AN EVALUATION OF A MOBILE TASK REPORTING SYSTEM FOR DIFFERENT MOBILE DEVICES

Carl-Fredrik Sørensen* carlfrs@idi.ntnu.no
Alf Inge Wang alfw@idi.ntnu.no
Øystein Hoftun oysthof@idi.ntnu.no
Kenneth Aron Bjørkhaugen kenneb@idi.ntnu.no

ABSTRACT

This paper describes an evaluation of using three different types of mobile devices as clients in a mobile task reporting system. The heterogeneity of hardware and software introduced by various mobile devices has made it harder for software developers to make systems that can support this broad spectrum of devices. We have made a prototype of a system to handle management of tasks for mobile workers. Mobile workers are offered to bring parts of the system on mobile devices, making it possible to update task status and write small reports on the spot when finishing a task. We have in this paper evaluated how well the system has worked for the different devices based on several evaluation criteria like screen size, input device, physical device size, battery life, connectivity etc. In this evaluation we have used a portable PC, a PDA, and a mobile phone. We found that the system should provide different support dependent on the device. The mobile phone was found to be useful for checking task states and for limited information browsing. The PDA was very useful for writing end-reports when finishing tasks (on the actual location), and the portable PC was best for managing tasks (getting overview of the different tasks and states), and for task delegating tasks. We also discuss the problem of keeping data consistent when several off-line mobile devices can update the same data, and we indicate some solutions to this problem.

Keywords: *Usability, mobile work, mobile devices, data consistency problems, mobile software systems*

1 INTRODUCTION

Many people use mobile electronic devices both professionally and for pleasure. These mobile devices span from small mobile phones to powerful portable PCs with different screen sizes, input devices, CPUs, memory sizes, network connectivity, and operating systems. The diversity of device types and capabilities are substantial challenges for software providers that want to make systems to be used on different devices. *First*, different user-interfaces that fit the different screen sizes have to be provided. This is not only a question about how the graphics and text is formatted, but also how much information and functionality should be available for the different devices [10]. The functionality provided by the system is also dependent of the supported input capabilities of the device, such as a limited keyboard (mobile phones), pen-based input, voice recognition, or a full-fledged keyboard. The format of the graphics and text is not only dependent of screen size, but also dependent of the screen's ability to display colours (might only support grey-scale or black and white). *Second*, the software providers must also consider the connectivity of the devices. Some mobile devices have a network connection only through other devices such as mobile phones (via infra-red) or PCs. To support such devices, a software system must store the necessary information on the device to enable off-line operation. Other devices such as GPRS mobile phones have a fixed network connection and do not need to buffer the information. *Third*, the software providers should be aware of other limitations of the devices such as battery lifetime and network connectivity only within limited areas (e.g. Wireless LAN). As can be observed, software development for mobile devices must allow heterogeneous software and hardware to live side by side, and make the necessary bridges and adaptations to facilitate the view of one system.

In this paper, we describe our experiences from implementing a prototype of a mobile task reporting system (MTRS) for the IT-support department at the Norwegian University of Science and Technology (NTNU). The daily routines of the IT-support staff involve mobile work where they have to go to specific locations (offices, computer labs etc.) to repair or change hardware, or to fix or install software. Today, the IT-support staff is supposed to use a system called RUST [8] for handling inquiries (tickets) from users about e.g. faulty hardware and software. RUST is used to allocate personnel to handle the different inquires. The personnel should write a short report on experiences and change the state of the inquiry after completion of the task. RUST is web-based and the user must login to use the system. Today, few tasks are reported as completed in RUST. Often the tasks are forgotten or another task is started before the staff returns to their offices. To attack this problem, we proposed to make a mobile task reporting system where the IT-support staff can carry the system with them on mobile devices, and use the mobile devices to enter the task completion reports. This paper describes the mobile task reporting system and evaluates the usability of this system on three different devices: A WAP mobile phone, a PDA, and a laptop PC. In this evaluation we wanted to investigate the main differences between the devices and how well the mobile task reporting system worked on the different de-

*Dept. of Computer and Information Science, Norwegian University of Science and Technology (NTNU), N-7035 Trondheim, Norway. Phone: +47 73594485, Fax: + 47 73594466

vices.

The rest of the paper is organised as follows: Section 2 describes the Mobile Task Reporting System, and Section 3 describes the criteria we used to evaluate the devices used on the system and the result from this evaluation. Further, in Section 4 we relate our work to similar research, and finally, Section 5 concludes the paper.

2 THE MOBILE TASK REPORTING SYSTEM (MTRS)

The Mobile Task Reporting System (MTRS) [15] does not provide all functionality needed to be a deployable system for the support staff, but it is capable of adding and removing tasks, assign tasks to personnel, and entering end reports after task completions. To make a system that was capable of interacting with various devices, we chose to implement a three-tier architecture as shown in figure 1. The client tier consists of a WAP [12] or a Web browser dependent on what device is used. For mobile phones, a WAP browser provides the user interface. For PDAs, we used the Web browser AvantGo [1] that runs both on PalmOS and PocketPC. For portable PCs, Netscape or Internet Explorer was used as the client software. The Web tier detects the client types interacting with the MTRS and provides the correct client data dependent on the type of device. WML is used for WAP mobile phones [12], HTML [14] adjusted to small screens is used for PDAs, and normal HTML is used for portable PCs. Task information is stored as XML-files [13] in the data tier independent of client type. XML was chosen for storage of data because it is easy to transform XML to WML or to HTML, and to change the definition of the task information.

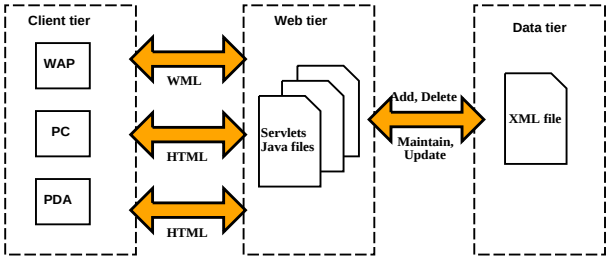

Figure 1. The Mobile Task Reporting System Architecture

3 THE EVALUATION

This section describes how we evaluated the MTRS and the results from this evaluation. First, we will present the criteria we used to evaluate the usage of MTRS on mobile devices. Next, we will give a description of the mobile devices used in the evaluation, and we will present the results of the evaluation.

3.1 THE EVALUATION CRITERIA

The criteria we used to evaluate the mobile task reporting system on the different devices are as follows:

E1 **Connectivity:** This criterion describes how well the MTRS worked related to the network connectivity of the device.

E2 **Display:** This criterion describes how well the information of the MTRS could be displayed to the user of the devices.

E3 **Input:** This criterion describes how easy it was to enter information into the MTRS related to the available input capabilities.

E4 **Portability:** This criterion describes how easy it was to carry the device when mobile.

E5 **Cost of usage:** This criterion describes the cost of using the device (depends on the network connection) as a client of MTRS.

E6 **Battery lifetime:** This criterion describes how the battery lifetime of the device affects the usage of the MTRS.

E7 **Price:** This criterion describes how much the device costs.

E8 **Device capabilities:** This criterion describes the devices capabilities in terms of CPU, memory, and secondary storage.

E7 and E8 are not directly related to the actual usage of the MTRS.

3.2 THE MOBILE DEVICES USED IN THE EVALUATION

When we selected devices for our evaluation, we wanted to have three devices with different sizes and capabilities. Thus, a portable PC, a PDA, and a WAP mobile phone were selected:

- **Dell Inspiron 8000:** This is a rather powerful portable PC with the dimensions 326 x 276 x 44 mm and its weight is 3.26 Kilograms. Battery operation time is about two hours. Dell Inspiron has a large 15” screen that can display up to 1400 x 1050 pixels in 32-bit colours. It comes with a 1 GHz Intel Pentium III processor and has between 128–512 MB of memory. Secondary storage can be 10–60 GB hard drive. DVD ROM or a combined DVD CD-RW is provided. The input devices are an 88 key keyboard and a touch pad pointer. There are a number of different possibilities to connect this device to a network. The network capabilities of the tested Dell were: 56 KBit/s analogue modem, 100 MBit/s LAN, 11 MBit/s WLAN, and GSM via infrared (mobile-phone). In addition, there are a wide range of possibilities for other kind of network connections such as BlueTooth, GPRS, GSM-card, etc. The device can also be extended with a wide range of input devices such as voice control, Virtual Reality gloves, etc.
- **Palm m505:** This is a small PDA with the dimensions 114 x 78 x 13 mm, and its weight is 140 gram. It has a 16-bit colour (65,000) screen of 160 x 160 pixels. The

CPU of Palm m505 is a Dragonball VZ processor from Motorola running at 33 MHz, and it comes with 8 MB of memory and storage. In addition small memory expansions cards up to 128 MB can be used as secondary storage. Text can be entered using the provided stylus or an additional portable keyboard. The Palm m505 primarily connects to the network through a cradle connected to a PC. In addition, it is possible to connect to the network through mobile phones (via infrared or BlueTooth). GPS and GPRS expansion cards are also expected in near future. The battery operation time is about 4-6 hours dependent on which kind of applications that are running.

- **Nokia 7110 WAP mobile phone:** This phone is an average sized phone with the dimensions 125 x 53 x 24mm and its weight is 141 gram. Nokia 7110 has a rather large black and white display of 96 x 65 pixels and up to 6 lines of text. The device uses GSM for communication, and provides data transmissions up to 14,400 bps. It has an infrared port for communication with other devices (like PDAs and PCs). For browsing the Internet, Nokia 7110 uses an Internet WAP micro-browser. Battery operation time for continuous WAP-usage is about 1 hour.

3.3 THE RESULT

This section describes the results from the evaluation of using the MTRS on the three different devices.

3.3.1 Evaluation of MTRS on a portable PC

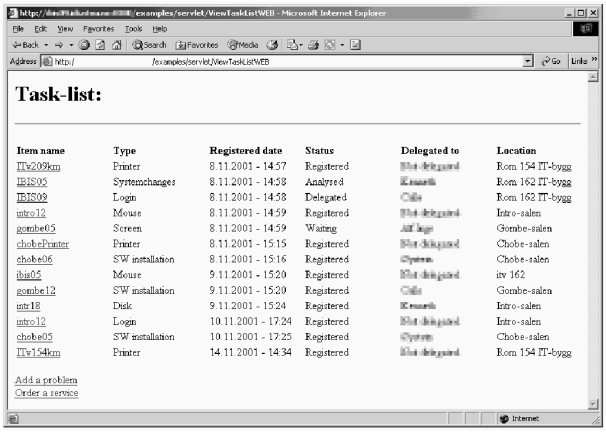

Figure 2. Screenshot from MTRS on Dell Inspiron 8000 portable PC

Figure 2 shows a screenshot when running MTRS on a portable PC. When using a portable PC as a client for MTRS, there are many options of network connectivity (E1). One option for the IT-support staff is to connect the portable PC to a network plug in the office or computer-lab when doing the work. This could be a problem because some offices or labs may not have available network plugs. A more convenient connection type is to use Wireless LAN (radio network). Presently, parts of the campus do not have Wireless LAN network coverage making this option uncertain. However, combinations of the different network connections made the PC on-line most of the time.

As shown in figure 2, the display (E2) of the portable PC is big enough to get a good overview of several tasks on the screen. The full-sized keyboard on the portable PC makes it very easy to enter input (E3). The main problem using the portable PC is portability (E4). The Dell Inspiron 8000 we used is rather big and heavy, and is not a device you would carry with you all the time. To solve this problem we could use a mini-PC that is a lot smaller and lighter, but it would still not be small enough to be put into a pocket. There were not any extra costs (E5) when running MTRS on a portable PC because only the campus network was used.

3.3.2 Evaluation of MTRS on a PDA

Since the Palm m505 does not come with any network connections (E1), the PDA was used as an off-line client of MTRS. This meant that all necessary information and functionality were downloaded to the device before the user could start to use the MTRS. When finishing a task, the task report from the IT-support staff was buffered on the device after entry, and this information was synchronised with the rest of the system the next time the PDA was connected to a PC (via a cradle). There are two main problems by using MTRS off-line:

1. The IT-support staff will not be notified about new tasks when working away from the office. This means that if a person is working in a PC-lab, that person will not be notified about any new tasks in the same lab before returning to his office.
2. By using off-line clients, several persons may update the same data. This might cause two persons to incorrectly work on the same task, and that task status and information are incorrectly overwritten when synchronising the PDA.

The only solution to the first problem is to make the PDA online, e.g. by adding a Wireless LAN card or similar. To the second problem there are several possible solutions. The simple solution is to use *exclusive locks*. This means that only one person can access the data (for read and write) of a task simultaneously. However, this solution has several disadvantages. First, if an IT-support person using a PDA exclusively locks access to the tasks he is responsible for, the manager cannot check the status of these tasks before the locks are released. Also the tasks can then only be accessed and changed from the mobile device, even if the same person wants to make some changes on his office PC. The advantage with this approach is that the task information will always be correct (both for read and write). Another solution is to use *write locks*. This means that only

one person can update a task, while others can have read access to the task information. A problem with this approach is to decide when to set the write lock. Since the update of task information is not performed before the PDA is synchronised with the PC, the write locks must be set when the PDA synchronise with the MTRS system before starting to work on a task. Although this approach solves the problem of letting others view task information, it also has some negative consequences. The main problem is that all the tasks that are transferred to the PDA have to have write locks to ensure correct updates, even if only a few of them will actually be updated. This means that a lot of tasks will have write locks for long periods of time making the MTRS rather inflexible. Another solution is to avoid locks by letting the last person updating the task decide what should be the correct task information and status. Here, it is also possible to provide a merge function that merges the task log. However, if there are two different status of a task, the last person updating must decide what the correct status is. It is also a middle course, by using *soft locks*. This means that the user indicates what tasks he is likely to update. If another person indicates that he wants to update the same task, a communication channel (email, SMS, phone, IRC, instant messenger etc.) between these two persons will be established to negotiate about how to proceed. A detailed description and discussions about solutions to this problem are described in [11].

Figure 3. Screenshots from MTRS on Palm m505 PDA

Figure 3 shows screenshots from running MTRS on the Palm device. As the screenshots show, it is not possible to get the same perspicuous overview as we did on the portable PC. However, the display (E2) was big enough to get a reasonable overview of the tasks. The pen with the touch screen, and letter recognition were used to get input (E3) from the user. The speed of entering text using a pen on a PDA is highly depended on the skill of the user. A skilled PDA user can write about 20 words per minute, while 60 words a minute is normal when using a full size keyboard. However, if the IT-staff have to think while writing the end task report, the speed of entering the words does not count that much.

The Palm m505 is very light and therefore an easily portable (E4) device. It is no problem to put the device in a pocket and bring it with you wherever you go. Since the PDA was used offline there were not additional costs (E5) related to usage.

3.3.3 Evaluation of MTRS on a mobile phone

The Nokia 7110 is the smallest device of the three (though not in thickness), but it has the same weight as the PDA we used. This means that the mobile phone is highly portable (E4), and easily can be put into a pocket. Because of the size, the display (E2) and the keyboard (E3) of the phone are small. In figure 4 we can see screenshots from the MTRS using the WAP browser on the mobile phone. The screen of the phone is very limited, and it is almost impossible to get an overview of the tasks using this small display. Another problem is the provided user interaction when using the WAP browser. To change an item in a menu, the user has to go though several steps. First the user must select the item he wants to change. Then from the phone's menu he must choose to edit the item, before he can choose between the different options available. The user must also go through these steps for entering text. The extra steps that are required through the WAP interface, makes it very time-consuming and tedious to do even simple tasks.

Figure 4. Screenshots from MTRS on Nokia 7110 WAP mobile phone

The network connection (E1) of the mobile phone is excellent. The user can be connected to the network everywhere and at any time. However the network connection is rather expensive (about $1/8minutes or more in Norway). This means that it will be expensive to have users of the system connected at all time.

3.3.4 The Comparison of the Three Devices

In table 1, we can see a summary of the results from the evaluation of using MTRS on the portable PC, the PDA and the mobile phone.

From the table we notice that the price (E7) of the different devices is very high. The portable PC is 10 times as expensive as the PDA. The PDA is almost 3 times as expensive as the mobile phone. However, the mobile phone adds additional costs in usage that must be taken into account. We can also from the table see that the battery lifetime (E6) of the three devices varies. The PDA can be used up to

Evaluation criteria	Dell Inspiron	Palm m505	Nokia 7110
E1. Connectivity	LAN/WLAN/Modem	using PC/Mob Phone	GSM
E2. Display	colour 1400 x 1050	colour 160 x 160	b/w 96 x 65
E3. Input	keyboard + touch pad	pen-based	num keyboard
E4. Portability	32x27x4cm, 3300g	11x8x1cm, 140g	12x5x2, 141g
E5. Cost of usage	None	None	$1/8min
E6. Battery lifetime	2 hours	4-6 hours	1 hour
E7. Price	$3500	$350	$125
E8. Device capabilities	Very high	Medium	Very low

Table 1. Evaluation Summary

4-6 hours continuously, the portable PC up to two hours, and the mobile phone up to one. If an IT-support person spend most of the day at different locations, all the devices should last one working day before recharging. The mobile phone's battery lifetime will be too short if the user stays on-line most of the day. When using the portable PC, it is important the battery saving operations is activated while not using the PC (as putting the PC in sleep-mode etc). It is also possible to add an extra battery to the portable PC that can extend the operation time up to four hours. However, an extra battery will add at least 1 Kilogram of weight to the PC.

Because of the limited input and display of the mobile phone, the MTRS should only provide services for checking status of tasks and to get online information for mobile devices. The PDA is ideal as a mobile tool for entering end-task reports, since it is small and the pen-based input is sufficient for entering short texts. Also the price of the PDA makes it a price-efficient alternative as a mobile toolbox. However, the PDA solution requires a PC to update the changes of the MTRS, which demands an advanced configuration management system to manage control of update of task information. Since all IT-staff have their own desktop PCs today, synchronisation with a PDA does not add any additional costs. The portable PC solution is very good for getting an overview of the system and for delegating tasks.

4 RELATED WORK

In [3], Johnson reflects on the usability when interacting with mobile devices. Johnson has identified four problems to be faced when addressing usability in mobile systems: 1) The demands of designing for mobile users, their tasks and contexts, 2) Accommodating the diversity and integration of devices, network services and applications, 3) The current inadequacy of human computer interaction models to address the varied demands of mobile systems, and 4) The demands of evaluating mobile systems. Our paper explores the problems 1, 2 and 4. When we designed the MTRS, we did not put too much effort in considering design issues for the various mobile devices. However, some preliminary human interaction analysis of the usage of the system shows that the design for each device should be more specialised. The most usual method of solving problem 2 is to use the Web as an infrastructure for a system. We chose to use XML as the data representation to make it easy to process different user interfaces (e.g., WML and HTML) dependent on the device. The disadvantage by using this approach is that WEB interfaces offer rather limited set of graphical user interface components. The fourth problem Johnson identified was the lack of evaluation of mobile systems. Our paper describes a simple evaluation of a mobile system.

In [4], Marcus introduces the term *babyface design* to describe the user interface design for small displays on mobile devices. Challenges in babyface design include limited spatial and colour resolution, limited number of fonts, limited visualisation space for charts, maps and diagrams, and table/list navigation. These are all challenges that we have addressed in our paper. Further Marcus points out that because the diversity of user communities and usage contexts of mobile devices, software design for such devices must also consider cultural context and support for multiple languages. Cultural differences may imply that the mobile client software should use icons bound to a culture as well as using colours and formatting that are understood by the culture. When designing a mobile software system these user interface components must be considered: Metaphors, mental models, navigation, interaction, and appearance. Babyface design introduces some challenges to these user interface components. Marcus argues that the traditional desktop metaphors used for PCs do not necessarily fit for mobile devices. Candidates might be maps, dashboards, etc. For mental models and navigation, it is important that the user interfaces for mobile devices only presents the most important information at a given context. For interaction it is important that the user interfaces are adapted to the input device available, and not bound to traditional entry based on full size keyboards. For the last component, appearance, it is important that the appearance of the mobile system is adjusted to the device in terms of screen size, fonts and colours. Our paper describes experiences from using mobile devices as a web-client. As Marcus pointed out, we found e.g. that using the traditional web interface for selecting an item is not easy to do on a mobile phone. We also experienced that the information provided to the user should be dependent on the device and context of the user.

Buchanan et al. [2] discusses the usability aspects of the mobile Internet, and suggests that a reason for the failure of WAP, as well as potential failures of other approaches, is that not enough time is spent to think about the human factors of such systems. Buchanan presents some design guidelines for WAP usability related to both the reading and writing of content. The need for navigation should be minimised and the number of keystrokes necessary to interact with the service should be reduced as much as possible.

5 CONCLUSION

In this paper we have presented a mobile task reporting system and shown how this system depends on the mobile device used as the client. We found that the mobile phone can be a helpful tool for checking the status of tasks because of its portability and connectivity. The limited user-interaction capabilities of WAP is however a problem when writing text. The input capability of can however be improved by integrating intelligent dictionaries into the application like done in e.g. the Nokia SMS application [6], and by providing other types of interaction like selections. The number of keystrokes needed to write a word might thereby in the best case be reduced to the number of characters in the word, or just a simple selection from a list. The PDA with bigger screen and more convenient text input is a very useful device for entering end-task reports. The lack of connectivity can however be a problem because of possible conflicts with other off-line devices. To overcome this problem, we have described some solutions in section 3.3.2. This means that the MTRS system must include some advanced transaction support that can handle off-line devices and conflicts between these devices between data synchronisations. Because several policies can be used to solve this problem (from exclusive locks to anarchy), it is important that a flexible transaction framework is used that can adapt to policies of different companies. One example of such a framework is the CAGIS-trans [7], which is likely to be integrated to the MTRS system in the future. A portable PC can be too big to be carried around as a mobile toolbox. However, for getting overview of tasks and for managing tasks (delegation etc.) big screens are needed. Also if a lot of text should be entered into the system, the portable PC's keyboard is very convenient.

The process of designing and finding requirements for a mobile system is not always straightforward. As an aid to identify requirements for systems to support different mobile scenarios, we recommend the MOWAHS characterisation framework for mobile work [9]. This framework can be used to characterise mobile work in order to elicit functional and non-functional requirements for a mobile process support system, and specify and analyse mobile scenarios in detail. The main advantage with this framework is that it is independent of hardware and network resources. This means that it is possible to analyse future mobile systems without the constraints of the existing mobile devices.

6 ACKNOWLEDGEMENT

This paper is a result of work in a project called MObile Work Across Heterogeneous Systems (MOWAHS) [5]. The MOWAHS project is sponsored by the Norwegian Research Council's IT2010 program.

REFERENCES

[1] AvantGo. Web browsing and caching for wireless devices. http://avantgo.com, Accessed February 18th 2003.

[2] George Buchanan, Sarah Farrant, Matt Jones, Harold Thimbleby, Gary Marsden, and Michael Pazzani. Improving Mobile Internet Usability. In *Proceedings of the Tenth International Conference on World Wide Web*, pages 673–680. ACM Press, 2001.

[3] Peter Johnson. Usability and Mobility; Interactions on the move. In *First Workshop on Human Computer Interaction with Mobile Devices*, May 1998.

[4] Aaron Marcus. Babyface Design for Mobile Devices and the Web. In *Human-Computer Interface International Conference*, volume 2, 5-10 August 2001.

[5] MOWAHS. MObile Work Across Heterogeneous Systems. Web: http://www.mowahs.com, 2001.

[6] Nokia. Nokia on the Web. http://www.nokia.com, 2002.

[7] Heri Ramampiaro and Mads Nygård. CAGISTrans: A Transactional Framework for Cooperative Work. In *Proc. of the 14th Inter'l Conf. on Parallel and Distributed Computing Systems (PDCS2001)*, pages 43–50, August 2001.

[8] Craig Ruefenacht. The RUST Mail System. Web: http://www.eng.utah.edu/rust/docs/, 1996.

[9] 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 the IASTED International Conference in Applied Informatics 2002 (AI'2002)*, pages 258–264, February 18-22 2002.

[10] Jonathan Trevor, David M. Hilbert, Bill N. Schilit, and Tzu Khiau Koh. From Desktop to Phonetop: A UI For Web Interaction On Very Small Devices. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, pages 121–130. ACM Press, 2001.

[11] Alf Inge Wang, Jens-Otto Larsen, Reidar Conradi, and Bjørn Munch. Improving Cooperation Support in the EPOS CM System. In Volker Grühn, editor, *Proc. EWSPT'98, London, 18-19. Sept. 1998*, page 17, September 1998.

[12] WapForum. Wireless Application Protocol Architecture Specification. http://www.wapforum.org, 2002.

[13] World Wide Web Consortium – W3C. Extensible Markup Language (XML). http://www.w3.org/XML/, 2002.

[14] World Wide Web Consortium (w3c). HyperText Markup Language (HTML). http://www.w3c.org/MarkUp/, 2002.

[15] Øystein Hoftun and Kenneth Aron Bjørkhaugen. Task Reporting System. Technical report, Dep. of Computer and Information Science, NTNU, Norway, November 2001.