.

KEY WORDS

Requirement engineering, mobile requirements, mobile process system.

1 INTRODUCTION

The proliferation of network and wireless technology has increasingly influenced our way of performing work. People carry out their work while they are on move - e.g., on their way to work, on a trip or even while they are on vacation. Also, several companies have started to work as virtual organisations, where people collaborate over several locations and time zones. This means that these people are distributed, collaborative, mobile and require asynchronous technology that enables sharing of documents and work plans. However, the infrastructure and tools for carrying out projects in virtual organisations are still immature. We must deal with a heterogeneity of tools, equipment (laptops, PDAs, mobile phones) and work models. In addition, the mobility of devices and the partial lack of connectivity require regular synchronisation of such devices against stationary PCs (including servers). Hence, there are several challenges that must be faced.

Our MOWAHS framework[13] is aimed at revealing these challenges. It presents relevant characteristic elements of mobile work. From such characteristics, there is a gap between characteristics and requirements that must be bridged in order to provide appropriate technology that supports a specific type of mobile work, and defining the actual requirements. To deal with this, we have identified several indicators that are useful in defining the requirements working as a guideline for designating the needed technology. This paper presents these indicators and discusses their usefulness.

The rest of this paper is organised as follows. To put work in perspective, Section 2 discusses the relation of our work to existing frameworks and approaches. Section 3 summarises our mobile work characterisation framework based on [13]. Section 4 describes the requirement indicators that we have identified based on the mobile work characteristics. The use of these indicators are illustrated in more detail in Section 5. Finally, in Section 6 we discuss and conclude our paper.

2 RELATED WORK

Some research papers on mobile work (see below) have proposed systems to support specific mobile work scenarios, which has resulted in the development of tailor-made systems. In addition, there are papers that give an overview of characteristics of mobile computing focusing on state-of-the-art technology, and there are a few research papers that treat requirement engineering related to the development of applications for mobile environments. However, it has been difficult to find other work that can directly be compared to ours. Hence, to our knowledge, the approach presented in this paper is unique in that it is based on indicators derived from a specialised characterisation framework for mobile work.

Mead [8] and Leite [2] pointed out that there is a gap between research and practice in requirement engineering. In particular, they point out the lack of tools for new methods and technical support in requirement engineering. Further, Mead stresses the necessity of improving the management of visibility and control. However, although most of the ideas of the above papers are useful and can be applied to problems such as ours, none of them directly deals with mobile environments.

Maccari [7] presents an example of requirement engineering challenges in the mobile telephones industry due to the complexity of mobile phone architecture and performance. In addition, requirement engineering in terms of mobile telephones, has to cope with many different factors such as protocols and technology standards. Also, the limitation of wireless devices, such as network connectivity and network speed, yields important challenges that developers have to deal with. The author argues that requirement engineering is a collaborative task.

Hammond et al. [5] presents interesting scenarios such as air and railway traffic control, which reveal risks in requirement engineering. The authors argue that understanding the operating environment and application domain is important especially in system integration. Although, many of these arguments are valid, the authors only focus on the relationship and traceability among subsystem properties, regardless of the mobility properties of these subsystems. Note that, in mobile environments, mobility properties have a crucial impact on the system development.

Zimmerman [16] suggests the ”MOBILE” framework to determine when mobile computing technology should be used to solve challenges related to mobility. This framework focuses on the current technology and software development trends in mobile computing. Further, common related scenarios are discussed, including news reporting and hotel operations. The framework provides a useful overview of necessary support needed for specific mobile environments. However, the framework does not provide any guidelines for how to develop or design systems for mobile support.

Satyanarayanan [12] identifies four constraints of mobile computing which mainly are concerned with limited resources, physical security (e.g. hazards), and communication and durability issues. Another approach is proposed by Forman and Zahorjan [4] 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.

Rakotonirainy [10] 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 technology. Although characteristics of mobile work can be derived from this approach, he does not provide a comprehensive framework for characterising mobile work environments.

The related work mentioned above mainly focuses on the technical parts of mobility including mobile hardware, limitations, and benefits. Our framework focuses on the mobile work itself and tries to derive the functionality, architecture and hardware required to support specific mobile scenarios. In a real mobile work situation there are more factors than mobile technology to consider. Examples are co-operative work and time and location constraints of the tasks themselves. In addition, our framework can be used to divide mobile scenarios into groups with similar characteristics.

The MOWAHS characterisation framework has not a formal basis to describe mobility. In [14], the notation UNITY is extended to formally describe mobility. The goal of the extension is to establish a formal specification and design technique that can accommodate the concepts of place, time, and action in a manner consistent with the design requirements for mobile systems.

3 THE MOWAHS CHARACTERISATION FRAMEWORK FOR MOBILE WORK

The MOWAHS characterisation framework is aimed at helping to specify requirements for an end-system supporting mobile computing. Such an end-system can be a process support system for mobile work consisting of a server or a set of servers, some mobile clients (laptops, PDAs, mobile-phones etc.), and human resources required to carry out a work process. We believe the characteristics elements of our framework are comprehensive enough to indicate the requirements that are relevant in terms of necessary and/or appropriate architecture (topology, connectivity, network, type of client(s), type of server(s), reliability, security, performance, etc.), services needed, and requirements for hardware to be used.

The task characteristics are divided into four main categories: *general* –install used to describe task structures and attributes that are indirectly related to mobility, *information* – used to specify the information requirements that must be fulfilled to support a task, *location* – used to specify if a task or back-end system support depends on geographical locations, and *time* – used to investigate the temporal properties of a task, and the system support for these properties. Table 1 summarises our characteristics for mobile work. From the framework, we can derive both non-functional and functional requirements for mobile work. We believe that the usage of such a framework enables us to explore typical classes of mobile work with different processes and transactional support.

In [13], we have outlined the steps necessary to effectively apply our characterisation framework. These steps are as follows:

1. Select the mobile scenario
2. Identify the different roles in the scenario
3. For each role; identify tasks
4. For each task:
    a) Write a task description using the task description template
    b) Assign task weight
    c) Characterise the task using the characterisation framework
5. Apply the process characteristics:
    a) Calculate weighted mean values for each characteristic
    b) Calculate an overall scenario complexity indicator
6. Derive system requirements and priorities from the process characteristics

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 a 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.

Table 1. The MOWAHS characteristics for mobile work

To measure the different characteristics in the framework, we use an ordinal scale (1-5). High values indicate more complexity in terms of system requirements, while low values indicate low complexity. Some of the characteristics do not use the full scale, but only the values 1, 3 and 5 to get a uniform representation of extreme values.

The framework should be applied to a mobile scenario focusing on one role and one task at a time. The importance of a specific task will usually vary depending on the user. This means that a task may be important for a specific user, while it is less important for another.

To define this importance, in our framework, each task is assigned a weight on a scale 1-5 (very low, low, medium, high, very high). After analysing several tasks, the mobile process characteristics can be determined based on the weights of each task. The mobile process characteristics are established by calculating the mean value of each characteristic for every task.

4 REQUIREMENT INDICATORS

From the scores of our framework we can compute different indicators that can help us analysing the mobile scenarios and help us to prioritise and extract non-functional and functional requirements. The four first indicators are characteristics computed from the four groups in our framework (see Section 4.1: *General, information, location,* and *time.* The other five indicators are computed by combining characteristics from different groups of the framework (see Section 4.2. These indicators are computed based on some key and additional characteristics. The additional characteristics are not relevant, if the average of the key characteristics have a score more than 3. A summary of all the indicators can be found in Table 2.

To be able to see the complexity of a whole scenario, we can compute the *Overall scenario complexity indicator*. This indicator is the average of all characteristics of our framework.

4.1 DIRECTLY COMPUTED INDICATORS

The following are indicators that are computed from how the mobile work characteristics are grouped in the MOWAHS framework (see Table 1):

- **General Task Indicator (GTI)** is an average of all general characteristics (G1-G5), constituting its key characteristics. A GTI score indicates the complexity of the analysed task. A high score may, for instance, mean that an underlying process and transaction system must be able to support complex – i.e., flexible and advanced – tasks. Further, this indicator specifies the basic functionality required to handle the process (tasks) and data exchange. It also determines correspondingly the complexity of the required tools (general support) – e.g., workflow tools.
- **Information Complexity Indicator (ICI)** is an average of all information characteristics (I1-I5), which constitute its key characteristics. A high ICI score indicates that the end-system must cope with complex information presentation, management, and transmission. It thus determines the complexity of information management and information exchange mechanisms, including the *quality of service*. The ICI can also be used to select the suitable mobile host device (software and hardware) and the appropriate server (software and hardware).
- **Location Complexity Indicator (LCI)** is an average of all location characteristics (L1-L5), constituting its key characteristics. A high LCI score indicates that the end-system should possibly include GIS-support and should put high requirements on the type of equipment that can be used as mobile clients. In addition, it tells how the location affects the actual task performance. It also determines the required portability of the relevant mobile devices – i.e., the weight, the size, etc. A high score also indicates that some GIS-support should be supported to define the location of the task and the need for representing location in the process modelling language. Finally, it can determine the need for the system to be aware of location. This means that some data may be relevant at a specific location, other data may not, etc.
- **Time Complexity Indicator (TCI)** is an average of all time characteristics (T1-T5), which constitute its key characteristics. A high TCI score means that management of tasks must be timed, and that it may require more advanced transactional support. It also determines how time affects the actual task performance. Further, the TCI score indicates the (non-functional) system performance and system availability. Moreover, it specifies the required (functional) task scheduling mechanism and task synchronisation or coordination.

4.2 DERIVED INDICATORS

We have identified some indicators that we find useful in order to specify the requirements (both functional and non-functional) for mobile work support. The following are indicators that are derived by combining some specific characteristics from different groups. They are useful in identifying the degree of complexity of both the mobile clients and the service providers, including hardware and software issues.

- **Network Connectivity Indicator (NCI)** indicates the need for being online. The key characteristics for this indicator are G3, G4, G5, L4 and T1. In addition, T4 may be used to derive NCI. A high NCI score means that the mobile client must be online most of the time. It thus determines the required networking capabilities for the actual mobile host. Also it specifies non-functional requirements for the system such as reliability and latency.
- **Network Speed Indicator (NSI)** specifies the needed transmission speed between a mobile client and the end system. In other words, a high NSI score means that the transmission speed must be high, while a low score tells that even low transmission speed may work. The

Indicator	**Key Characteristics**	**Additional Characteristics**
General Task Indicator (GTI)	G1 – G5	NONE
Information Complexity Indicator (ICI)	I1 – I5	NONE
Location Complexity Indicator (LCI)	L1 – L5	NONE
Time Complexity Indicator (TCI)	T1 – T5	NONE
Network Connectivity Indicator (NCI)	G3 – G5, L4, T1	T4
Network Speed Indicator (NSI)	I3 – I5	G5, I2
Energy Consumption Indicator (ECI)	L1 – L3, T5	G5, I1, I2, L4
Transactional Support Indicator (TSI)	G3 – G5, T1, T4, T5	NONE
Mobility Indicator (MI)	L4, L5	L1, L3, T3

Table 2. Summary of the indicators

key characteristics for this indicator is I3, I4 and I5, while G5 and I2 may be needed as supplements. From above, we may conclude that the NSI would be useful to set some non-functional requirements that are relevant, such as performance, quality of service and latency. It also indicates the kind of network technology that must be available for use.

- **Energy Consumption Indicator (ECI)** describes the expected required battery lifetime on mobile device. Its key characteristics are L1, L2, L3 and T5. In addition, G5, I1, I2 and L4 can be used to derive this indicator.

  A high ECI score means that the mobile device requires supplies that are able to deliver high energy, while a low score signifies low energy consumption. Based on this, the ECI indicates the energy technology needed that should be provided or selected, considering portability versus energy cost. In other words, this indicator helps designating appropriate energy supplies in accordance with the actual energy consumption. It also indicates the required and suitable mobile device hardware. In addition, this indicator implicitly shows how long the task should be executed – i.e., task lifetime, which often corresponds to the actual energy consumption.

  High-speed networks and high performance multimedia applications or hardware often has high energy consumption. Therefore, tasks requiring such types of equipment may have to consider the necessity to provide long-lasting energy supplies. In our framework, this can be specified in I1, I2 and I5. Moreover, the available energy supplies may affect task executions. Sometimes, it may be necessary to reschedule a task if the power supplied is lower than that required. For example, delegation or postponing of tasks may be useful to deal with this.

- **Transactional Support Indicator (TSI)** describes the need for flexible/advanced transactional support, and is primarily an average of the following characteristics: G3, G4, G5, T1, T4 and T5. A high TSI score indicates that the transactional support must be flexible and advanced; while a low score may indicate that traditional transactional support (ACID transactions) is sufficient.

  Hence, the TSI is useful in determining the suitable transactional support to be provided. It mainly indicates how advanced the transactional support should be in performing tasks. In effect, the TSI will help us identify the suitable transaction models or transactional frameworks, including transactional tools and applications that have to be provided. High TSI scores require advanced transactional models [3, 9, 6] or even customisable transaction models [11]. This means that we may have to relax the ACID properties to allow the execution of a specific task or preserve these properties to ensure correct executions and consistent results. Note that in some cases we may be forced to allow temporal inconsistencies to be able to execute a task. Thus, we have to allow the transactional support to be non-ACID. How this can be achieved, has been shown in [11]. Further, as cooperation between tasks may be necessary, transactions must be able to cooperate, and the necessary flexibility may require customisable transaction models.

- **Mobility Indicator (MI)** describes how much mobility is involved in task execution. Its key characteristics are L4 and L5, but in addition, it may be derived from L1, L3 and T3. A mobility indicator can thus be found by computing the average of the characteristics L1, L3, L4, L5 and T3. Here, L1, L3, L4 and L5 are directly related to location and mobility of a task, while T3 implicitly affects the mobility of tasks. For example, if the value of T3 is 3 or 5 – i.e., tasks have temporally to be coordinated; and this may require changing the involved task locations. This means that a task may have to be executed at a specific location in order for it to be coordinated with another task.

  A high average score of the above characteristics indicates that a task is highly mobile, while a low score tells us that a task may not be mobile or has low mobility. It is important to note that the value of this indicator will perhaps be affected by the value of other characteristics. More specifically, location dependency (L1) may be triggered by the available transmission speed (I5). Further, information produced at a specific location (L3) may relate to the information required related to time (I3) or the information produced related to time (I4). In this respect, MI is indirectly affected by I3, I4, and I5.

  Intuitively, this indicator is useful in determining the complexity of a task based. It thus affects the type of equipments (devices) and tools that are necessary to accomplish a task. From this perspective, the MI may indicate that some specific mobile devices are necessary to perform a task. In addition it may specify appropriate tools (e.g., WAP, Lightweight Java, etc.). In this view, it may also be used to determine relevant technology that is applicable, such as network, mobile device and mobile computer technology.

5 INDICATORS APPLIED TO VARIOUS SCENARIOS

In this section we will show the results from analysing various mobile scenarios using the indicators described in section 4.

Here is a short description of the four scenarios we have analysed:

1. **IT support scenario:** IT support people handles inquiries from users about hardware and software, and install new of software and hardware. To perform their work they have to move around to specific locations (like offices and computer labs) where the computers and equipment are located. The tasks for the IT support people that we have considered in this scenario, are: *Setting up new computers*, *computer upgrades*, and *assist employees to deal with software/hardware problems*. A more detailed description of this scenario can be found in [15] based on interviews with the IT-support department at our university.

2. **Mobile researcher scenario:** This scenario describes a researcher travelling to a conference and working abroad. The tasks that we have considered for the travelling researcher, are: *Prepare presentation*, *perform presentation*, *write on scientific papers*, and *miscellaneous administrative tasks* (email, coordination with colleagues, etc.). This scenario is further described in [13] and is based on our own experiences from travelling and working.

3. **News production scenario:** This scenario describes the work in a newspaper with the purpose of reporting the latest events happening in a city, involving the roles reporter and photographer. The reporter will *collect data from the site of event* (like interviews and notes), *write a report*, and then *review the report*. The photographer will *take pictures from the site of event*, and *edit pictures*. This scenario was produced based on an interview with a journalist in the biggest newspaper in Norway.

4. **Mobile learning scenario:** Mobile learning (mLearning) is an extension of eLearning making learning available anywhere at anytime. In our scenario we describe typical tasks for students and teachers performed in a mobile setting in a biology course at a university. Students perform the following tasks (using mobile devices): *Watch video lessons*, *read course material*, *do exercises*, *work with a group on a project*, and *check course status*. Teachers are involved in the following mobile tasks: *interactive field excursion reporting* (using audio/video recording), and *record live video/audio lessons*. This scenario describes typical tasks for a course at our university put in a mobile setting.

After analysing the four scenarios, we found that the news production and the mobile learning scenarios had the highest *Overall scenario complexity indicator*, with the scores 3.35 and 3.32 respectively. The IT support scenario scored 2.31, and the mobile researcher scenario scored only 1.90 on the same indicator. This result confirms our suspicion that a system to support a mobile researcher would be the least complex system, and the mobile learning and news production scenarios involving real-time video transmission would be the most complex ones. Also note that the mobile researcher scenario represents nomadic work, which normally can be supported by a standard distributed system. We can analyse these results further by looking at the more specific indicators as shown in Figure 1. Looking at the *General Task Indicators*, we can see that scenarios 3 and 4 demand a more advanced (flexible) task support/infrastructure (e.g. workflow system). Advanced task support often means that you need advanced transactional support in order to allow such a flexibility. If we look at the *Transactional Support Indicator*, we can see that the scenarios 3 and 4 also score high here.

The *Information Complexity Indicator* (ICI) is used to indicate how the end-system must cope with presentation, management, and transmission of information. Also here, we can see that scenarios 3 and 4 have high scores for the ICI. This is probably because both scenario 3 and scenario 4 involve real-time video, while the scenarios 1 and 2 involve text or possibly graphics that are not transmitted frequently.

Looking at the *Location Complexity Indicator* (LCI), we can see that here scenario 1 is the most complex in this respect. This is because the IT-support scenario involves several tasks that are locked to specific locations and it is important to be able to locate the IT-support people.

The *Time Complexity Indicator* (TCI) indicates how complex the time management of tasks need to be in the end-system. Here, the news production scenario is the most complex, due to strict time constraints and coordination of tasks. The next two indicators are the *Network Connectivity Indicator* (NCI) and the *Network Speed Indicator* (NSI). Also here scenarios 3 and 4 have the highest scores due to a need to transfer big amounts of data real-time. The IT-support scenario also has a high NCI score because the need to the report location of IT-support people.

The scores of the *Energy Consumption Indicator* (ECI) shows that mobile clients required in scenario 1 consume most energy, followed by scenario 4. The IT-support scenario has a high ECI because the IT-support people needs to go to specific locations over a long period of time using a mobile client. In the mobile learning scenario, a lot of energy is consumed when watching a video on a mobile device or making video recording at a field excursion over several minutes or possibly hours. The last indicator is the

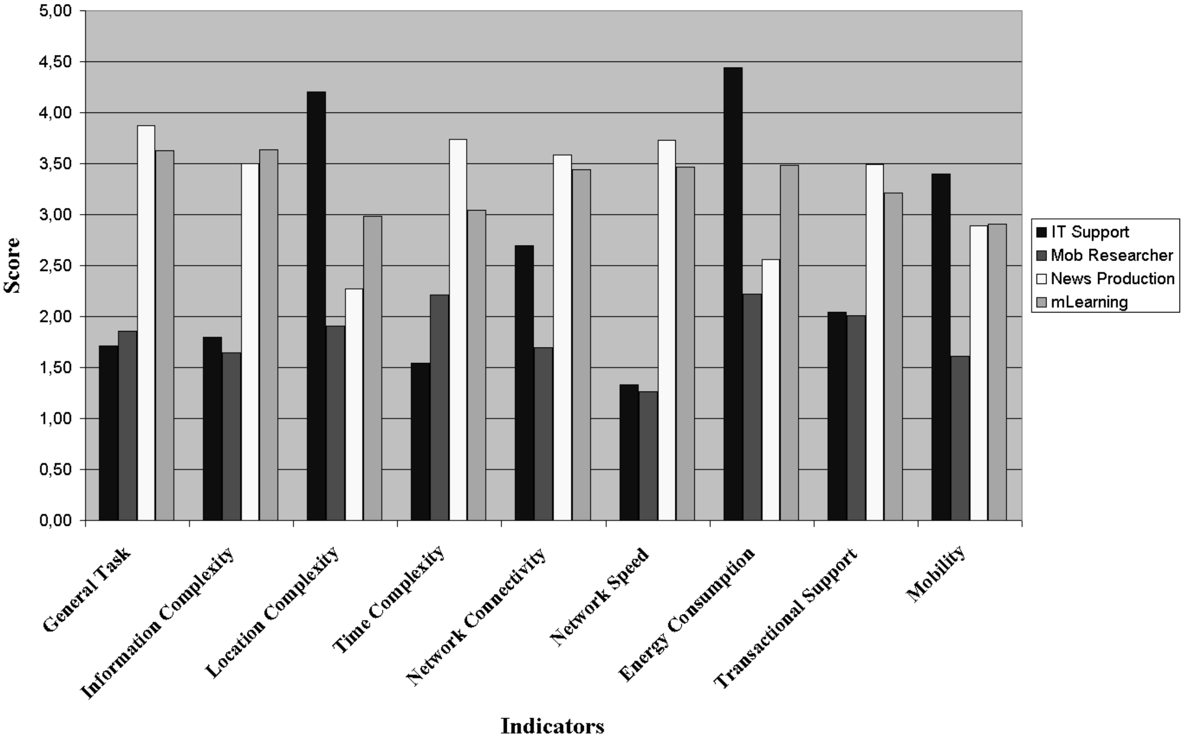

Figure 1. The results of applying the indicators to the scenarios

Mobility Indicator, which describes how much mobility is involved in the execution of tasks. Also here the IT-support scenario has the highest score followed by news production and mobile learning. The mobility indicator does not indicate the mobility measured in distance, but rather how much of the work involves moving around or to specific locations. The tasks of the IT-support scenario must be performed at a specific location to be completed, thus a high mobility indicator. The mobile researcher is not so much bound to specific locations, giving a low mobility indicator score.

From these examples, we can show that our indicators are valid for several different situations. This also means that for future development of mobile systems, we can use them to set up relevant priorities and requirements.

6 DISCUSSION AND CONCLUSION

The indicators presented in this paper are oriented towards different implementation aspects of a system. It is therefore interesting to discuss what the indicators mean in terms of the development of a system. We have found that the indicators represent tools to find which parts of a system are the most challenging (and uncertain) in terms of complexity, and therefore also solvability.

In agile methodologies, like eXtreme Programming (XP) [1], architectural prototypes are used to delimit the uncertainties of the estimates to implement certain requirements (user story). To illustrate, a high ICI indicates that the system should be able to handle rather complex information. To make it certain that the forthcoming system is able to provide the necessary capability to handle the data, a prototype can be built to ensure that the different parts of the system is able to handle the required data, e.g., in a timely manner, on a specific equipment or a wireless network. In some cases, the hardware or network has limitations that must be taken into account. Then it might be of interest to build a prototype system that adapts to the capabilities of different devices. Here, it may be argued that the limitations are known. However, due to the constant evolution of all components of the mobile infrastructure, software solutions not only have to adapt to the current (state-of-the-art) technology, but also have to deal with future developments. An example is the availability of components – e.g., component of the shelf (COTS) – to integrate into the system to solve particular issues such as an uncertainty with respect to applicability, functionality etc.

Both the characteristics and the indicators can be used as inputs to harvest components for the system architecture. However, it may be hard to have a precise opinion of what is really needed, even if all the major requirements are captured. The indicators can be used as a means to find out whether components could be developed in-house (in the case of low indicators), or should be bought and integrated into the system (in the case of high indicators), and even investigate if similar systems are already available.

In many cases, however, it is the use of the framework on several scenarios that will be of the highest value. Certain characteristics of systems are often recurrent, and it may exist design patterns to formulate and solve similar problems. In the case of, e.g., the information complexity, it is possible to find certain kind of architectures that can provide the required level of quality of service, such as in database technology and server proxies.

This paper has presented eight complexity indicators that we believe are useful to set up and prioritise requirements for a mobile work environment. We have also shown that these indicators can be applied to real-world scenarios. This has provided us the possibility to assess the complexity of such mobile scenarios. Our indicators can also be used to group scenarios with common characteristics. This means, in turn, that scenarios of the same group may reuse similar technology and architecture.

Currently, we are using our framework and indicators to develop a mobile IT-support system. This will help us to evaluate the strengths and weaknesses of our approach in a practical setting. For example, this may reveal what indicators are important and what are less important.

ACKNOWLEDGEMENT

This paper is part of the MObile Work Across Heterogeneous System (MOWAHS) project. The MOWAHS project is sponsored by the Norwegian Research Council's IKT 2010 programme.

REFERENCES

[1] Kent Beck. *eXtreme Programming Explained : Embrace Change*. The XP series. Addison-Wesley, Reading, Mass., 1999.

[2] Julio Cesar Sampaio do Prado Leite. Is there a Gap between RE Research and RE Practice? In *Proc. of Fourth International Conference on Requirement Engineering (ICRE 2000)*, pages 73–74. IEEE Computer Society, 2000.

[3] Ahmed K. Elmagarmid, Yungho Leu, James G. Mullen, and Omran Bukhres. Introduction to advenced transaction models. In Ahmed K. Elmagarmid, editor, *Database Transacation Models for Advanced Appliactions*, pages 35–52. Morgan Kaufmann, 1992.

[4] George. H. Forman and John Zahorjan. The Challenges of Mobile Computing. *IEEE Computer*, 27(4):38–47, April 1994.

[5] Jonathan Hammond, Rosamund Rawlings, and Anthony Hall. Will it work? In *Proc. of Fifth IEEE International Symposium on Requirements Engineering (RE'01)*, page 102, 2001.

[6] Sushil Jajodia and Larry Kerschberg. *Advanced Transaction Models and Architectures*. Kluwer Academic Publisher, 1997.

[7] Alessandro Maccari. The Challenges of Requirements Engineering in Mobile Telephones Industry. In *Proc. of 10th International Conference and Workshop on Database and Expert Systems Applications (DEXA '99)*, Lecture Notes in Computer Science 1677. Springer, 1999.

[8] Nancy R. Mead. Why is it so difficult to introduce requirements engineering research results into mainstream requirements engineering practice? In *Proc. of Fourth International Conference on Requirement Engineering (ICRE 2000)*, pages 75–76. IEEE Computer Society, 2000.

[9] C. Mohan. Tutorial: Advanced Transaction Models - Survey and Critique. In *Proc. of the ACM International Conference on Management of Data (SIGMOD'94)*, page 521, May 1994.

[10] Andry Rakotonirainy. Trends and Future of Mobile Computing. In *10th International Workshop on Database and Expert Systems Applications*, 1–3 September 1999.

[11] Heri Ramampiaro and Mads Nygård. CAGISTrans: Providing Adaptable Transactional Support for Cooperative Work – An Extended Treatment. *Accepted for publication in the Special Issue of the International Journal of Information Technology and Management*, 2002.

[12] M. Satyanarayanan. Fundamental Challenges in Mobile Computing. In *Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing*, pages 1–7. ACM Press, 1996.

[13] Carl-Fredrik Sørensen, Alf Inge Wang, Hien Nam Le, Heri Ramampiaro, Mads Nygård, and Reidar Conradi. The MOWAHS characterisation framework for mobile work. In *Proceedings of IASTED Internatiol Conferenve in Applied Informatics 2002 (AI'2002)*, February 18-22 2002.

[14] C. Donald Wilcox and Gruia-Catalin Roman. Reasoning About Places, Times, and Actions in the Presence of Mobility. *IEEE Transactions on Software Engineering*, 22(4):225–247, April 1996.

[15] Øystein Hoftun. MObile Nagging Geek Organizer. Technical report, Dep. of Computer and Information Science, NTNU, Norway, June 2002. Diploma thesis (311 p.).

[16] James Bryan Zimmerman. Mobile Computing: Characteristics, Business Benefits, and the Mobile Framework. Technical Report INSS 690 CC, University of Maryland European Division - Bowie State, April 2 1999.